

INSTITUTO TECNOLÓGICO DE ESTUDIOS SUPERIORES DE ZAMORA

ARQUITECTURA DE SERVICIOS

ROBERTO SUAREZ ZINZUN

SEGUNDO AVANCE

MONSERRAT GUADALUPE HERRERA MÉNDEZ

21010288

DIEGO ARMANDO CAZAREZ RAMIREZ

21010268

BRAYAN VARGAS BECERRA

21010324

GERARDO DAVID RAMIREZ CANCHOLA

21010304

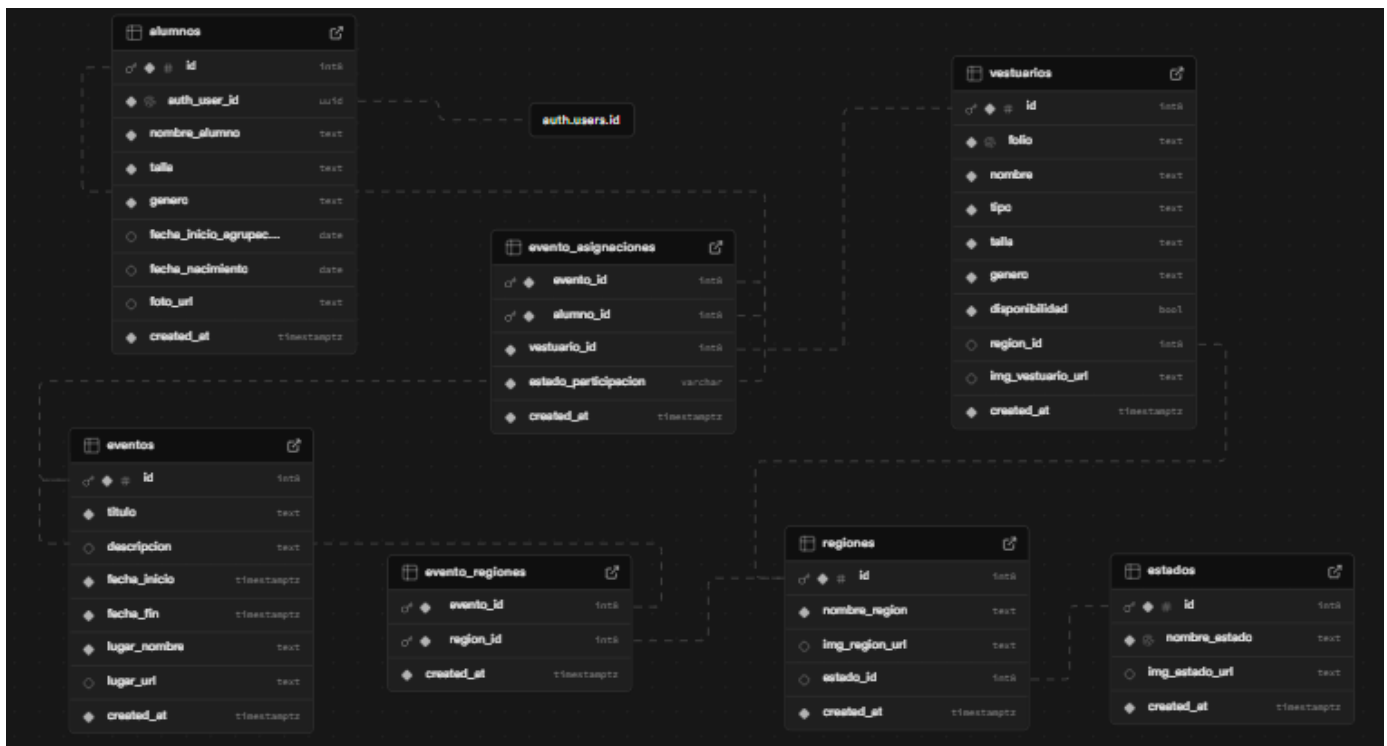
ISC

14-03-2025

ZAMORA, MICHOACÁN

1. Diseño del modelo de datos.

El diseño elegido para el desarrollo de este proyecto es de una BD SQL relacional debido a que por la naturaleza de la aplicación resulta conveniente al momento de estructurar los datos y del manejo de las relaciones de datos que nuestra aplicación requiere. Debido a esto, se mantuvo el manejo de base de datos que se venía usando previo a la decisión de implementar arquitectura de microservicios, manejando PostgreSQL para la base de datos.



