

Sistema de Gestión de Inventario

1. Definición de Entidades y Relaciones

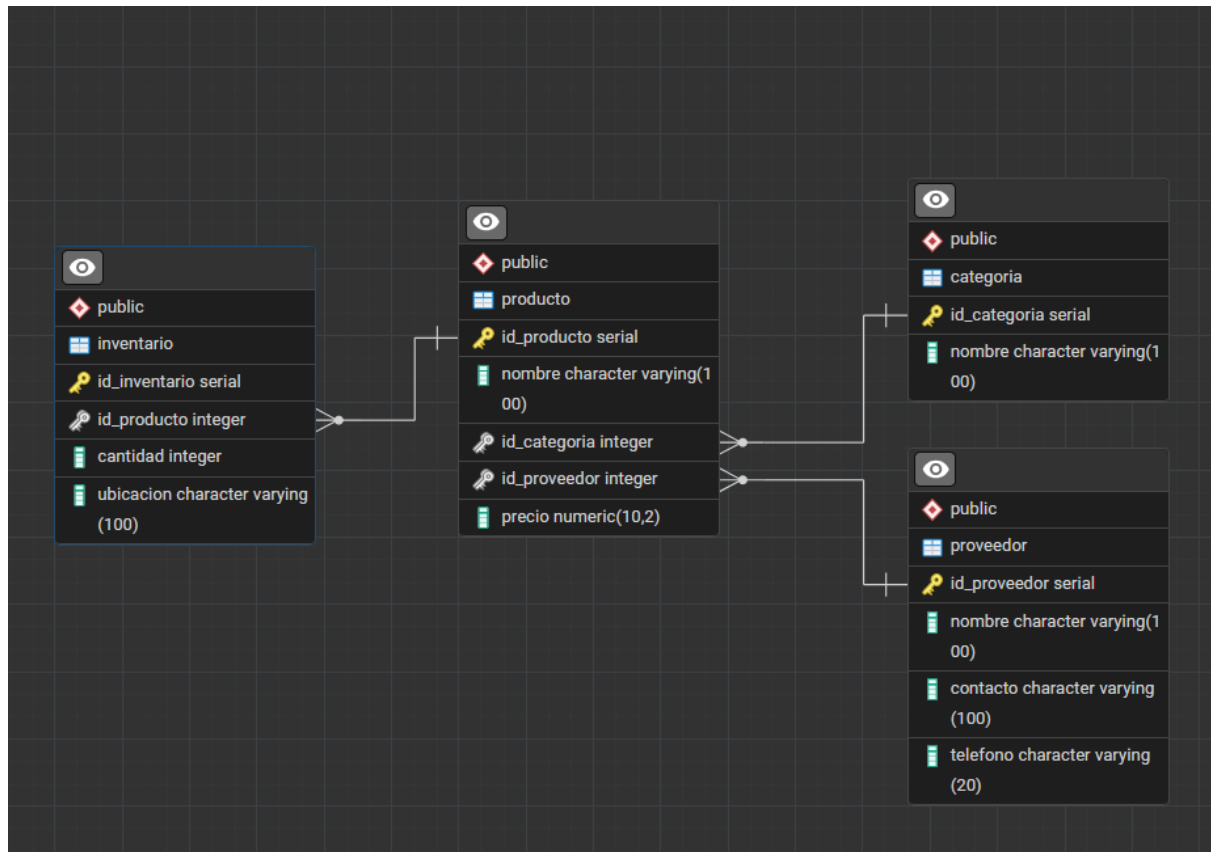
Para la gestión del inventario, se identifican las siguientes entidades clave:

- **TipoProducto:** Representa las categorías de los productos en el inventario. Contiene atributos como id_tipo y descripcion.
- **Empresa:** Representa a las empresas proveedoras de los artículos. Contiene atributos como id_empresa, nombre, contacto y telefono.
- **Articulo:** Representa los productos en inventario. Contiene atributos como id_articulo, nombre, id_tipo, id_empresa y precio.
- **Stock:** Representa la cantidad de artículos en un almacén determinado. Contiene id_stock, id_articulo, cantidad y almacen.

Relaciones

- Un **TipoProducto** puede estar asociado con varios **Articulos**, pero un **Articulo** pertenece a un solo **TipoProducto**.
- Una **Empresa** puede suministrar varios **Articulos**, pero un **Articulo** proviene de una sola Empresa.
- Un **Articulo** puede estar en múltiples **Stocks** con diferentes cantidades en distintos almacenes.

2. Diseño del Modelo Conceptual (Diagrama E-R)



3. Conversión a Esquema Relacional

A partir del modelo E-R, se definen las siguientes tablas:

TipoProducto (id_tipo, descripcion)

Empresa (id_empresa, nombre, contacto, telefono)

Articulo (id_articulo, nombre, id_tipo, id_empresa, precio)

id_tipo es clave foránea que referencia a TipoProducto(id_tipo).

id_empresa es clave foránea que referencia a Empresa(id_empresa).

Stock (id_stock, id_articulo, cantidad, almacen)

id_articulo es clave foránea que referencia a Articulo(id_articulo).

4. Implementación en SQL

Creación de Tablas

```

-- Creacion de tablas

CREATE TABLE TipoProducto (
    id_tipo SERIAL PRIMARY KEY,
    descripcion VARCHAR(100) NOT NULL
);

CREATE TABLE Empresa (
    id_empresa SERIAL PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    contacto VARCHAR(100),
    telefono VARCHAR(20)
);

CREATE TABLE Artículo (
    id_articulo SERIAL PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    id_tipo INT NOT NULL,
    id_empresa INT NOT NULL,
    precio DECIMAL(10,2) NOT NULL,
    FOREIGN KEY (id_tipo) REFERENCES TipoProducto(id_tipo),
    FOREIGN KEY (id_empresa) REFERENCES Empresa(id_empresa)
);

CREATE TABLE Stock (
    id_stock SERIAL PRIMARY KEY,
    id_articulo INT NOT NULL,
    cantidad INT NOT NULL,
    almacen VARCHAR(100),
    FOREIGN KEY (id_articulo) REFERENCES Artículo(id_articulo)
);

```

Inserción de datos

```
-- Insercion de datos

-- Tipos de Producto
INSERT INTO TipoProducto (descripcion) VALUES
('Tecnología'),
('Vestimenta'),
('Alimentación'),
('Juguetes'),
('Electrodomésticos'),
('Deportes'),
('Educación'),
('Salud'),
('Oficina'),
('Mascotas');

-- Empresas
INSERT INTO Empresa (nombre, contacto, telefono) VALUES
('TechWorld Ltd.', 'Miguel Gómez', '777-1234'),
('ModaFina', 'Elena Castro', '777-5678'),
('Delicias Gourmet', 'Pablo Méndez', '777-9012'),
('Toys Universe', 'Mariana Ríos', '777-3456'),
('HomeStyle', 'Sergio López', '777-7890'),
('FitZone', 'Andrea Medina', '777-2345'),
('BookSmart', 'Jorge Herrera', '777-6789'),
('HealthCare', 'Lucía Pérez', '777-4321'),
('OfficePlus', 'Fernando Torres', '777-8765'),
('PetCare', 'Raquel Suárez', '777-6543');
```

```

-- Artículos
INSERT INTO Articulo (nombre, id_tipo, id_empresa, precio) VALUES
('Tablet Pro', 1, 1, 950.00),
('Smartwatch Z', 1, 1, 320.00),
('Chaqueta de Cuero', 2, 2, 120.00),
('Vestido de Noche', 2, 2, 85.00),
('Galletas Orgánicas', 3, 3, 12.50),
('Barra Energética', 3, 3, 6.00),
('Muñeco Articulado', 4, 4, 45.00),
('Aspiradora Inteligente', 5, 5, 200.00),
('Raqueta de Tenis', 6, 6, 75.00),
('Libro "Bases de Datos"', 7, 7, 35.00);

-- Stock
INSERT INTO Stock (id_articulo, cantidad, almacen) VALUES
(1, 12, 'Depósito A1'),
(2, 20, 'Depósito A2'),
(3, 35, 'Depósito B1'),
(4, 25, 'Depósito B2'),
(5, 40, 'Depósito C1'),
(6, 30, 'Depósito C2'),
(7, 15, 'Depósito D1'),
(8, 18, 'Depósito E1'),
(9, 22, 'Depósito F1'),
(10, 55, 'Depósito G1');

```

5. Consultas SQL

- **Consulta requerida:** Obtener la lista de productos con sus respectivas categorías y proveedores, ordenados alfabéticamente por nombre de producto.

```

87  -- Consulta Requerida
88
89  SELECT
90      a.nombre AS articulo,
91      tp.descripcion AS tipo,
92      e.nombre AS empresa
93  FROM Articulo a
94  JOIN TipoProducto tp ON a.id_tipo = tp.id_tipo
95  JOIN Empresa e ON a.id_empresa = e.id_empresa
96  ORDER BY a.nombre ASC;

