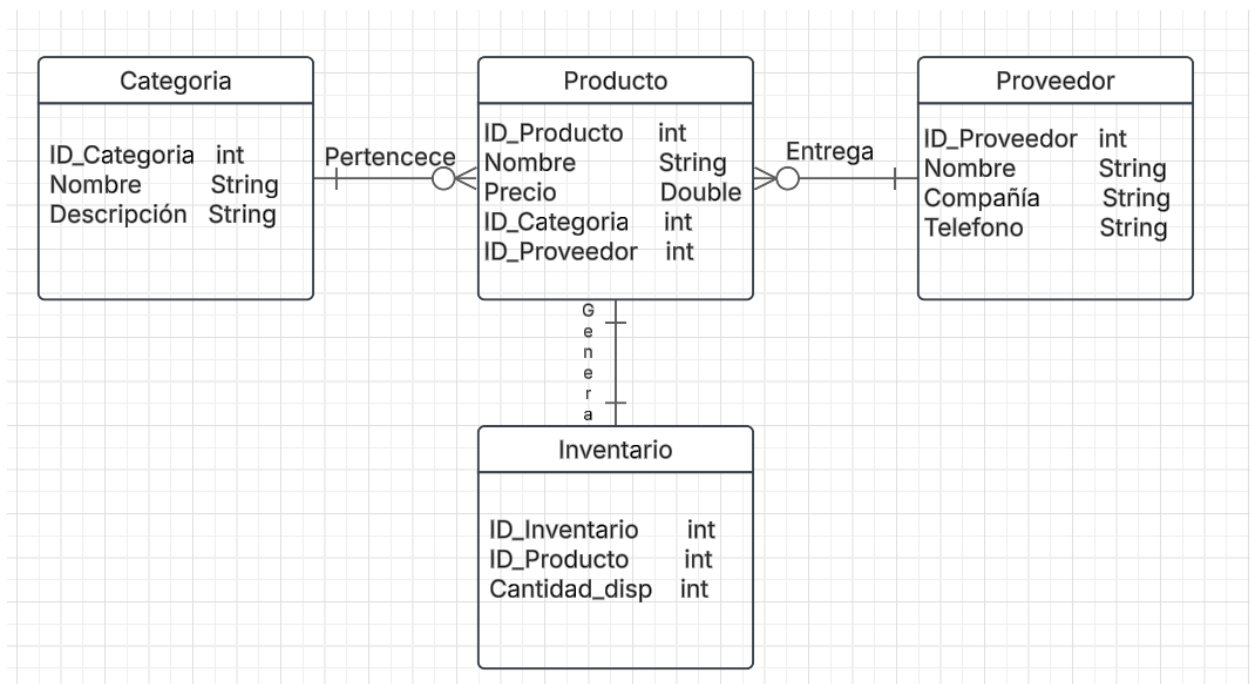


# 1. Sistema de Gestion de Inventarios

Una empresa desea controlar su inventario de productos y proveedores.

- Entidades clave: **Producto, Proveedor, Categoría, Inventario.**
- Diseñar Modelo E-R
- Convertir modelo a un esquema relacional.
- Implementar la base de datos en SQL.
- Obtener la lista de productos con sus respectivas categorías y proveedores ordenados alfabéticamente por nombre de producto