2. Script de la BD:

A continuación, se adjuntan los scripts necesarios para la creación de la BD SQL para FolklorTrack, como Supabase desde su interfaz permite crear la base de datos, solo es necesario generar las tablas:

-- Tabla estados

```
CREATE TABLE public.estados (
```

```
id bigint GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
```

```
nombre_estado text NOT NULL UNIQUE,  
img_estado_url text NULL,  
created_at timestamp with time zone DEFAULT now() NOT NULL  
);
```

-- Tabla regiones

```
CREATE TABLE public.regiones (  
  id bigint GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,  
  nombre_region text NOT NULL,  
  img_region_url text NULL,  
  estado_id bigint NULL,  
  created_at timestamp with time zone DEFAULT now() NOT NULL,  
  CONSTRAINT fk_estado  
    FOREIGN KEY (estado_id)  
    REFERENCES public.estados(id)  
    ON DELETE SET NULL  
    ON UPDATE CASCADE,  
  UNIQUE (nombre_region, estado_id)  
);
```

-- Tabla alumnos

```
CREATE TABLE public.alumnos (  
  id bigint GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY, -- ID propio de la tabla  
  alumnos  
  auth_user_id uuid NOT NULL UNIQUE, -- Clave Foránea al ID del usuario en  
  Supabase Auth  
  nombre_alumno text NOT NULL, -- Puedes mantenerlo o tomarlo de user_metadata  
  talla text NOT NULL,  
  genero text NOT NULL CHECK (genero IN ('Hombre', 'Mujer', 'Otro')),
```

```
fecha_inicio_agrupacion date NULL,  
fecha_nacimiento date NULL,  
created_at timestamp with time zone DEFAULT now() NOT NULL,  
CONSTRAINT fk_auth_user  
    FOREIGN KEY (auth_user_id)  
    REFERENCES auth.users(id) -- Referencia a la tabla de usuarios de Supabase Auth  
    ON DELETE CASCADE  
    ON UPDATE CASCADE  
);
```

-- Tabla vestuarios

```
CREATE TABLE public.vestuarios (  
    id bigint GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,  
    folio text UNIQUE NOT NULL,  
    nombre text NOT NULL,  
    tipo text NOT NULL,  
    talla text NOT NULL,  
    genero text NOT NULL CHECK (genero IN ('Hombre', 'Mujer', 'Unisex')),  
    disponibilidad boolean DEFAULT true NOT NULL,  
    region_id bigint NULL, -- FK a regiones  
    img_vestuario_url text NULL,  
    created_at timestamp with time zone DEFAULT now() NOT NULL,  
    CONSTRAINT fk_region_vestuario  
        FOREIGN KEY(region_id)  
        REFERENCES public.regiones(id)  
        ON DELETE SET NULL  
        ON UPDATE CASCADE
```

);

-- Tabla eventos

```
CREATE TABLE public.eventos (  
    id bigint GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,  
    titulo text NOT NULL,  
    descripcion text NULL,  
    fecha_inicio timestamp with time zone NOT NULL,  
    fecha_fin timestamp with time zone NOT NULL,  
    lugar_nombre text NOT NULL,  
    lugar_url text NULL,  
    created_at timestamp with time zone DEFAULT now() NOT NULL,  
    CONSTRAINT chk_fechas_evento CHECK (fecha_fin >= fecha_inicio)  
);
```