```

	articulo character varying (100)	tipo character varying (100)	empresa character varying (100)
1	Aspiradora Inteligente	Electrodomésticos	HomeStyle
2	Barra Energética	Alimentación	Delicias Gourmet
3	Chaqueta de Cuero	Vestimenta	ModaFina
4	Galletas Orgánicas	Alimentación	Delicias Gourmet
5	Libro "Bases de Datos"	Educación	BookSmart
6	Muñeco Articulado	Juguetes	Toys Universe
7	Raqueta de Tenis	Deportes	FitZone
8	Smartwatch Z	Tecnología	TechWorld Ltd.
9	Tablet Pro	Tecnología	TechWorld Ltd.
10	Vestido de Noche	Vestimenta	ModaFina

- Consulta SELECT

```

115  -- Artículos con stock menor a 20 unidades (usando SELECT)
116  SELECT
117      a.nombre AS articulo,
118      s.cantidad AS stock
119  FROM Stock s
120  JOIN Articulo a ON s.id_articulo = a.id_articulo
121  WHERE s.cantidad < 20
122  ORDER BY s.cantidad ASC;

```

	articulo character varying (100)	stock integer
1	Tablet Pro	12
2	Muñeco Articulado	15
3	Aspiradora Inteligente	18

- Consulta WHERE

```

139 -- Artículos de una empresa específica ("TechWorld Ltd.") (usando WHERE)
140 SELECT
141     a.nombre AS articulo,
142     e.nombre AS empresa
143 FROM Articulo a
144 JOIN Empresa e ON a.id_empresa = e.id_empresa
145 WHERE e.nombre = 'TechWorld Ltd.';

```

Data Output Messages Notifications

	articulo character varying (100)	empresa character varying (100)
1	Tablet Pro	TechWorld Ltd.
2	Smartwatch Z	TechWorld Ltd.

- Consulta ORDER BY

```

123
124 -- Lista de artículos y sus precios, ordenados por precio descendente (ORDER BY)
125 SELECT
126     a.nombre AS articulo,
127     a.precio
128 FROM Articulo a
129 ORDER BY a.precio DESC;

```

Data Output Messages Notifications

	articulo character varying (100)	precio numeric (10,2)
1	Tablet Pro	950.00
2	Smartwatch Z	320.00
3	Aspiradora Inteligente	200.00
4	Chaqueta de Cuero	120.00
5	Vestido de Noche	85.00
6	Raqueta de Tenis	75.00
7	Muñeco Articulado	45.00
8	Libro "Bases de Datos"	35.00
9	Galletas Orgánicas	12.50
10	Barra Energética	6.00

- Consulta GROUP BY

```

106 -- Cantidad total de artículos en el stock (usando GROUP BY y SUM)
107 SELECT
108     tp.descripcion AS tipo,
109     SUM(s.cantidad) AS total_stock
110 FROM Stock s
111 JOIN Artículo a ON s.id_articulo = a.id_articulo
112 JOIN TipoProducto tp ON a.id_tipo = tp.id_tipo
113 GROUP BY tp.descripcion;

```

	tipo character varying (100)	total_stock bigint
1	Educación	55
2	Vestimenta	60
3	Tecnología	32
4	Electrodomésticos	18
5	Juguetes	15
6	Deportes	22
7	Alimentación	70

- Consulta JOIN

```

130
131 -- Cantidad de artículos suministrados por cada empresa (usando JOIN)
132 SELECT
133     e.nombre AS empresa,
134     COUNT(a.id_articulo) AS total_articulos
135 FROM Artículo a
136 JOIN Empresa e ON a.id_empresa = e.id_empresa
137 GROUP BY e.nombre;

```

	empresa character varying (100)	total_articulos bigint
1	HomeStyle	1
2	BookSmart	1
3	Toys Universe	1
4	FitZone	1
5	Delicias Gourmet	2
6	ModaFina	2
7	TechWorld Ltd.	2

- **Consulta LIKE**

```
98 -- Buscar artículos por nombre o tipo (usando LIKE)
99 SELECT
100     a.nombre AS articulo,
101     tp.descripcion AS tipo
102 FROM Articulo a
103 JOIN TipoProducto tp ON a.id_tipo = tp.id_tipo
104 WHERE a.nombre LIKE '%Tablet%' OR tp.descripcion LIKE '%Tecnología%';
105
```

Data Output Messages Notifications

	articulo character varying (100)	tipo character varying (100)
1	Tablet Pro	Tecnología
2	Smartwatch Z	Tecnología

Sistema de Gestión de Eventos

1. Definición de Entidades y Relaciones

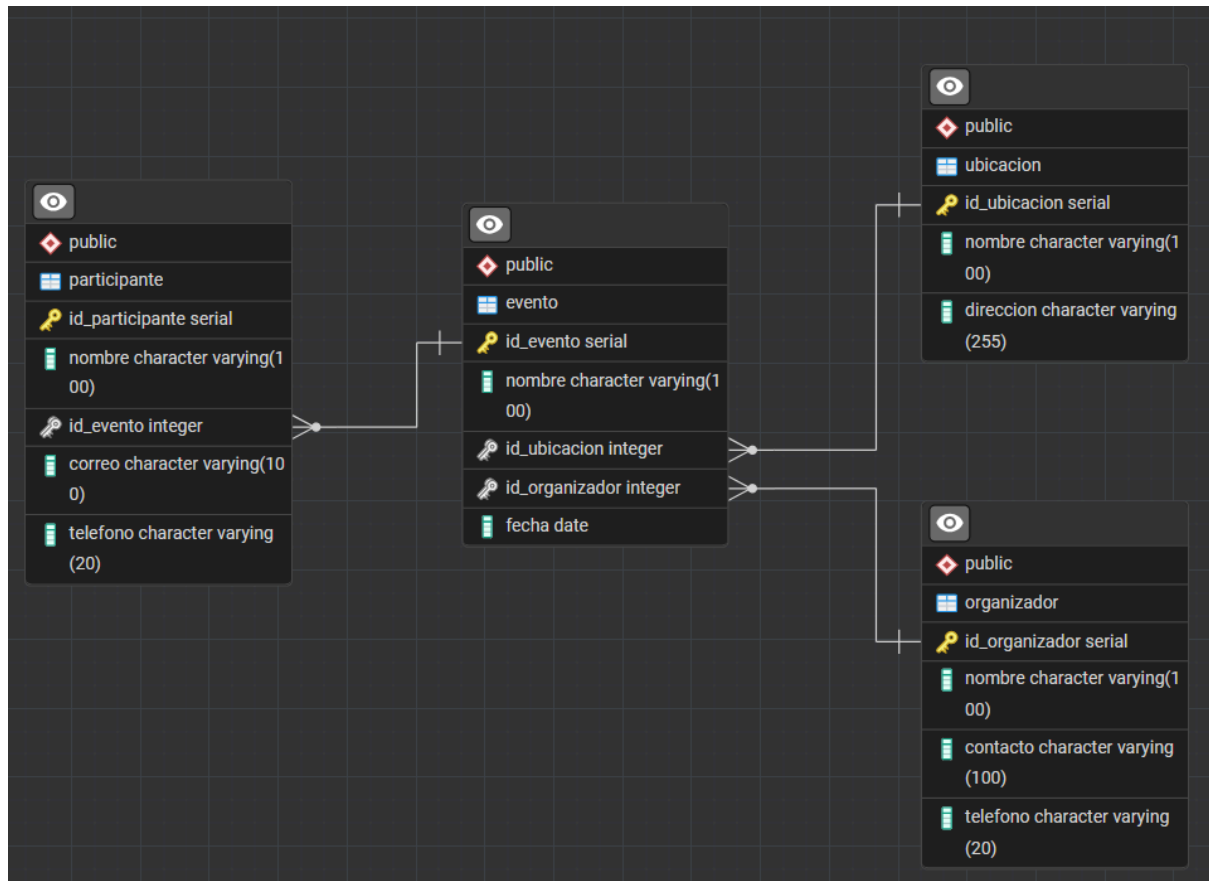
Para la gestión de eventos, se identifican las siguientes entidades clave:

- **Lugar:** Representa los espacios físicos donde se realizan los eventos. Contiene atributos como id_lugar, nombre y dirección.
- **Coordinador:** Representa a las personas encargadas de organizar y gestionar los eventos. Contiene atributos como id_coordinador, nombre, email y teléfono.
- **Evento:** Representa los eventos organizados en distintos lugares y gestionados por coordinadores. Contiene atributos como id_evento, título, id_lugar, id_coordinador y fecha_evento.
- **Asistente:** Representa a las personas registradas en un evento específico. Contiene atributos como id_asistente, nombre, email, teléfono y id_evento.

Relaciones

- Un **coordinador** puede gestionar múltiples eventos, pero cada evento tiene un solo coordinador.
- Un **evento** se lleva a cabo en una única ubicación, pero una ubicación puede albergar múltiples eventos.
- Un **evento** puede tener múltiples asistentes, pero cada asistente solo puede estar registrado en un evento a la vez.