## MODELO E-R



## BASE DE DATOS

```
1  ✓ CREATE TABLE Proveedor
2  (
3  id_proveedor SERIAL PRIMARY KEY,
4  nombre VARCHAR(100) NOT NULL,
5  compañía VARCHAR(100) NOT NULL,
6  telefono VARCHAR(20) NOT NULL
7  );
8
9  ✓ CREATE TABLE Categoria
10 (
11 id_categoria SERIAL PRIMARY KEY,
12 nombre VARCHAR(100) NOT NULL,
13 descripcion VARCHAR(100) NOT NULL
14 );
15
16 ✓ CREATE TABLE Producto
17 (
18 id_producto SERIAL PRIMARY KEY,
19 id_categoria INT NOT NULL,
20 id_proveedor INT NOT NULL,
21 nombre VARCHAR(100) NOT NULL,
22 precio DECIMAL(10,2) NOT NULL,
23 FOREIGN KEY (id_categoria) REFERENCES Categoria(id_categoria) ON DELETE CASCADE,
24 FOREIGN KEY (id_proveedor) REFERENCES Proveedor(id_proveedor) ON DELETE CASCADE
25 );
26
27 ✓ CREATE TABLE Inventario
28 (
29 id_inventario SERIAL PRIMARY KEY,
30 id_producto INT NOT NULL UNIQUE,
31 cantidad_disp INT NOT NULL,
32 FOREIGN KEY (id_producto) REFERENCES Producto(id_producto) ON DELETE CASCADE
33 );
```

# CONSULTA

```
SELECT
p.nombre AS nombre_producto,
c.nombre AS nombre_categoria,
pr.nombre AS nombre_proveedor
FROM
Producto p
INNER JOIN
Categoria c ON p.id_categoria = c.id_categoria
INNER JOIN
Proveedor pr ON p.id_proveedor = pr.id_proveedor
ORDER BY
p.nombre ASC;
```

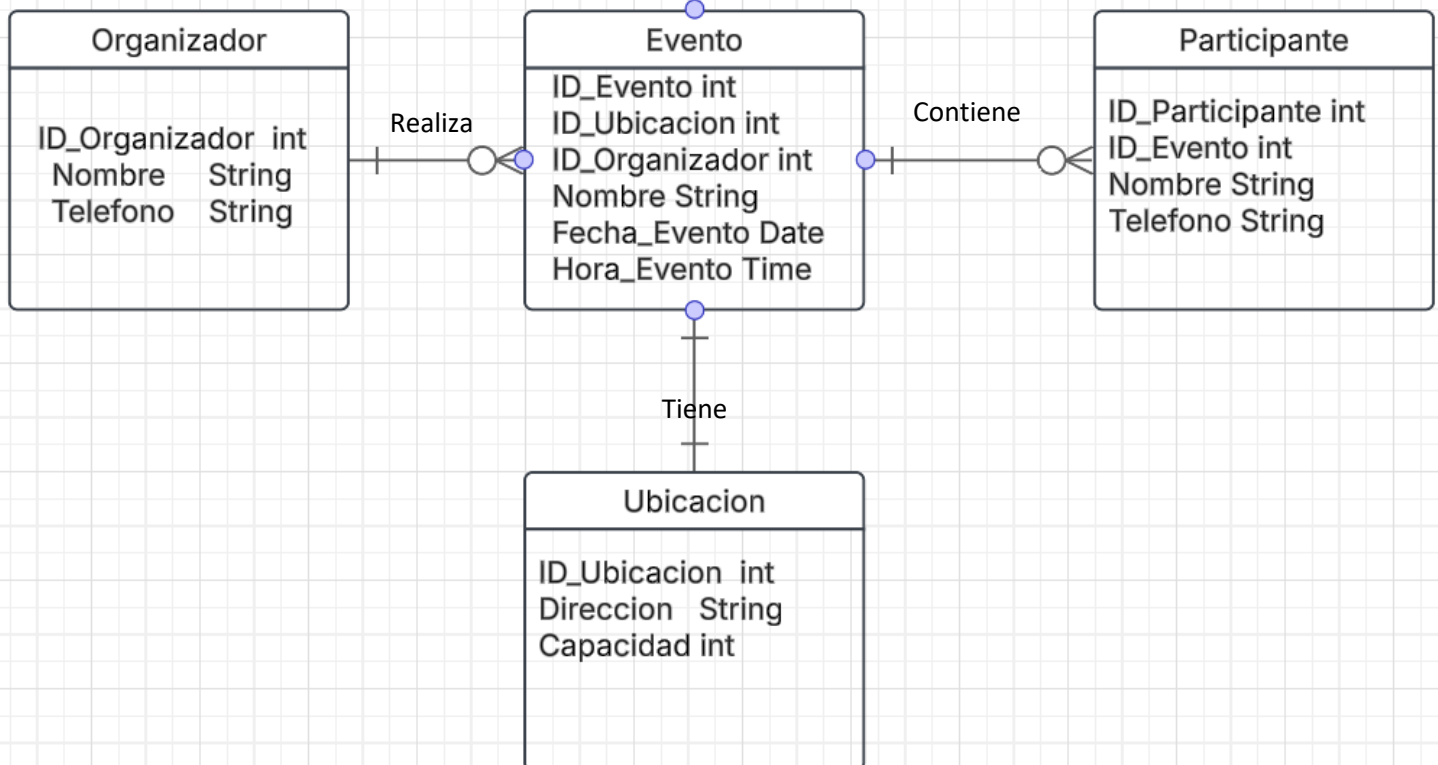
	nombre_producto character varying (100) 🔒	nombre_categoria character varying (100) 🔒	nombre_proveedor character varying (100) 🔒
1	Balón de fútbol americano	Deportes	Marta López
2	Camiseta Adidas	Ropa	Ana Gómez
3	Laptop HP	Electrónica	Juan Pérez
4	Muñeca Barbie	Juguetes	Luis Martínez
5	Olla de presión	Hogar	Carlos Rodríguez
6	Pantalones Levis	Ropa	Ana Gómez
7	Pelota de fútbol	Juguetes	Luis Martínez
8	Raqueta de tenis Wilson	Deportes	Marta López
9	Smartphone Samsung	Electrónica	Juan Pérez
10	Tetera eléctrica	Hogar	Carlos Rodríguez