-- Tabla de Unión: evento_regiones

```
CREATE TABLE public.evento_regiones (  
    evento_id bigint NOT NULL,  
    region_id bigint NOT NULL,  
    created_at timestamp with time zone DEFAULT now() NOT NULL,  
    PRIMARY KEY (evento_id, region_id),  
    CONSTRAINT fk_evento_er  
        FOREIGN KEY(evento_id)  
        REFERENCES public.eventos(id)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE,  
    CONSTRAINT fk_region_er  
        FOREIGN KEY(region_id)
```

```

REFERENCES public.regiones(id)

ON DELETE CASCADE

ON UPDATE CASCADE

);

-- Tabla evento_asignaciones

CREATE TABLE public.evento_asignaciones (

evento_id bigint NOT NULL,

alumno_id bigint NOT NULL, -- FK a public.alumnos.id

vestuario_id bigint NOT NULL, -- FK a public.vestuarios.id

estado_participacion VARCHAR(20) NOT NULL DEFAULT 'Invitado' CHECK
(estado_participacion IN ('Invitado', 'Confirmado', 'Rechazado', 'Asistio')),

created_at timestamp with time zone DEFAULT now() NOT NULL,

UNIQUE (evento_id, alumno_id), -- Un alumno solo puede tener un registro por
evento

CONSTRAINT fk_evento_ea

FOREIGN KEY (evento_id)

REFERENCES public.eventos(id)

ON DELETE CASCADE

ON UPDATE CASCADE,

CONSTRAINT fk_alumno_ea

FOREIGN KEY (alumno_id)

REFERENCES public.alumnos(id)

ON DELETE CASCADE

ON UPDATE CASCADE,

CONSTRAINT fk_vestuario_ea

FOREIGN KEY (vestuario_id)

REFERENCES public.vestuarios(id)

```

ON DELETE RESTRICT -- Evita eliminar un vestuario si está activamente asignado

ON UPDATE CASCADE

);

3. Carga inicial de datos

Creación de usuarios en Supabase necesaria para vincular alumnos:

--Crear usuario Admin en Supabase:

--email: admin@folklor.com

--contraseña: adminFolk123

--Modificación en SQL para rol y nombre:

UPDATE auth.users

SET raw_app_meta_data = COALESCE(raw_app_meta_data, '{}::jsonb') || '{"rol":
"Admin"}::jsonb

WHERE id = '49241247-2a6f-4340-910a-0b2a73658c5e';

UPDATE auth.users

SET raw_user_meta_data = COALESCE(raw_user_meta_data, '{}::jsonb') ||
'{"display_name": "Juan Martinez"}::jsonb

WHERE id = '49241247-2a6f-4340-910a-0b2a73658c5e';

--Crear usuario alumno en Supabase:

--email: gera.ramirez@folklor.com

--contraseña: geraRmz123

--Modificación en SQL para rol y nombre:

UPDATE auth.users

SET raw_app_meta_data = COALESCE(raw_app_meta_data, '{}::jsonb') || '{"rol":
"Alumno"}::jsonb

WHERE id = '084e9a18-5c40-4d48-aaca-f6b100e9ae79';

UPDATE auth.users

SET raw_user_meta_data = COALESCE(raw_user_meta_data, '{}::jsonb') ||
'{"display_name": "Gerardo Ramirez"}::jsonb

WHERE id = '084e9a18-5c40-4d48-aaca-f6b100e9ae79';

Insertar Estados con csv

id,nombre_estado,img_estado_url,created_at

1,Michoacán,https://yclpghrhxzbdbszvnnq.supabase.co/storage/v1/object/public/folklortrack/estados/Michoacan.jpeg,2024-03-15 10:00:00-06

Insertar Regiones con csv

id,nombre_region,estado_id,img_region_url,created_at
1,Tierra Caliente
(Mich.),1,https://yclpghrhxzbdbszvnnq.supabase.co/storage/v1/object/public/folklortrack/regiones/TierraCaliente.jpg,2024-03-15 10:05:00-06

Insertar Alumnos con csv

id,auth_user_id,nombre_alumno,talla,genero,fecha_inicio_agrupacion,fecha_nacimiento,foto_url,created_at
1,084e9a18-5c40-4d48-aaca-f6b100e9ae79,Gerardo
Ramirez,M,Hombre,2017-10-19,2003-05-25,,2024-03-15 10:10:00-06