2. Diseño del Modelo Conceptual (Diagrama E-R)



3. Conversión a Esquema Relacional

A partir del modelo E-R, se definen las siguientes tablas:

Lugar (id_lugar, nombre, dirección)
id_lugar es la clave primaria.

Coordinador (id_coordinador, nombre, email, teléfono)
id_coordinador es la clave primaria.

Evento (id_evento, titulo, id_lugar, id_coordinador, fecha_evento)
id_evento es la clave primaria.
id_lugar es clave foránea que referencia a Lugar(id_lugar).
id_coordinador es clave foránea que referencia a Coordinador(id_coordinador).

Asistente (id_asistente, nombre, email, teléfono, id_evento)
id_asistente es la clave primaria.
id_evento es clave foránea que referencia a Evento(id_evento).

4. Implementación en SQL

Creación de Tablas

```
-- Tabla de Lugares
CREATE TABLE Lugar (
    id_lugar SERIAL PRIMARY KEY,
    nombre VARCHAR(150) NOT NULL,
    direccion TEXT NOT NULL
);

-- Tabla de Coordinadores
CREATE TABLE Coordinador (
    id_coordinador SERIAL PRIMARY KEY,
    nombre VARCHAR(150) NOT NULL,
    email VARCHAR(150) UNIQUE,
    telefono VARCHAR(20)
);

-- Tabla de Eventos
CREATE TABLE Evento (
    id_evento SERIAL PRIMARY KEY,
    titulo VARCHAR(150) NOT NULL,
    id_lugar INT NOT NULL,
    id_coordinador INT NOT NULL,
    fecha_evento DATE NOT NULL,
    FOREIGN KEY (id_lugar) REFERENCES Lugar(id_lugar),
    FOREIGN KEY (id_coordinador) REFERENCES Coordinador(id_coordinador)
);

-- Tabla de Asistentes
CREATE TABLE Asistente (
    id_asistente SERIAL PRIMARY KEY,
    nombre VARCHAR(150) NOT NULL,
    email VARCHAR(150) UNIQUE,
    telefono VARCHAR(20),
    id_evento INT NOT NULL,
    FOREIGN KEY (id_evento) REFERENCES Evento(id_evento)
);
```

Inserción de datos

```
-- Insercion de datos

-- Datos para Lugares
INSERT INTO Lugar (nombre, direccion) VALUES
('Salón Diamante', 'Av. Reforma 123, CDMX'),
('Centro de Eventos Real', 'Calle Bolívar 456, Guadalajara'),
('Auditorio Central', 'Plaza Mayor, Monterrey'),
('Foro Internacional', 'Paseo del Bosque 890, CDMX'),
('Explanada del Sol', 'Carretera Nacional 320, Monterrey'),
('Centro de Convenciones', 'Calle Libertad 78, Puebla'),
('Teatro Cultural', 'Avenida Insurgentes 200, CDMX'),
('Espacio Creativo', 'Calle Morelos 56, Querétaro'),
('Palacio de Congresos', 'Paseo de la Victoria 300, León'),
('Anfiteatro del Sur', 'Carretera Federal 45, Mérida');

-- Datos para Coordinadores
INSERT INTO Coordinador (nombre, email, telefono) VALUES
('Paola Estrada', 'paola.estrada@mail.com', '555-9988'),
('Roberto Jiménez', 'roberto.jimenez@mail.com', '555-7766'),
('Mariana Duarte', 'mariana.duarte@mail.com', '555-5544'),
('Ernesto Valdés', 'ernesto.valdes@mail.com', '555-3322'),
('Fernanda Ríos', 'fernanda.rios@mail.com', '555-1100'),
('José Carranza', 'jose.carranza@mail.com', '555-2211'),
('Liliana Mendoza', 'liliana.mendoza@mail.com', '555-3344'),
('Santiago Vargas', 'santiago.vargas@mail.com', '555-4455'),
('Claudia Rosales', 'claudia.rosales@mail.com', '555-5566'),
('Emilio Salgado', 'emilio.salgado@mail.com', '555-6677');
```

```

-- Datos para Eventos
INSERT INTO Evento (titulo, id_lugar, id_coordinador, fecha_evento) VALUES
('Conferencia de IA', 1, 1, '2025-06-15'),
('Feria de Emprendedores', 2, 2, '2025-07-20'),
('Simposio de Robótica', 3, 3, '2025-08-10'),
('Taller de Innovación', 4, 4, '2025-09-05'),
('Congreso de Marketing', 5, 5, '2025-10-22'),
('Convención de Videojuegos', 6, 6, '2025-11-18'),
('Festival de Arte Digital', 7, 7, '2025-12-12'),
('Cumbre de Energías Renovables', 8, 8, '2026-01-14'),
('Foro de Educación y Tecnología', 9, 9, '2026-02-27'),
('Congreso de Medicina Avanzada', 10, 10, '2026-03-10');

-- Datos para Asistentes
INSERT INTO Asistente (nombre, email, telefono, id_evento) VALUES
('Laura Pérez', 'laura.p@mail.com', '555-1122', 1),
('Daniel Gómez', 'danielg@mail.com', '555-2233', 1),
('Carmen Salinas', 'carmen.s@mail.com', '555-3344', 2),
('Hugo Torres', 'hugotorres@mail.com', '555-4455', 2),
('Beatriz Mendoza', 'beatriz.m@mail.com', '555-5566', 3),
('Felipe Cruz', 'felipe.cruz@mail.com', '555-6677', 3),
('Andrea Villalobos', 'andrea.v@mail.com', '555-7788', 4),
('Jorge Ramírez', 'jorger@mail.com', '555-8899', 4),
('Diana Hernández', 'diana.h@mail.com', '555-9900', 5),
('Emilio Castillo', 'emilioc@mail.com', '555-0011', 5),
('Alejandra Ruiz', 'alejandra.ruiz@mail.com', '555-1221', 6),
('Rodrigo Pacheco', 'rodrigo.p@mail.com', '555-2332', 6),
('Patricia López', 'patricia.l@mail.com', '555-3443', 7),
('Luis Martínez', 'luis.martinez@mail.com', '555-4554', 7),
('Monserrat Sánchez', 'monserrat.s@mail.com', '555-5665', 8),
('Victor Orozco', 'victor.orozco@mail.com', '555-6776', 8),
('Esteban Fernández', 'esteban.f@mail.com', '555-7887', 9),
('María Navarro', 'maria.navarro@mail.com', '555-8998', 9),
('Natalia Pineda', 'natalia.p@mail.com', '555-9111', 10),
('Fernando Castro', 'fernando.c@mail.com', '555-0222', 10);

```

5. Consultas SQL

- **Consulta requerida:** Obtener la lista de eventos programados junto con la cantidad de participantes registrados por evento.

```

101  -- Consulta Requerida
102
103  -- Obtener lista de eventos con cantidad de asistentes
104  SELECT
105      e.titulo AS evento,
106      COUNT(a.id_asistente) AS total_asistentes
107  FROM Evento e
108  LEFT JOIN Asistente a ON e.id_evento = a.id_evento
109  GROUP BY e.id_evento
110  ORDER BY e.fecha_evento ASC;

```

Data Output Messages Notifications

	evento character varying (150)	total_asistentes bigint
1	Conferencia de IA	2
2	Feria de Emprendedores	2
3	Simposio de Robótica	2
4	Taller de Innovación	2
5	Congreso de Marketing	2
6	Convención de Videojuegos	2
7	Festival de Arte Digital	2
8	Cumbre de Energías Renovabl...	2
9	Foro de Educación y Tecnología	2
10	Congreso de Medicina Avanza...	2

- **Consulta SELECT**

```

--Obtener la lista de eventos programados junto con la cantidad de participantes registrados por evento. (usando SELECT)

SELECT e.titulo AS evento, COUNT(a.id_asistente) AS total_asistentes
FROM Evento e
LEFT JOIN Asistente a ON e.id_evento = a.id_evento
GROUP BY e.id_evento
ORDER BY e.fecha_evento ASC;

```

Output Messages Notifications

evento character varying (150)	total_asistentes bigint
Conferencia de IA	2
Feria de Emprendedores	2
Simposio de Robótica	2
Taller de Innovación	2
Congreso de Marketing	2
Convención de Videojuegos	2
Festival de Arte Digital	2
Cumbre de Energías Renovabl...	2
Foro de Educación y Tecnología	2
Congreso de Medicina Avanza...	2

- **Consulta WHERE**

```
-- Obtener los eventos programados después del 1 de agosto de 2025. (Usando WHERE)
```

```
SELECT * FROM Evento
WHERE fecha_evento > '2025-08-01';
```

Output Messages Notifications

id_evento [PK] integer	título character varying (150)	id_lugar integer	id_coordinador integer	fecha_evento date
3	Simposio de Robótica	3	3	2025-08-10
4	Taller de Innovación	4	4	2025-09-05
5	Congreso de Marketing	5	5	2025-10-22
6	Convención de Videojuegos	6	6	2025-11-18
7	Festival de Arte Digital	7	7	2025-12-12
8	Cumbre de Energías Renovabl...	8	8	2026-01-14
9	Foro de Educación y Tecnología	9	9	2026-02-27
10	Congreso de Medicina Avanza...	10	10	2026-03-10