## 2. Sistema de Gestión de Eventos

Una empresa de organización de eventos necesita administrar sus eventos y participantes.

- Entidades clave: **Evento**, **Participante**, **Ubicación**, **Organizador**
- Diseñar Modelo E-R.
- Crear el esquema relacional.
- Implementar la base de datos.
- Obtener la lista de eventos programados junto con la cantidad de participantes registrados por evento.

### MODELO E-R







## BASE DE DATOS

```
1  ✓ CREATE TABLE Ubicacion (  
2    id_ubicacion SERIAL PRIMARY KEY,  
3    direccion VARCHAR(100) UNIQUE NOT NULL,  
4    capacidad INT NOT NULL  
5  );  
6  
7  ✓ CREATE TABLE Organizador (  
8    id_organizador SERIAL PRIMARY KEY,  
9    nombre VARCHAR(100) NOT NULL,  
10   telefono VARCHAR(20) UNIQUE NOT NULL  
11  );  
12  
13  
14  ✓ CREATE TABLE Evento (  
15    id_evento SERIAL PRIMARY KEY,  
16    id_ubicacion INT UNIQUE NOT NULL,  
17    id_organizador INT NOT NULL,  
18    nombre VARCHAR(100),  
19    fecha_evento DATE NOT NULL,  
20    hora_evento TIME NOT NULL,  
21    FOREIGN KEY (id_ubicacion) REFERENCES Ubicacion(id_ubicacion),  
22    FOREIGN KEY (id_organizador) REFERENCES Organizador(id_organizador)  
23  );  
24  
25  
26  ✓ CREATE TABLE Participante (  
27    id_participante SERIAL PRIMARY KEY,  
28    id_evento INT NOT NULL,  
29    nombre VARCHAR(100) NOT NULL,  
30    telefono VARCHAR(20) UNIQUE NOT NULL,  
31    FOREIGN KEY (id_evento) REFERENCES Evento(id_evento) ON DELETE CASCADE  
32  );  
33
```

# CONSULTA

```
94  SELECT
95  E.nombre AS evento,
96  COUNT(P.id_participante) AS participantes
97  FROM
98  Evento E
99  LEFT JOIN
100  Participante P ON E.id_evento = P.id_evento
101  GROUP BY
102  E.id_evento, E.nombre
103  ORDER BY
104  E.fecha_evento;
```