Insertar Vestuarios con csv

id,folio,nombre,tipo,talla,genero,disponibilidad,region_id,img_vestuario_url,created_at
1,MICH-TC-H01,Traje de Apatzingan Hombre,Traje
Completo,M,Hombre,true,1,https://yclpghrhxzbdbszvnnq.supabase.co/storage/v1/object/public/folklortrack/vestuarios/TierraCaliente_Hombre.png,2024-03-15 10:15:00-06

Insertar Eventos con csv

id,titulo,descripcion,fecha_inicio,fecha_fin,lugar_nombre,lugar_url,created_at
1,Gala Anual Folklorica,Presentación de los mejores bailes del año,2025-12-07
19:00:00-06,2025-12-07 21:30:00-06,Teatro Principal,None,2025-04-12
10:20:00-06

Insertar Regiones participantes en Eventos con csv

evento_id,region_id,created_at
1,1,2025-04-12 10:25:00-06

Insertar asignación a participantes en Eventos con csv

evento_id,alumno_id,vestuario_id,estado_participacion,created_at
1,1,1,Confirmado,2025-04-15 15:26:00-06

4. Definición de los requerimientos de servicios:

Nombre del Servicio	Operaciones Expuestas	Actores	Encargado(s)	URL	Método HTTP	Lógica de Negocio	Entrada (JSON)	Salida (JSON)
Servicio de Miembros	Registrar Nuevo Usuario Alumno	Admin	Brayan	Miembros/RegistrarMiembro	POST	Crea un nuevo usuario en Supabase Auth con rol "Alumno" y contraseña. Luego, crea un registro en la tabla public.alumnos con los detalles del perfil, vinculándolo al auth_user_id recién creado. El nombre_alumno también se guarda en raw_user_meta_data de Supabase.	{ "email": "nuevo.alumno@folklor.com", "password": "claveSegura123", "nombre_alumno": "Juan Pérez", "talla": "M", "genero": "Hombre", "fecha_inicio_agrupacion": "2023-01-15", "fecha_nacimiento": "2000-05-20", "foto_url": "url_a_foto.jpg" }	{ "id_alumno": 1, "auth_user_id": "uuid-generado-por-supabase", "email": "nuevo.alumno@folklor.com", "nombre_alumno": "Juan Pérez", "rol": "Alumno", "talla": "M", "genero": "Hombre", "fecha_inicio_agrupacion": "2023-01-15", "fecha_nacimiento": "2000-05-20", "foto_url": "url_a_foto.jpg", "created_at": "timestamp" }
Servicio de Miembros	Autenticar Usuario (Login)	Admin, Alumno	Monseratt	Miembros/IngresarMiembro	POST	Valida las credenciales (email y contraseña) del usuario contra Supabase Auth. Si son correctas, devuelve un token de acceso (JWT) y datos básicos del usuario, incluyendo su rol.	{ "email": "usuario@folklor.com", "password": "suClave" }	{ "access_token": "jwt_token_aqui", "token_type": "bearer", "expires_in": 3600, "refresh_token": "jwt_refresh_token_aqui", "user": { "id": "uuid-del-usuario", "email": "usuario@folklor.com", "rol": "Alumno", "nombre_display": "Nombre Usuario" }}}
Servicio de Miembros	Ver Mi Perfil	Admin, Alumno	Diego	Miembros/MiPerfil	GET	Obtiene y devuelve la información detallada del perfil del usuario autenticado. La información se extrae de public.alumnos y se complementa con datos de auth.users (como el email y rol). El ID del usuario se	N/A	{ "id_alumno": 1, "auth_user_id": "uuid-del-usuario-autenticado", "email": "usuario.actual@folklor.com", "nombre_alumno": "Nombre Alumno Actual", "rol": "Alumno", "talla": "M", "genero": "Hombre", "fecha_inicio