- Consulta ORDER BY

```
-- Obtener la lista de eventos ordenados por fecha en orden descendente. (Usando ORDER BY)
```

```
SELECT * FROM Evento
ORDER BY fecha_evento DESC;
```

Output Messages Notifications

id_evento [PK] integer	título character varying (150)	id_lugar integer	id_coordinador integer	fecha_evento date
10	Congreso de Medicina Avanza...	10	10	2026-03-10
9	Foro de Educación y Tecnología	9	9	2026-02-27
8	Cumbre de Energías Renovabl...	8	8	2026-01-14
7	Festival de Arte Digital	7	7	2025-12-12
6	Convención de Videojuegos	6	6	2025-11-18
5	Congreso de Marketing	5	5	2025-10-22
4	Taller de Innovación	4	4	2025-09-05
3	Simposio de Robótica	3	3	2025-08-10
2	Feria de Emprendedores	2	2	2025-07-20
1	Conferencia de IA	1	1	2025-06-15

- Consulta GROUP BY

```
-- Obtener la cantidad de eventos organizados por cada coordinador. (Usando GROUP BY)

SELECT c.nombre AS coordinador, COUNT(e.id_evento) AS eventos_organizados
FROM Evento e
JOIN Coordinador c ON e.id_coordinador = c.id_coordinador
GROUP BY c.nombre;
```

Output Messages Notifications

coordinador character varying (150)	eventos_organizados bigint
Liliana Mendoza	1
Ernesto Valdés	1
Emilio Salgado	1
Claudia Rosales	1
Fernanda Ríos	1
José Carranza	1
Mariana Duarte	1
Roberto Jiménez	1
Paola Estrada	1
Santiago Vargas	1

- Consulta JOIN

```
-- Obtener la lista de asistentes junto con el nombre del evento al que asistirán. (Usando JOIN)

SELECT a.nombre AS asistente, a.email, a.telefono, e.titulo AS evento
FROM Asistente a
JOIN Evento e ON a.id_evento = e.id_evento;
```

Output Messages Notifications

asistente character varying (150)	email character varying (150)	telefono character varying (20)	evento character varying (150)
Daniel Gómez	danielg@mail.com	555-2233	Conferencia de IA
Laura Pérez	laura.p@mail.com	555-1122	Conferencia de IA
Hugo Torres	hugotorres@mail.com	555-4455	Feria de Emprendedores
Carmen Salinas	carmen.s@mail.com	555-3344	Feria de Emprendedores
Felipe Cruz	felipe.cruz@mail.com	555-6677	Simposio de Robótica
Beatriz Mendoza	beatriz.m@mail.com	555-5566	Simposio de Robótica
Jorge Ramírez	jorger@mail.com	555-8899	Taller de Innovación
Andrea Villalobos	andrea.v@mail.com	555-7788	Taller de Innovación
Emilio Castillo	emilioc@mail.com	555-0011	Congreso de Marketing
Diana Hernández	diana.h@mail.com	555-9900	Congreso de Marketing

- Consulta LIKE

```
7 -- Buscar todos los eventos que contengan la palabra "Taller" en su título. (Usando LIKE)
8
9 SELECT * FROM Evento
10 WHERE titulo LIKE '%Taller%';
11
```

Data Output Messages Notifications

id_evento [PK] integer	titulo character varying (150)	id_lugar integer	id_coordinador integer	fecha_evento date
4	Taller de Innovación	4	4	2025-09-05

Plataforma de Streaming de Música

1. Definición de Entidades y Relaciones

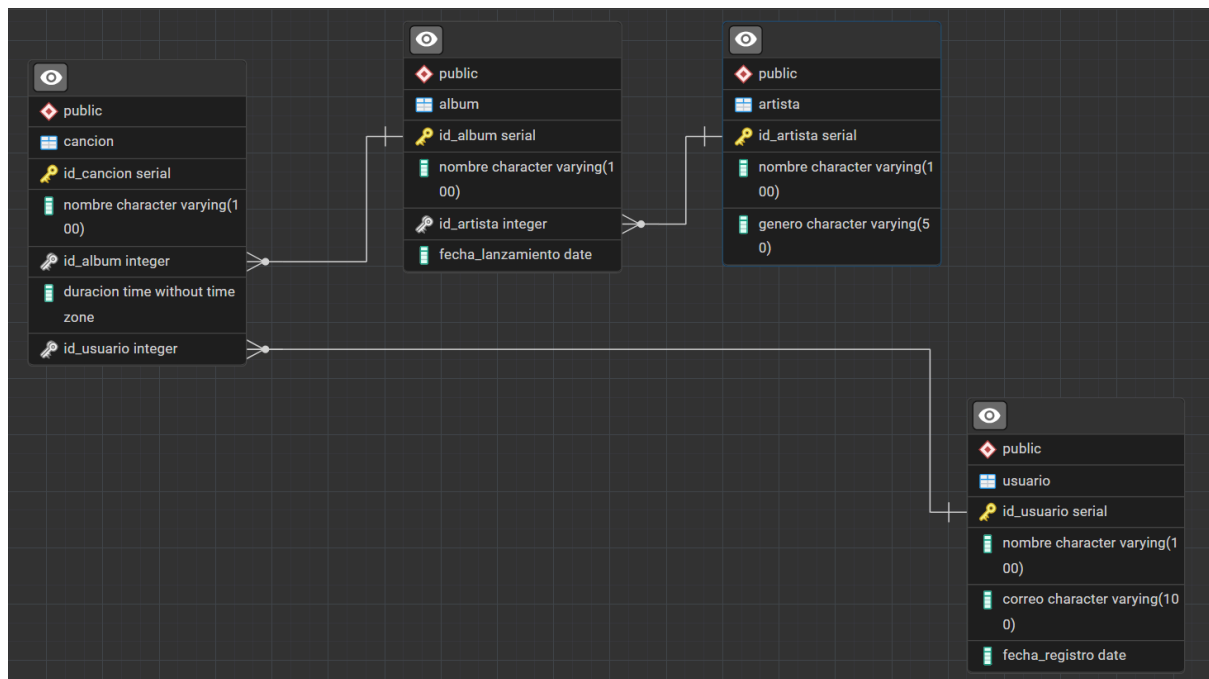
Para la gestión de eventos, se identifican las siguientes entidades clave:

- **Usuario:** Representa a los usuarios registrados en la plataforma. Tiene atributos como ID, nombre, correo electrónico y fecha de registro.
- **Artista:** Representa a los músicos o bandas. Contiene atributos como ID, nombre y género musical.
- **Álbum:** Representa los discos publicados por los artistas. Contiene atributos como ID, nombre, ID del artista y fecha de lanzamiento.
- **Canción:** Representa las canciones incluidas en los álbumes. Contiene atributos como ID, nombre, ID del álbum y duración.

Las relaciones entre estas entidades son las siguientes:

- Un **artista** puede tener varios álbumes, pero un álbum pertenece a un solo artista.
- Un **álbum** contiene varias canciones, pero una canción pertenece a un solo álbum.

2. Diseño del Modelo Conceptual (Diagrama E-R)



3. Conversión a Esquema Relacional

A partir del modelo E-R, se definen las siguientes tablas:

Usuario (ID_Usuario, Nombre, Correo, Fecha_Registro)

Artista (ID_Artista, Nombre, Género)

Álbum (ID_Álbum, Nombre, ID_Artista, Fecha_Lanzamiento)

ID_Artista es clave foránea que referencia a Artista(ID_Artista).

Canción (ID_Canción, Nombre, ID_Álbum, Duración)

ID_Álbum es clave foránea que referencia a Álbum(ID_Álbum).