Data Output Messages Notifications



Showing rows: 1 to 5 

Page

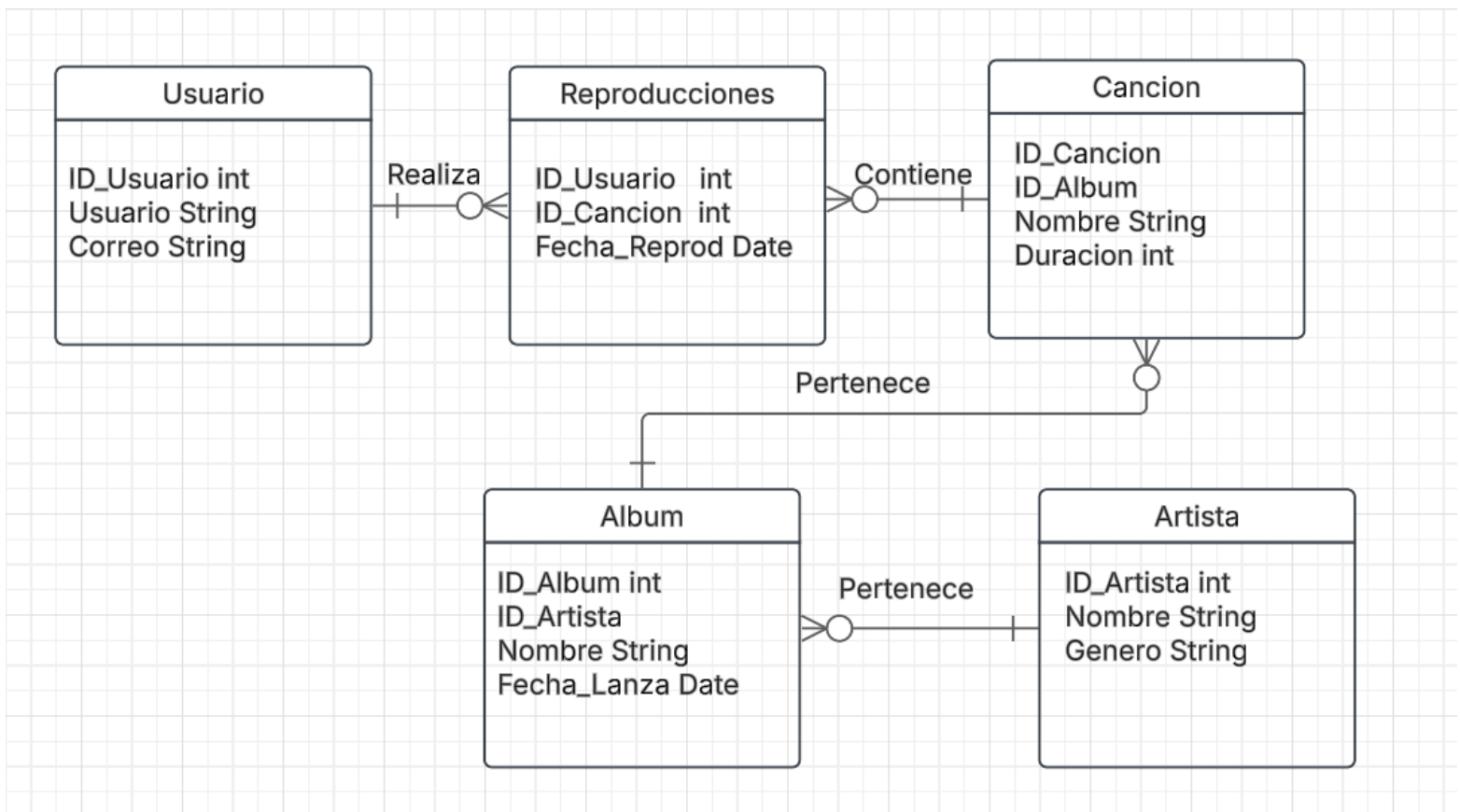
	evento character varying (100) 	participantes bigint 
1	Concierto de Rock	10
2	Feria de Tecnología	10
3	Conferencia de Marketing	10
4	Exposición de Arte	2
5	Festival Gastronómico	5

### 3. Plataforma de Streaming de Musica

Se desea un sistema para administrar usuarios, artistas y canciones

- Entidades clave: **Usuario, Artista, Álbum, Cancion.**
- Diseñar un Modelo E-R.
- Transformarlo en un esquema relacional.
- Implementar la base de datos.
- Listas las canciones reproducidas por un usuario específico, incluyendo el nombre del artista y álbum.