						obtiene del token de autenticación.		_agrupacion": "2023-01-15", "fecha_nacimiento": "2000-05-20", "foto_url": "url_a_foto.jpg", "created_at": "timestamp" }
Servicio de Miembros	Actualizar Mi Perfil	Admin, Alumno	Gerardo	Miembros/ MiPerfil/Act	PUT	Permite al usuario autenticado modificar los datos de su perfil en public.alumnos . Si se actualiza nombre_alumno, también se actualiza display_name en raw_user_meta_data de auth.users. No se permite cambiar email, password o rol aquí.	{ "nombre_alumno": "Juan Pérez Actualizado", "talla": "L", "genero": "Hombre", "fecha_inicio_agrupacion": "2023-02-01", "fecha_nacimiento": "2000-05-21", "foto_url": "nueva_url_foto.jpg" }	{ "id_alumno": 1, "auth_user_id": "uuid-del-usuario-autenticado", "email": "usuario.actual@folklor.com", "nombre_alumno": "Juan Pérez Actualizado", "rol": "Alumno", ... }
Servicio de Miembros	Listar Perfiles Alumnos	Admin	Brayan	Miembros/ alumnos	GET	Obtiene una lista de todos los usuarios que tienen el rol "Alumno". Devuelve información resumida de cada alumno, incluyendo datos de public.alumnos y auth.users. (Solo accesible para Administradores).	N/A	[{ "id_alumno": 1, "auth_user_id": "uuid-alumno-1", "email": "alumno1@folklor.com", "nombre_alumno": "Alumno Uno", "talla": "M", "genero": "Mujer" // ... otros campos relevantes }, { "id_alumno": 2, "auth_user_id": "uuid-alumno-2", "email": "alumno2@folklor.com", "nombre_alumno": "Alumno Dos", "talla":

								"S", "genero": "Hombre", ... }]
Servicio de Miembros	EliminarAlumno	Admin	Monseratt	Miembros/alumnos/<IdAlumno>	DELETE	Elimina un alumno específico identificado por su id (de public.alumnos). Esto debe implicar la eliminación del registro en public.alumnos y también la eliminación del usuario correspondiente en Supabase Auth. (Solo accesible para Administradores).	N/A	{ "mensaje": "Alumno eliminado exitosamente" }
Servicio de Inventario	Crear Nuevo Estado	Admin	Gerardo	Estados/RegistrarRegion	POST	Crea un nuevo registro de estado en la tabla public.estados.	{ "nombre_estado": "Jalisco", "img_estado_url": "url_imagen_jalisco.jpg" }	{ "id":1, "nombre_estado": "Jalisco", "img_estado_url": "url_imagen_jalisco.jpg" "created_at": "timestamp" }
Servicio de Inventario	Listar Todos los Estados	Admin, Alumno	Brayan	Estados/	GET	Obtiene una lista de todos los estados registrados en public.estados.	N/A	[{ "id":1, "nombre_estado": "Jalisco", "img_estado_url": "url_imagen_jalisco.jpg", "created_at": "timestamp" }, { "id":2, "nombre_estado": "Michoacán", "img_est

								ado_url": "url_imagen_michoacan.jpg", "created_at": "timestamp"}]
Servicio de Inventario	Obtener Detalles de un Estado	Admin, Alumno	Diego	Estados/<IdEstado>	GET	Obtiene la información detallada de un estado específico por su id.	N/A	{ "id":1,"nombre_estado": "Jalisco", "img_estado_u rl": "url_imagen_jalisco.jpg", "created_at": "timestam p" }
Servicio de Inventario	Eliminar un Estado	Admin	Monserat	Estados/<IdEstado>	DELETE	Elimina un estado. Considerar la restricción ON DELETE SET NULL en public.regiones.estado_id. Las regiones asociadas tendrán estado_id = NULL.	N/A	{ "mensaje": "Estado eliminado exitosamente" }
Servicio de Inventario	Crear nueva región	Admin	Gerardo	Regiones/ RegistrarRegion	POST	Crea una nueva región, asociándola a un estado_id existente. El campo nombre_region y estado_id combinados deben ser únicos.	{ "nombre_region": "Tierra Caliente", "img_region_u rl": "url_tierra_caliente.jp g", "estado_id": 1 }	{ "id":1, "nombre_region": "Tierra Caliente", "img_region_u rl": "url_tierra_caliente.jp g", "estado_id": 1, "created _at": "timestamp" }
Servicio de Inventario	Listar Todas las Regiones	Admin/Alumno	Brayan	Regiones/	GET	Obtiene una lista de todas las regiones. Se puede incluir el nombre del estado asociado.	N/A	[{ "id":1, "nombre_region": "Tierra Caliente", "img_region_u rl": "url_tierra_caliente.jp g",