4. Implementación en SQL

Creación de Tablas

Inserción de datos

5. Consultas SQL

Consulta requerida: Listar las canciones reproducidas por un usuario específico, incluyendo el nombre del artista y del álbum.

```
-- Listar las canciones reproducidas por un usuario específico, incluyendo el nombre del artista y del álbum
SELECT
    c.nombre AS cancion,
    a.nombre AS album,
    ar.nombre AS artista
FROM Cancion c
JOIN Album a ON c.id_album = a.id_album
JOIN Artista ar ON a.id_artista = ar.id_artista
ORDER BY ar.nombre, a.nombre, c.nombre;
```

Output Messages Notifications

cancion character varying (100)	album character varying (100)	artista character varying (100)
Vete	YHLQMDLG	Bad Bunny
La Bicicleta	Cumbiana	Carlos Vives
Mi Gente	Colores	J Balvin
A Dios le Pido	La Vida es un Ratico	Juanes
Hasta Hoy	Baila que Baila	Kany García
Despacito	Despacito	Luis Fonsi
Felices los 4	11:11	Maluma
Por Fin	Terral	Pablo Alborán
Fuiste Tú	Viaje al Centro del Corazón	Ricardo Arjona
Chantaje	El Dorado	Shakira

- Consulta SELECT

```
-- Recupera los nombres de todas las canciones de la tabla Cancion. (SELECT)
SELECT nombre FROM Cancion;
```

Output Messages Notifications

nombre character varying (100)
Chantaje
A Dios le Pido
La Bicicleta
Vete
Despacito
Felices los 4
Por Fin
Hasta Hoy
Fuiste Tú
Mi Gente

- Consulta WHERE

```
-- Filtra los resultados para mostrar solo las canciones que pertenecen al álbum con id_album = 1. (WHERE)
SELECT nombre
FROM Cancion WHERE id_album = 1;
```

Output Messages Notifications

nombre
Chantaje

- Consulta ORDER BY

```
-- Ordena las canciones por su duración en orden descendente (de mayor a menor duración). (ORDER BY)
SELECT nombre, duracion
FROM Cancion ORDER BY duracion DESC;
```

Output Messages Notifications

nombre	duracion
Despacito	04:41:00
Fuiste Tú	04:11:00
Hasta Hoy	04:10:00
Felices los 4	03:56:00
La Bicicleta	03:47:00
Por Fin	03:42:00
A Dios le Pido	03:39:00
Vete	03:21:00
Chantaje	03:11:00
Mi Gente	03:09:00

- Consulta GROUP BY

```
-- Agrupa las canciones por id_album y cuenta cuántas canciones tiene cada álbum. (GROUP BY)
SELECT id_album, COUNT(*) AS total_canciones
FROM Cancion GROUP BY id_album;
```

Output Messages Notifications

id_album integer	total_canciones bigint
9	1
3	1
5	1
4	1
10	1
6	1
2	1
7	1
1	1
8	1

- Consulta JOIN

```
-- Une las tablas Cancion y Album para mostrar el nombre de cada canción junto con el álbum al que pertenece. (JOIN)
SELECT c.nombre AS cancion, a.nombre AS album
FROM Cancion c
JOIN Album a ON c.id_album = a.id_album;
```

Output Messages Notifications

cancion character varying (100)	album character varying (100)
Chantajé	El Dorado
A Dios le Pido	La Vida es un Ratón
La Bicicleta	Cumbiana
Vete	YHLQMDLG
Despacito	Despacito
Felices los 4	11:11
Por Fin	Terral
Hasta Hoy	Baila que Baila
Fuiste Tú	Viaje al Centro del Corazón
Mi Gente	Colores

- Consulta LIKE

```
-- Busca usuarios cuyos nombres comiencen con la letra "J".
-- El % actúa como un comodín que permite cualquier combinación de caracteres después de la "J". (LIKE)
SELECT nombre FROM Usuario WHERE nombre LIKE 'J%';
```

Output Messages Notifications

nombre character varying (100)
Juan Pérez

4. Sistema de Control de Proyectos

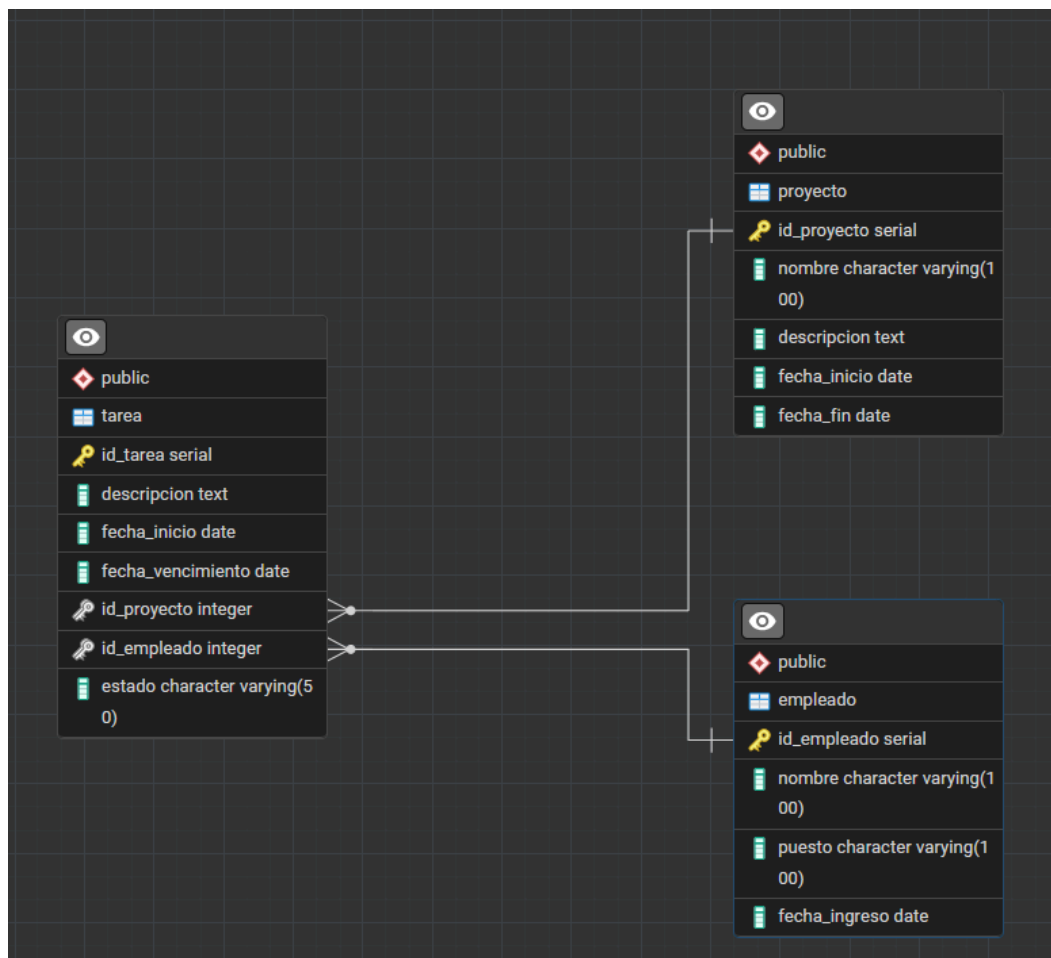
Una empresa desea hacer seguimiento de sus proyectos y tareas.

- **Empleado:** Representa a los trabajadores de la empresa. Tiene atributos como ID, nombre, puesto y fecha de ingreso.
- **Proyecto:** Representa los proyectos en desarrollo. Contiene atributos como ID, nombre, descripción, fecha de inicio y fecha de finalización.
- **Tarea:** Representa las actividades asignadas a los empleados dentro de un proyecto. Tiene atributos como ID, descripción, fecha de inicio, fecha de vencimiento, estado, ID del proyecto y ID del empleado.

Las relaciones entre estas entidades son las siguientes:

- Un **empleado** puede estar asignado a múltiples tareas, pero una tarea solo puede estar asignada a un empleado.
- Un **proyecto** puede tener varias tareas, pero cada tarea pertenece a un solo proyecto.
- Un **proyecto** puede estar relacionado con múltiples empleados a través de las tareas.

2. Diseño del Modelo Conceptual (Diagrama E-R)



3. Conversión a Esquema Relacional

A partir del modelo E-R, se definen las siguientes tablas:

Empleado (id_empleado, nombre, puesto, fecha_ingreso)

Proyecto (id_proyecto, nombre, descripcion, fecha_inicio, fecha_fin)

Tarea (id_tarea, descripcion, fecha_inicio, fecha_vencimiento, id_proyecto, id_empleado, estado)

id_proyecto en la tabla Tarea referencia a Proyecto(id_proyecto).

id_empleado en la tabla Tarea referencia a Empleado(id_empleado).