#### MODELO E-R



## BASE DE DATOS

Query Query History



```
1  ✓ CREATE TABLE Usuario (  
2    id_usuario SERIAL PRIMARY KEY,  
3    usuario VARCHAR(100) UNIQUE NOT NULL,  
4    correo VARCHAR(100) UNIQUE NOT NULL  
5  );  
6  
7  ✓ CREATE TABLE Artista (  
8    id_artista SERIAL PRIMARY KEY,  
9    nombre VARCHAR(100) UNIQUE NOT NULL,  
10   genero VARCHAR(50) NOT NULL  
11  );  
12  
13  ✓ CREATE TABLE Album (  
14    id_album SERIAL PRIMARY KEY,  
15    nombre VARCHAR(100) NOT NULL,  
16    fecha_lanzamiento DATE NOT NULL,  
17    id_artista INT NOT NULL,  
18    FOREIGN KEY (id_artista) REFERENCES Artista(id_artista) ON DELETE CASCADE  
19  );  
20  
21  ✓ CREATE TABLE Cancion (  
22    id_cancion SERIAL PRIMARY KEY,  
23    nombre VARCHAR(100) NOT NULL,  
24    duracion INT NOT NULL,  
25    id_album INT NOT NULL,  
26    FOREIGN KEY (id_album) REFERENCES Album(id_album) ON DELETE CASCADE  
27  );  
28  
29  ✓ CREATE TABLE Reproduccion (  
30    id_usuario INT NOT NULL,  
31    id_cancion INT NOT NULL,  
32    fecha_reproduccion DATE NOT NULL,  
33    PRIMARY KEY (id_usuario, id_cancion),  
34    FOREIGN KEY (id_usuario) REFERENCES Usuario(id_usuario),  
35    FOREIGN KEY (id_cancion) REFERENCES Cancion(id_cancion)  
36  );  
37
```



## CONSULTA

```
127 ✓ SELECT
128 u.usuario,
129 c.nombre AS cancion,
130 a.nombre AS album,
131 ar.nombre AS artista
132 FROM
133 Reproduccion r
134 INNER JOIN
135 Usuario u ON r.id_usuario = u.id_usuario
136 INNER JOIN
137 Cancion c ON r.id_cancion = c.id_cancion
138 INNER JOIN
139 Album a ON c.id_album = a.id_album
140 INNER JOIN
141 Artista ar ON a.id_artista = ar.id_artista
142 WHERE
143 u.usuario = 'john_doe'
```

Data Output   Messages   Notifications

Showing rows: 1 to 2

	usuario character varying (100) 🔒	cancion character varying (100) 🔒	album character varying (100) 🔒	artista character varying (100) 🔒
1	john_doe	No Tears Left to Cry	Sweetener	Ariana Grande
2	john_doe	Blinking Lights	After Hours	The Weeknd

## 4. Sistema de Control de Proyectos

Una empresa desea hacer seguimiento de sus proyectos y tareas.

- Entidades clave: **Proyecto, Empleado, Tarea.**
- Diseñar el diagrama E-R.
- Transformarlo en tablas relacionales.
- Implementar la base de datos.
- Mostrar todas las tareas pendientes de un proyecto específico, ordenadas por fecha de vencimiento.