								"estado_id":1, "nombre_estado": "Michoacán", "created_at":"timestamp", ...]
Servicio de Inventario	Obtener detalles de una región	Admin/Alumno	Diego	Regiones/ <IdRegion >	GET	Obtiene la información detallada de una región específica por su id, incluyendo el nombre del estado asociado.	N/A	{ "id":1, "nombre_region": "Tierra Caliente", "img_region_url":"url_tierra_caliente.jpg", "estado_id":1, "nombre_estado":"Michoacán", "created_at":"timestamp" }
Servicio de Inventario	Eliminar una Región	Admin	Monseratt	Regiones/ <IdRegion >	DELETE	Elimina una región. Considerar la restricción ON DELETE SET NULL en public.vestuarios.region_id. Los vestuarios asociados tendrán region_id = NULL.	N/A	{ "mensaje":"Región eliminada exitosamente" }
Servicio de Inventario	Registrar nuevo vestuario	Admin	Gerardo	Vestuarios/RegisVestuario	POST	Crea un nuevo registro de vestuario. El folio debe ser único. El género se valida con CHECK constraint.	{ "folio":"JAL-CH-H01", "nombre":"Traje de Charro Gala", "tipo":"Traje Completo", "talla":"M", "genero": "Hombre", "disponibilidad":true, "region_id":2, "img_vestuario_url":"url_charro_hombre.jpg" }	{ "id":1, "folio": "JAL-CH-M01", "nombre": "Traje de Charro Gala", "tipo":"Traje Completo", "talla":"M", "genero":"Hombre", "disponibilidad":true,"region_id":2, "img_vestuario_url":"url_charro_hombre.jpg", "created_at": "timestamp" }

Servicio de Inventario	Listar vestuarios	Admin/Alumno	Brayan	Vestuarios /	GET	Obtiene una lista de todos los vestuarios. Permite filtros opcionales por region_id, tipo, genero, disponibilidad, talla. Incluye nombre de la región.	N/A	[{ "id":1, "folio":"JAL-CH-H01", "nombre":"Traje de Charro Gala", "tipo":"Traje Completo", "talla":"M", "genero":"Mujer", "disponibilidad":true, "region_id":2, "nombre_region":"Jalisco Sierra","img_vestuario_url":"url_charro_hombre.jpg", "created_at":"timestamp" }, ...]
Servicio de Inventario	Obtener Detalles de un Vestuario	Admin, Alumno	Diego	Vestuarios /<IdVestuario>	GET	Obtiene la información detallada de un vestuario específico por su id, incluyendo el nombre de la región asociada (si existe).	N/A	{ "id":1, "folio": "JAL-CH-H01", "nombre":"Traje de Charro Gala", "tipo":"Traje Completo", "talla":"M", "genero": "Mujer", "disponibilidad":true, "region_id":2, "nombre_region":"Jalisco Sierra", "img_vestuario_url":"url_charro_hombre.jpg", "created_at":"timestamp" }
Servicio de Inventario	Actualizar Detalles de un Vestuario	Admin	Monseratt	Vestuarios /<IdVestuario>	PUT	Modifica la información de un vestuario existente.	{ "folio":"JAL-CH-H01-MOD", "nombre":"Traje de Charro Gala Hombre","tipo": "Gala", "talla": "S", "genero":"Hombre", "region_id":3,"img_vestuario_url":"nueva_url.jpg"	{ "id":1,,"folio":"JAL-CH-H01-MOD,..." }