4. Implementación en SQL

Creación de Tablas

```
-- Crear tabla de Empleados
CREATE TABLE Empleado (
    id_empleado SERIAL PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    puesto VARCHAR(100) NOT NULL,
    fecha_ingreso DATE NOT NULL
);

-- Crear tabla de Proyectos
CREATE TABLE Proyecto (
    id_proyecto SERIAL PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    descripcion TEXT,
    fecha_inicio DATE NOT NULL,
    fecha_fin DATE
);

-- Crear tabla de Tareas
CREATE TABLE Tarea (
    id_tarea SERIAL PRIMARY KEY,
    descripcion TEXT NOT NULL,
    fecha_inicio DATE NOT NULL,
    fecha_vencimiento DATE NOT NULL,
    id_proyecto INT NOT NULL,
    id_empleado INT NOT NULL,
    estado VARCHAR(50) NOT NULL, -- Pendiente, En progreso, Completada
    FOREIGN KEY (id_proyecto) REFERENCES Proyecto(id_proyecto),
    FOREIGN KEY (id_empleado) REFERENCES Empleado(id_empleado)
);
```

Inserción de datos

```
-- Insertar datos en Empleados
INSERT INTO Empleado (nombre, puesto, fecha_ingreso) VALUES
('Pedro Sánchez', 'Ingeniero de Software', '2021-02-20'),
('Lucía Fernández', 'Desarrolladora Frontend', '2022-06-05'),
('Carlos Pérez', 'Director de Proyectos', '2020-03-18'),
('Laura López', 'Diseñadora UI/UX', '2019-11-25'),
('Martín Díaz', 'Desarrollador Backend', '2021-08-30'),
('Sofía García', 'Especialista en QA', '2022-10-14'),
('Javier Ramírez', 'Analista de Datos', '2021-05-17'),
('Ana Torres', 'Ingeniera de Sistemas', '2022-07-21'),
('Raúl Martínez', 'Líder de Proyecto', '2020-09-09'),
('Eva Gómez', 'Desarrolladora Full Stack', '2021-03-12');










-- Insertar datos en Proyectos
INSERT INTO Proyecto (nombre, descripcion, fecha_inicio, fecha_fin) VALUES
('Sistema de Gestión Empresarial', 'Desarrollo de software ERP para pequeñas empresas', '2023-04-15', '2023-10-15'),
('Aplicación Móvil de Compras', 'Aplicación para compras online con integración de pago', '2023-03-01', '2023-08-15'),
('Portal Web de Servicios Financieros', 'Rediseño de plataforma de servicios bancarios en línea', '2023-02-10', '2023-07-10'),
('Automatización de Procesos Administrativos', 'Implementación de RPA en procesos de la empresa', '2023-05-05', '2023-11-01'),
('Plataforma de Streaming de Video', 'Desarrollo de plataforma de transmisión de contenidos en vivo', '2023-01-05', '2023-12-15'),
('Sistema de Inteligencia Artificial para Soporte', 'Desarrollo de IA para mejorar el servicio al cliente', '2023-06-15', '2024-06-15'),
('Plataforma para Cursos Online', 'Desarrollo de plataforma de educación en línea con tutoría', '2023-07-01', '2023-12-01'),
('Aplicación de Realidad Aumentada para Compras', 'Desarrollo de app móvil con RA para visualizar productos', '2023-08-01', '2024-02-01'),
('Sistema de Gestión de Inventarios', 'Desarrollo de un sistema de inventarios para pequeñas empresas', '2023-01-01', '2023-09-30'),
('Transformación Digital y Migración a la Nube', 'Migración de infraestructuras locales a soluciones en la nube', '2023-03-01', '2023-10-31');

-- Insertar datos en Tareas
INSERT INTO Tarea (descripcion, fecha_inicio, fecha_vencimiento, id_proyecto, id_empleado, estado) VALUES
('Definir arquitectura del sistema', '2023-04-20', '2023-04-30', 1, 1, 'Pendiente'),
('Diseñar la interfaz de usuario', '2023-03-10', '2023-03-20', 2, 4, 'Pendiente'),
('Configurar base de datos', '2023-02-15', '2023-03-01', 3, 2, 'En progreso'),
('Crear módulos de pagos', '2023-04-01', '2023-04-10', 4, 3, 'Pendiente'),
('Realizar pruebas de rendimiento', '2023-05-10', '2023-05-20', 5, 6, 'Completada'),
('Implementar API de integración', '2023-06-10', '2023-06-25', 6, 5, 'Pendiente'),
('Automatizar integración continua', '2023-07-10', '2023-07-20', 7, 9, 'Pendiente'),
('Desarrollar funciones de búsqueda', '2023-08-05', '2023-08-18', 8, 7, 'Pendiente'),
('Actualizar documentación del sistema', '2023-09-10', '2023-09-25', 9, 8, 'Pendiente'),
('Migrar servidores a la nube', '2023-10-05', '2023-10-12', 10, 10, 'En progreso'),
('Auditar seguridad del sistema', '2025-03-25', '2025-04-05', 1, 1, 'Pendiente');
```

5. Consultas SQL

Consulta requerida: Mostrar todas las tareas pendientes de un proyecto específico, ordenadas por fecha de vencimiento

```
-- Mostrar todas las tareas pendientes de un proyecto específico, ordenadas por fecha de vencimiento
SELECT
    t.descripcion AS tarea,
    t.fecha_vencimiento,
    p.nombre AS proyecto,
    e.nombre AS empleado
FROM Tarea t
JOIN Proyecto p ON t.id_proyecto = p.id_proyecto
JOIN Empleado e ON t.id_empleado = e.id_empleado
WHERE p.id_proyecto = 1 -- Cambiar el ID de proyecto según sea necesario
AND t.estado = 'Pendiente'
ORDER BY t.fecha_vencimiento ASC;
```

Output		Messages	Notifications
			
			
			
tarea	fecha_vencimiento	proyecto	empleado
text	date	character varying (100)	character varying (100)
Definir arquitectura del sistema	2023-04-30	Sistema de Gestión Empresarial	Pedro Sánchez
Auditar seguridad del sistema	2025-04-05	Sistema de Gestión Empresarial	Pedro Sánchez







- **Consulta SELECT**


```
--Esta consulta selecciona todas las filas de la tabla Tarea donde el estado de la tarea es 'Pendiente'. (SELECT)
SELECT * FROM Tarea WHERE estado = 'Pendiente';
```

[illegible]

- **Consulta WHERE**

```
--Esta consulta selecciona todas las filas de la tabla Proyecto donde la fecha de inicio es mayor o igual a '2023-01-01'. (WHERE)
SELECT * FROM Proyecto WHERE fecha_inicio >= '2023-01-01';
```

     				
id_proyecto [PK] integer	nombre character varying (100)	descripcion text	fecha_inicio date	fecha_fin date
1	Sistema de Gestión Empresarial	Desarrollo de software ERP para pequeñas empresas	2023-04-15	2023-10-15
2	Aplicación Móvil de Compras	Aplicación para compras online con integración de pago	2023-03-01	2023-08-15
3	Portal Web de Servicios Financieros	Rediseño de plataforma de servicios bancarios en línea	2023-02-10	2023-07-10
4	Automatización de Procesos Administrativos	Implementación de RPA en procesos de la empresa	2023-05-05	2023-11-01
5	Plataforma de Streaming de Video	Desarrollo de plataforma de transmisión de contenidos en vivo	2023-01-05	2023-12-15
6	Sistema de Inteligencia Artificial para Soporte	Desarrollo de IA para mejorar el servicio al cliente	2023-06-15	2024-06-15
7	Plataforma para Cursos Online	Desarrollo de plataforma de educación en línea con tutoría	2023-07-01	2023-12-01
8	Aplicación de Realidad Aumentada para Co...	Desarrollo de app móvil con RA para visualizar productos	2023-08-01	2024-02-01
9	Sistema de Gestión de Inventarios	Desarrollo de un sistema de inventarios para pequeñas empresas	2023-01-01	2023-09-30
10	Transformación Digital y Migración a la Nube	Migración de infraestructuras locales a soluciones en la nube	2023-03-01	2023-10-31

- **Consulta ORDER BY**

```
95
96 --Esta consulta selecciona el nombre y la fecha de inicio de todos los proyectos y los ordena por fecha de inicio en orden descendente. (ORDER BY)
97 SELECT nombre, fecha_inicio FROM Proyecto ORDER BY fecha_inicio DESC;
98
```

	nombre character varying (100)	fecha_inicio date
1	Aplicación de Realidad Aumentada para Co...	2023-08-01
2	Plataforma para Cursos Online	2023-07-01
3	Sistema de Inteligencia Artificial para Soporte	2023-06-15
4	Automatización de Procesos Administrativos	2023-05-05
5	Sistema de Gestión Empresarial	2023-04-15
6	Aplicación Móvil de Compras	2023-03-01
7	Transformación Digital y Migración a la Nube	2023-03-01
8	Portal Web de Servicios Financieros	2023-02-10
9	Plataforma de Streaming de Video	2023-01-05
10	Sistema de Gestión de Inventarios	2023-01-01

- **Consulta GROUP BY**

```
--Agrupa los resultados por el id del proyecto, mostrando cuántas tareas están asociadas a cada proyecto. (GROUP BY)
SELECT id_proyecto, COUNT(*) AS tareas_totales FROM Tarea GROUP BY id_proyecto;
```

Output Messages Notifications

id_proyecto integer	tareas_totales bigint
9	1
3	1
5	1
4	1
10	1
6	1
2	1
7	1
1	2
8	1

- Consulta JOIN

```
--Esta consulta realiza una unión entre las tablas Tarea y Empleado usando el id del empleado para relacionarlas. (JOIN)
SELECT e.nombre, t.descripcion
FROM Tarea t
JOIN Empleado e ON t.id_empleado = e.id_empleado
WHERE t.estado = 'Pendiente';
```

Data Output Messages Notifications

nombre character varying (100)	descripcion text
Pedro Sánchez	Auditar seguridad del sistema
Pedro Sánchez	Definir arquitectura del sistema
Carlos Pérez	Crear módulos de pagos
Laura López	Diseñar la interfaz de usuario
Martín Díaz	Implementar API de integración
Javier Ramírez	Desarrollar funciones de búsqueda
Ana Torres	Actualizar documentación del sistema
Raúl Martínez	Automatizar integración continua