### MODELO E-R



## BASE DE DATOS

```
1  ✓ CREATE TABLE Proyecto (  
2    id_proyecto SERIAL PRIMARY KEY,  
3    nombre VARCHAR(100) NOT NULL,  
4    fecha_inicio DATE NOT NULL,  
5    fecha_finalizacion DATE NOT NULL  
6  );  
7  
8  ✓ CREATE TABLE Empleado (  
9    id_empleado SERIAL PRIMARY KEY,  
10   nombre VARCHAR(100) NOT NULL,  
11   num_empleado INT UNIQUE NOT NULL,  
12   puesto VARCHAR(50) NOT NULL,  
13   fecha_ingreso DATE NOT NULL  
14  );  
15  
16  ✓ CREATE TABLE Tarea (  
17   id_tarea SERIAL PRIMARY KEY,  
18   descripcion TEXT NOT NULL,  
19   vencimiento DATE NOT NULL,  
20   estado VARCHAR(50) NOT NULL,  
21   id_proyecto INT NOT NULL,  
22   id_empleado INT NOT NULL,  
23   FOREIGN KEY (id_proyecto) REFERENCES Proyecto(id_proyecto),  
24   FOREIGN KEY (id_empleado) REFERENCES Empleado(id_empleado)  
25  );
```

# CONSULTA

```
63  SELECT
64  P.nombre AS proyecto,
65  T.id_tarea,
66  T.vencimiento,
67  T.estado
68
69  FROM
70  Tarea T
71  INNER JOIN Proyecto P ON T.id_proyecto = P.id_proyecto
72  INNER JOIN Empleado E ON T.id_empleado = E.id_empleado
73  WHERE
74  T.estado = 'Pendiente'
75  AND P.id_proyecto = 5
76  ORDER BY
77  T.vencimiento;
```

Data Output Messages Notifications

SQL

Showing rows: 1 to 1

Page

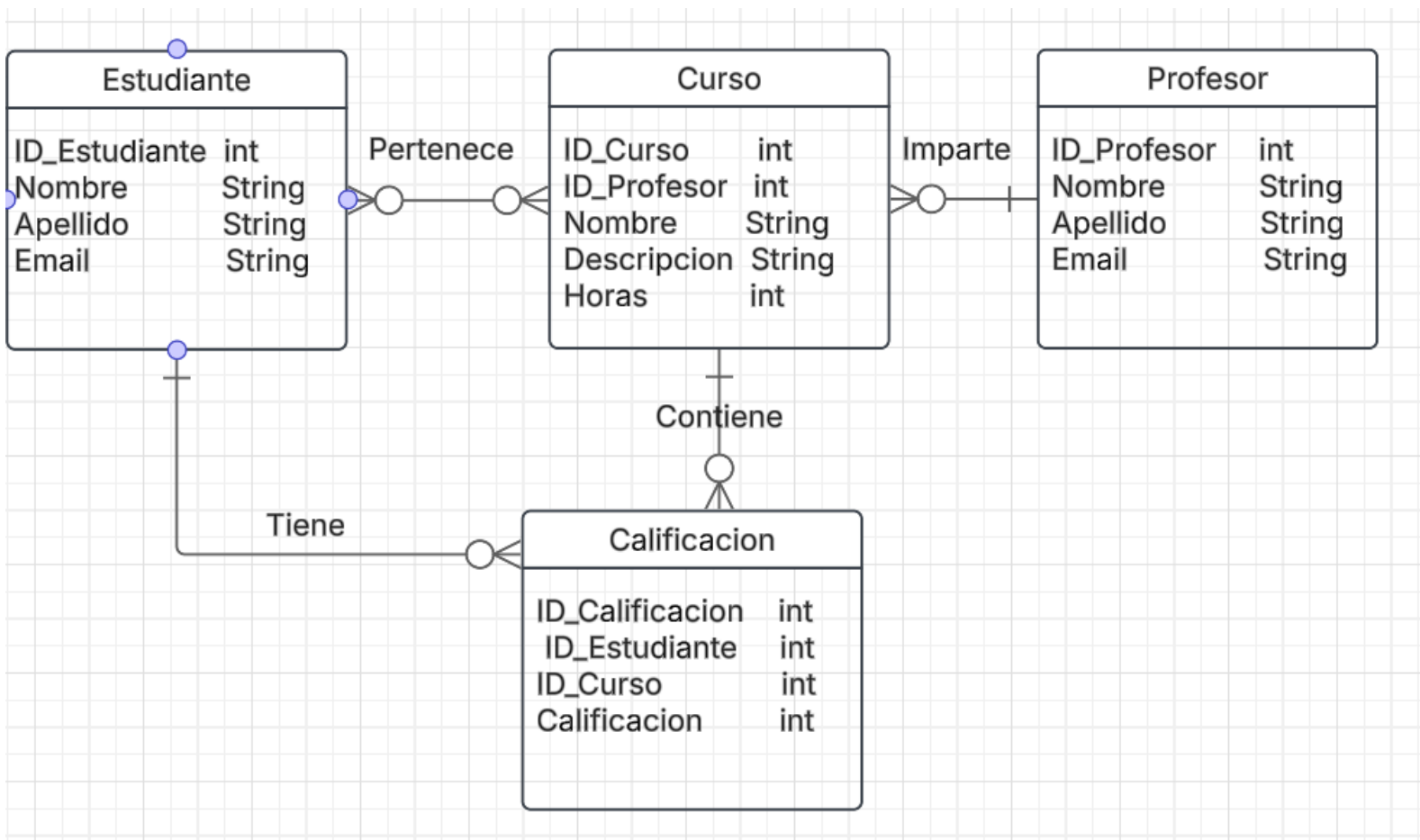
	proyecto character varying (100)	id_tarea integer	vencimiento date	estado character varying (50)
1	Desarrollo de Aplicación Móvil	5	2025-06-01	Pendiente