							}	
Servicio de Inventario	Actualizar Disponibilidad de un Vestuario	Admin	Gerardo	Vestuarios /<IdVestuario>	PUT	Cambia el estado de disponibilidad de un vestuario (true/false).	{ "disponibilidad":false }	{ "id":1, "folio":"JAL-CH-H01", "nombre":"Traje de Charro Gala Hombre", "disponibilidad":false, ... }
Servicio de Inventario	Eliminar un Vestuario	Admin	Gerardo	Vestuarios /<IdVestuario>	DELETE	Elimina un vestuario. La FK en evento_asignaciones.vestuario_id tiene ON DELETE RESTRICT, por lo que no se podrá eliminar si está asignado activamente a un evento.	N/A	{ "id":1,"titulo":"Gala Anual de Danza", "descripcion":"Presentación de los mejores bailes del año.", "fecha_inicio":"2024-12-07T19:00:00-06:00", "fecha_fin":"2024-12-07T21:30:00-06:00", "lugar_nombre":"Teatro Principal", "lugar_url":"http://maps.google.com/..", "created_at":"timestamp" }
Servicio de Eventos	Crear Nuevo Evento	Admin	Brayan	Eventos/CrearEvento	POST	Crea un nuevo evento en public.eventos. La fecha_fin debe ser mayor o igual a fecha_inicio.	{ "titulo": "Gala Anual de Danza","descripcion":"Presentación de los mejores bailes del año.", "fecha_inicio":"2024-12-07T19:00:00-06:00", "fecha_fin": "2024-12-07T21:30:00-06:00", "lugar_nombre":"Teatro Principal", "lugar_url":" http://maps.google.com/ . " "	{ "eventos": [{ "id":1, "titulo":"Gala Anual de Danza", "fecha_inicio":"2024-12-07T19:00:00-06:00", "lugar_nombre":"Teatro Principal" },], "total_eventos":25 }

							}	
Servicio de Eventos	Listar Todos los Eventos	Admin/Alumno	Diego	Eventos/	GET	Obtiene una lista de todos los eventos.	N/A	{ "eventos": [{ "id":1, "titulo":"Gala Anual de Danza", "fecha_inicio":"2024-12-07T19:00:00-06:00", "lugar_nombre":"Teatro Principal" },], "total_eventos":25 }
Servicio de Eventos	Obtener Detalles de un Evento	Admin/Alumno	Monseratt	Eventos/<IdEvento>	GET	Obtiene la información detallada de un evento específico por su id. Puede incluir las regiones asociadas.	N/A	{ id":1, "titulo":"Gala Anual de Danza,"regiones_asociadas": [{ "id_region":1,"nombre_region":"Tierra Caliente (Mich.)"}, { "id_region": 5, "nombre_region": "Zona Norte (Jal.)" }], "created_at":"timestamp"}, "created_at":"timestamp"} }
Servicio de Eventos	Actualizar un Evento	Admin	Gerardo	Eventos/<IdEvento>/Actualizar Evento	PUT	Modifica la información de un evento existente.	{ "titulo": "Gala Anual de Danza - Edición Especial", "descripcion": "Nueva descripción.", "fecha_inicio": "2024-12-08T19:00:00-06:00",... }	{ "id":1, "titulo": "Gala Anual de Danza - Edición Especial", ... }
Servicio de Eventos	Eliminar un Evento	Admin	Brayan	Eventos/<IdEvento>	DELETE	Elimina un evento. Las FKs en evento_regiones y evento_asignac	N/A	{ "mensaje":"Evento eliminado exitosamente"

						iones tienen ON DELETE CASCADE, por lo que las asociaciones y asignaciones se eliminarán automáticamente.		}
Servicio de Eventos	Asociar región a evento	Admin	Monseratt	Eventos/<ldEvento>/regiones	POST	Asocia una región existente a un evento existente, creando un registro en la tabla de unión public.evento_regiones. La combinación evento_id y region_id es la PK.	{ "region_id": 3 }	{ "evento_id":1,"region_id":3,"created_at":"timestamp" }
Servicio de Eventos	Desasociar región a evento	Admin	Diego	Eventos/<ldEvento>/regiones	DELETE	Elimina la asociación entre un evento y una región, borrando el registro de public.evento_regiones.	N/A	{ "mensaje":"Región desasociada del evento exitosamente" }
Servicio de Eventos	Listado regiones asociadas	Admin/Alumno	Gerardo	Eventos/<ldEvento>/regiones	GET	Obtiene una lista de todas las regiones asociadas a un evento específico.	N/A	[{ "region_id":3,"nombre_region":"Costa Sierra", "img_region_url": "https://regionURL", "estado_id":2, "nombre_estado":"Colima" }, ...]
Servicio de Participación	Asignar Alumno y	Admin	Brayan	Eventos_participante	POST	Crea un nuevo registro en evento_asignaci	{	{ "evento_id":<idEvento>,"alumno_id":5, "vestuario