- Consulta LIKE

```
--Esta consulta selecciona todos los proyectos cuyo nombre comienza con 'Sistema'. (LIKE)
SELECT * FROM Proyecto WHERE nombre LIKE 'Sistema%';
```

Data Output Messages Notifications

id_proyecto [PK] integer	nombre character varying (100)	descripcion text	fecha_inicio date	fecha_fin date
1	Sistema de Gestión Empresarial	Desarrollo de software ERP para pequeñas empresas	2023-04-15	2023-10-15
6	Sistema de Inteligencia Artificial para Soporte	Desarrollo de IA para mejorar el servicio al cliente	2023-06-15	2024-06-15
9	Sistema de Gestión de Inventarios	Desarrollo de un sistema de inventarios para pequeñas empresas	2023-01-01	2023-09-30

Sistema de Evaluación Académica

1.-Definición de Entidades y Relaciones Para la gestión de evaluación académica de una escuela:

Estudiante: Representa a los estudiantes registrados en el sistema educativo. Sus atributos incluyen un identificador único ID Estudiante que actúa como clave primaria, el Nombre del estudiante y la Fecha de nacimiento para identificar a cada estudiante de manera única en el sistema.

Curso: Representa los cursos ofrecidos en la institución. Esta entidad tiene como atributos el ID Curso como clave primaria, el Nombre del curso, y la Descripción del curso que detalla el contenido y objetivo del curso.

Profesor: Representa a los profesores que imparten los cursos. Los atributos de esta entidad son ID Profesor, que es la clave primaria, Nombre, que es el nombre del profesor, y Especialidad, que detalla el área de conocimiento o disciplina en la que el profesor se especializa.

Calificación: Representa las calificaciones obtenidas por los estudiantes en los cursos. Los atributos de esta entidad incluyen ID Calificación como clave primaria, además de las claves foráneas ID Estudiante, que hace referencia a la entidad Estudiante, ID Curso, que hace referencia a la entidad Curso, y ID Profesor, que hace referencia a la entidad Profesor.

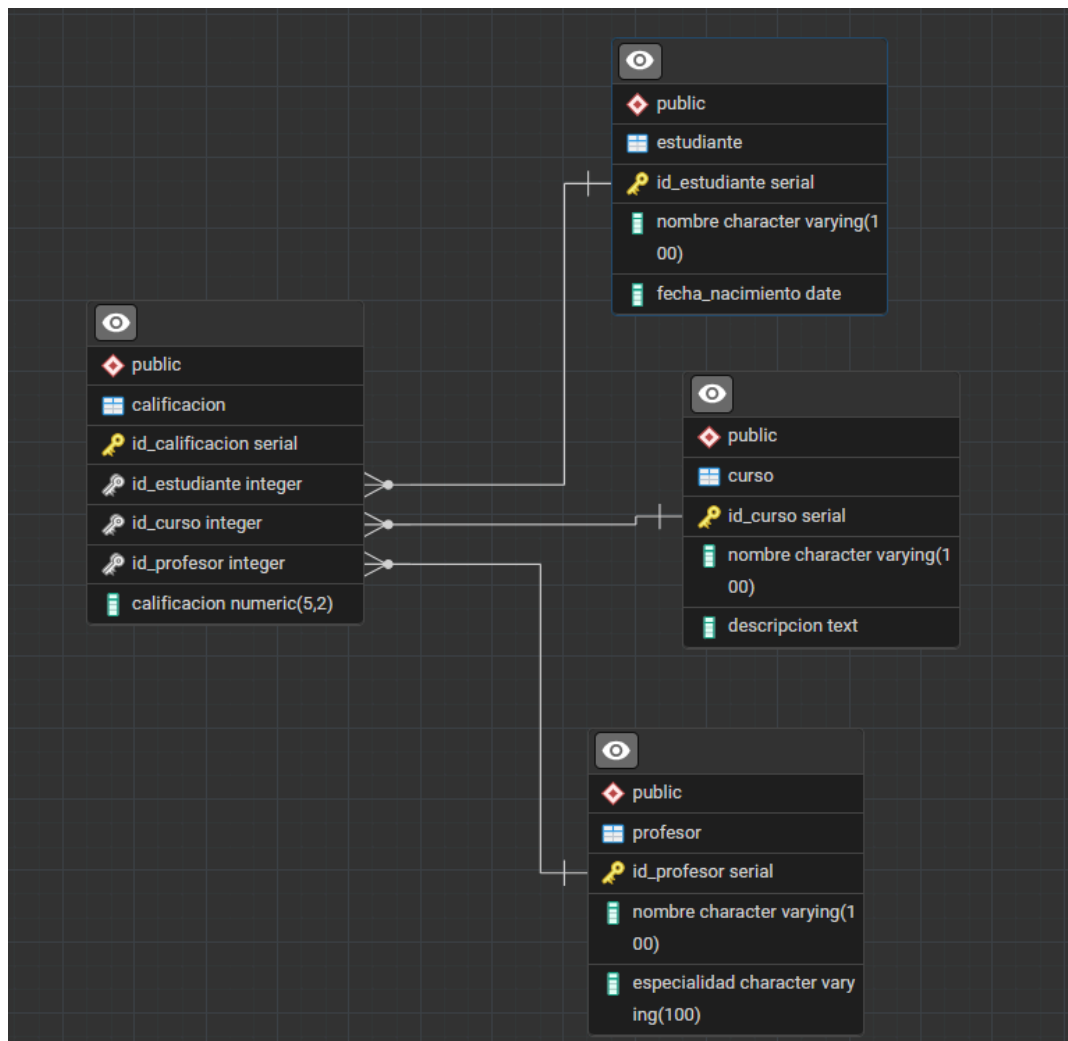
Las relaciones entre estas entidades son las siguientes:

Un **estudiante** puede tener varias calificaciones, pero cada calificación está asociada con un solo estudiante (relación uno a muchos).

Un **curso** puede tener varios estudiantes, y cada estudiante puede estar inscrito en varios cursos (relación muchos a muchos).

Un **profesor** puede impartir varios cursos y, por ende, evaluar a varios estudiantes, pero cada calificación está asociada con un único profesor (relación uno a muchos).

2.-Diseño del Modelo Conceptual (Diagrama E-R)



3.-Conversión a Esquema Relacional A partir del modelo E-R, se definen las siguientes tablas:

Estudiante

- **ID_estudiante** (clave primaria)
- Nombre
- Fecha_nacimiento

Curso

- **ID_curso** (clave primaria)
- Nombre
- Descripción

Profesor

- **ID_profesor** (clave primaria)
- Nombre
- Especialidad

Calificación

- **ID_calificacion** (clave primaria)
- **ID_estudiante** (clave foránea que referencia a Estudiante)
- **ID_curso** (clave foránea que referencia a Curso)
- **ID_profesor** (clave foránea que referencia a Profesor)
- Calificación (nota obtenida en el curso)

4.-Implementación en SQL

Creación de Tablas

```

-- Creacion de tablas

-- Crear tabla de Estudiantes
✓ CREATE TABLE Estudiante (
    id_estudiante SERIAL PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    fecha_nacimiento DATE NOT NULL
);

-- Crear tabla de Cursos
✓ CREATE TABLE Curso (
    id_curso SERIAL PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    descripcion TEXT
);

-- Crear tabla de Profesores
✓ CREATE TABLE Profesor (
    id_profesor SERIAL PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    especialidad VARCHAR(100)
);

-- Crear tabla de Calificaciones
✓ CREATE TABLE Calificacion (
    id_calificacion SERIAL PRIMARY KEY,
    id_estudiante INT NOT NULL,
    id_curso INT NOT NULL,
    id_profesor INT NOT NULL,
    calificacion DECIMAL(5,2) NOT NULL,
    FOREIGN KEY (id_estudiante) REFERENCES Estudiante(id_estudiante),
    FOREIGN KEY (id_curso) REFERENCES Curso(id_curso),
    FOREIGN KEY (id_profesor) REFERENCES Profesor(id_profesor)
);

```

Inserción de Datos

```

-- Insertar datos en Estudiantes
INSERT INTO Estudiante (nombre, fecha_nacimiento) VALUES
('Martín Díaz', '2000-09-10'),
('Lucía Fernández', '2001-05-22'),
('Carlos García', '2000-12-05'),
('Ana Ruiz', '2001-01-30'),
('David Martínez', '2000-11-15'),
('Elena Torres', '2001-07-09'),
('Ricardo López', '2000-03-25'),
('Laura Sánchez', '2001-02-18'),
('Pablo Fernández', '2000-08-04'),
('María González', '2001-04-12');

-- Insertar datos en Cursos
INSERT INTO Curso (nombre, descripcion) VALUES
('Matemáticas II', 'Curso avanzado de álgebra lineal y geometría analítica'),
('Física II', 'Estudio de la termodinámica y mecánica cuántica'),
('Química II', 'Avances en química orgánica e inorgánica'),
('Historia Contemporánea', 'Estudio de los acontecimientos mundiales desde el siglo XIX'),
('Literatura Española', 'Estudio de autores y movimientos literarios de España'),
('Programación Avanzada', 'Curso de desarrollo en Python y bases de datos'),
('Macroeconomía', 'Conceptos y teorías de la economía global'),
('Ecología', 'Estudio de los ecosistemas y su impacto ambiental'),
('Arte y Cultura', 'Análisis de la historia del arte y sus corrientes'),
('Geopolítica', 'Estudio de la política internacional y sus conflictos');

-- Insertar datos en Profesores
INSERT INTO Profesor (nombre, especialidad) VALUES
('Prof. Elena Martínez', 'Matemáticas'),
('Dr. Javier Gómez', 'Física'),
('Lic. Carmen Ruiz', 'Química'),
('Dr. Roberto López', 'Historia'),
('Prof. Susana García', 'Literatura'),
('Ing. Patricia Sánchez', 'Programación'),
('Lic. Mario Rodríguez', 'Economía'),
('Dr. Teresa Jiménez', 'Ecología'),
('Lic. Arturo Pérez', 'Arte'),
('Prof. Marcos Hernández', 'Geopolítica');

```

```
-- Insertar datos en Calificaciones
INSERT INTO Calificacion (id_estudiante, id_curso, id_profesor, calificacion) VALUES
(1, 1, 1, 7.8),
(1, 2, 2, 8.2),
(1, 3, 3, 6.9),
(1, 4, 4, 7.5),
(1, 5, 5, 8.1),
(2, 1, 1, 8.4),
(2, 2, 2, 7.3),
(2, 3, 3, 8.0),
(2, 4, 4, 6.7),
(2, 5, 5, 7.9),
(3, 1, 1, 6.5),
(3, 2, 2, 7.0),
(3, 3, 3, 8.3),
(3, 4, 4, 8.0),
(3, 5, 5, 9.1),
(4, 1, 1, 7.3),
(4, 2, 2, 6.6),
(4, 3, 3, 7.5),
(4, 4, 4, 8.1),
(4, 5, 5, 8.8),
(5, 1, 1, 9.0),
(5, 2, 2, 7.6),
(5, 3, 3, 7.8),
(5, 4, 4, 8.2),
(5, 5, 5, 9.3),
(6, 1, 1, 7.9),
(6, 2, 2, 9.2),
(6, 3, 3, 7.4),
(6, 4, 4, 8.4),
(6, 5, 5, 8.0),
(7, 1, 1, 8.2),
(7, 2, 2, 8.5),
(7, 3, 3, 7.6),
(7, 4, 4, 7.9),
(7, 5, 5, 8.3),
(8, 1, 1, 9.0),
(8, 2, 2, 7.2),
(8, 3, 3, 8.1),
(8, 4, 4, 7.4),
```

5. Consultas SQL

- **Consulta requerida:** Obtener el promedio de calificaciones de un estudiante en todos sus cursos.


```
-- Obtener el promedio de calificaciones de un estudiante en todos sus cursos
SELECT
    e.nombre AS estudiante,
    ROUND(AVG(c.calificacion), 2) AS promedio_calificacion
FROM Calificacion c
JOIN Estudiante e ON c.id_estudiante = e.id_estudiante
WHERE e.id_estudiante = 1 -- Cambiar el ID de estudiante según sea necesario
GROUP BY e.id_estudiante;
```

Output Messages Notifications

estudiante character varying (100)	promedio_calificacion numeric
Martín Díaz	7.90

- Consulta SELECT

```
-- Obtiene los datos de las calificaciones de un estudiante (SELECT)
SELECT
    e.nombre AS estudiante,
    c.calificacion,
    cu.nombre AS curso
FROM Calificacion c
JOIN Estudiante e ON c.id_estudiante = e.id_estudiante
JOIN Curso cu ON c.id_curso = cu.id_curso
WHERE e.id_estudiante = 1;
```

Output Messages Notifications

estudiante character varying (100)	calificacion numeric (5,2)	curso character varying (100)
Martín Díaz	7.80	Matemáticas II
Martín Díaz	8.20	Física II
Martín Díaz	6.90	Química II
Martín Díaz	7.50	Historia Contemporánea
Martín Díaz	8.10	Literatura Española
Martín Díaz	8.90	Programación Avanzada

- Consulta WHERE

```
-- Filtra los resultados para mostrar solo los estudiantes que tienen una calificación superior a 7 en un curso específico (WHERE)
SELECT
    e.nombre AS estudiante,
    cu.nombre AS curso,
    c.calificacion
FROM Calificacion c
JOIN Estudiante e ON c.id_estudiante = e.id_estudiante
JOIN Curso cu ON c.id_curso = cu.id_curso
WHERE c.calificacion > 7 AND cu.nombre = 'Física II';
```

estudiante	curso	calificacion
character varying (100)	character varying (100)	numeric (5,2)
Martín Díaz	Física II	8.20
Lucía Fernández	Física II	7.30
David Martínez	Física II	7.60
Elena Torres	Física II	9.20
Ricardo López	Física II	8.50
Laura Sánchez	Física II	7.20
Pablo Fernández	Física II	9.10
María González	Física II	8.40

- Consulta ORDER BY

```
-- Ordena los estudiantes por nombre y calificación (ORDER BY)
SELECT
    e.nombre AS estudiante,
    cu.nombre AS curso,
    c.calificacion
FROM Calificacion c
JOIN Estudiante e ON c.id_estudiante = e.id_estudiante
JOIN Curso cu ON c.id_curso = cu.id_curso
ORDER BY e.nombre, c.calificacion DESC;
```

estudiante	curso	calificacion
character varying (100)	character varying (100)	numeric (5,2)
Ana Ruiz	Literatura Española	8.80
Ana Ruiz	Historia Contemporánea	8.10
Ana Ruiz	Química II	7.50
Ana Ruiz	Matemáticas II	7.30
Ana Ruiz	Física II	6.60
Carlos García	Literatura Española	9.10
Carlos García	Química II	8.30
Carlos García	Historia Contemporánea	8.00
Carlos García	Física II	7.00
Carlos García	Matemáticas II	6.50

- Consulta GROUP BY

```
-- Obtiene el promedio de calificaciones por curso (GROUP BY)
SELECT
    cu.nombre AS curso,
    ROUND(AVG(c.calificacion), 2) AS promedio_calificacion
FROM Calificacion c
JOIN Curso cu ON c.id_curso = cu.id_curso
GROUP BY cu.id_curso
ORDER BY promedio_calificacion DESC;
```

Output Messages Notifications

curso character varying (100)	promedio_calificacion numeric
Programación Avanzada	8.90
Literatura Española	8.50
Matemáticas II	8.02
Historia Contemporánea	7.94
Física II	7.91
Química II	7.82

- Consulta JOIN

```
-- Obtiene el nombre del estudiante, curso y su calificación utilizando un JOIN entre las tablas
SELECT
    e.nombre AS estudiante,
    cu.nombre AS curso,
    c.calificacion
FROM Calificacion c
JOIN Estudiante e ON c.id_estudiante = e.id_estudiante
JOIN Curso cu ON c.id_curso = cu.id_curso;
```

Output Messages Notifications

estudiante character varying (100)	curso character varying (100)	calificacion numeric (5,2)
Martín Díaz	Matemáticas II	7.80
Martín Díaz	Física II	8.20
Martín Díaz	Química II	6.90
Martín Díaz	Historia Contemporánea	7.50
Martín Díaz	Literatura Española	8.10
Lucía Fernández	Matemáticas II	8.40
Lucía Fernández	Física II	7.30
Lucía Fernández	Química II	8.00
Lucía Fernández	Historia Contemporánea	6.70
Lucía Fernández	Literatura Española	7.90

- Consulta LIKE

```
-- Busca estudiantes cuyo nombre contenga la letra "a" (LIKE)
SELECT
    e.nombre AS estudiante,
    cu.nombre AS curso,
    c.calificacion
FROM Calificacion c
JOIN Estudiante e ON c.id_estudiante = e.id_estudiante
JOIN Curso cu ON c.id_curso = cu.id_curso
WHERE e.nombre LIKE '%a%';
```

Output Messages Notifications

estudiante character varying (100) 🔒	curso character varying (100) 🔒	calificacion numeric (5,2) 🔒
Martín Díaz	Matemáticas II	7.80
Martín Díaz	Física II	8.20
Martín Díaz	Química II	6.90
Martín Díaz	Historia Contemporánea	7.50
Martín Díaz	Literatura Española	8.10
Lucía Fernández	Matemáticas II	8.40
Lucía Fernández	Física II	7.30
Lucía Fernández	Química II	8.00
Lucía Fernández	Historia Contemporánea	6.70
Lucía Fernández	Literatura Española	7.90