	Vestuario a Evento			s/RegParticipantes		ones. El estado por defecto es 'Invitado'. La combinación evento_id y alumno_id debe ser única. Valida que el vestuario exista y esté disponible (esto último es una lógica de negocio, no una constraint de BD necesariamente).	"alumno_id":5, "vestuario_id":12 }	_id":12,"estado_participacion":"Invitado", "created_at":"timestamp"} }
Servicio de Participación	Listar asignados a evento	Admin	Diego	Eventos_participantes/<IdEvento>	GET	Obtiene una lista de todas las asignaciones para un evento específico, incluyendo detalles del alumno y vestuario.	N/A	[{ "evento_id":<idEvento>," alumno_id":5,"nombre_alumno":"Ana Pérez", "vestuario_id":12 , "nombre_vestuario":"Floreada", "estado_participacion":"Invitado", "created_at":"timestamp" },...]
Servicio de Participación	Confirmar Participación en Evento	Alumno	Montserrat	Eventos_participantes/<IdEvento>	PUT	Permite a un alumno autenticado cambiar el estado_participacion de una asignación específica a 'Confirmado'. <idEventoAsignacion> podría ser una PK compuesta	N/A	{ "evento_id":<idEvento>," alumno_id":<idAlumnoAutenticado>, "vestuario_id":12, "estado_participacion":"Confirmado", "updated_at":"timestamp" }

						de evento_id y alumno_id (identificado por el token). El alumno solo puede confirmar su propia asignación.		
Servicio de Participación	Modificar vestuario asignado	Admin	Gerardo	Eventos_participantes/<IdEvento>	PUT	Permite al administrador cambiar el vestuario_id para una asignación existente de un alumno a un evento. Valida que el nuevo vestuario exista y esté disponible.	{ "alumno_id":5, "nuevo_vestuario_id":3, }	{ "evento_id":<idEvento>," alumno_id":<idAlumno>," vestuario_id":18,"estado_participacion":"Invitado" }
Servicio de Participación	Rechazar Participación en Evento	Alumnp	Brayan	Eventos_participantes/<IdEvento>	PUT	Permite a un alumno autenticado cambiar el estado_participacion de una asignación específica a 'Rechazado'. El alumno solo puede rechazar su propia asignación.	N/A	{ "evento_id":<idEvento>," alumno_id":<idAlumnoAutenticado>," vestuario_id":12,"estado_participacion":"Rechazado", "updated_at":"timestamp" }
Servicio de Notificaciones	Enviar Notificación de Nuevo Evento Agendado	Sistema	Diego	Notificaciones/nuevo_evento	POST	Disparado cuando se crea un nuevo evento. Envía una notificación a los alumnos relevantes (ej. todos, o aquellos cuyas regiones de interés coinciden con las regiones	{ "evento_id":1, "titulo_evento": "Gala Anual de Danza", "fecha_inicio": "2024-12-07T19:00:00-06:00", "lugar_nombre": "Teatro Principal", "ids_regiones_evento": [1,5], "mensaje	{ "estado": "Notificaciones de nuevo evento encoladas/enviadas", "destinatarios_potenciales": 50 }

						del evento). El contenido incluye detalles básicos del evento.	_personalizado": "¡No te lo pierdas!" }	
Servicio de Notificaciones	Enviar Recordatorio de Evento Próximo	Sistema	Montserrat	Notificaciones/evento_prox	POST	Disparado periódicamente (ej. 1 día antes). Envía un recordatorio a los alumnos que han 'Confirmado' su participación en un evento próximo.	{ "evento_id": 1, "titulo_evento": "Gala Anual de Danza", "fecha_inicio": "2024-12-07T19:00:00-06:00", "tipo_recordatorio": "24_horas_antes" }	{ "estado": "Recordatorios de evento encolados/enviados", "destinatarios_confirmados": 15 }