## 5. Sistema de Evaluación Académica

Una institución educativa necesita administrar las calificaciones de los estudiantes.

- Entidades clave: **Estudiante**, **Curso**, **Profesor**, **Calificación**.
- Elaborar el modelo E-R.
- Convertirlo a un esquema relacional.
- Implementar la base de datos.
- Obtener el promedio de calificaciones de un estudiante en todos sus cursos.

### MODELO E-R



## BASE DE DATOS

Query Query History



```
1  ✓ CREATE TABLE Estudiantes (  
2    id_estudiante SERIAL PRIMARY KEY,  
3    nombre VARCHAR(50) NOT NULL,  
4    apellido VARCHAR(50) NOT NULL,  
5    email VARCHAR(100) UNIQUE NOT NULL  
6  );  
7  
8  ✓ CREATE TABLE Profesores (  
9    id_profesor SERIAL PRIMARY KEY,  
10   nombre VARCHAR(50) NOT NULL,  
11   apellido VARCHAR(50) NOT NULL,  
12   email VARCHAR(100) UNIQUE NOT NULL  
13  );  
14  
15  ✓ CREATE TABLE Cursos (  
16    id_curso SERIAL PRIMARY KEY,  
17    nombre_curso VARCHAR(100) NOT NULL,  
18    descripcion TEXT NOT NULL,  
19    horas INT NOT NULL,  
20    id_profesor INT NOT NULL,  
21    FOREIGN KEY (id_profesor) REFERENCES Profesores(id_profesor)  
22  );  
23  
24  
25  ✓ CREATE TABLE Calificaciones (  
26    id_calificacion SERIAL PRIMARY KEY,  
27    id_estudiante INT NOT NULL,  
28    id_curso INT NOT NULL,  
29    calificacion INT CHECK (calificacion BETWEEN 0 AND 100),  
30    FOREIGN KEY (id_estudiante) REFERENCES Estudiantes(id_estudiante),  
31    FOREIGN KEY (id_curso) REFERENCES Cursos(id_curso)  
32  );  
33
```

CONSULTA

```
95 -- Mostrar el promedio de calificaciones de un estudiante en todos sus cursos
96 SELECT e.nombre, e.apellido, ROUND(AVG(c.calificacion), 2) AS promedio_calificacion
97 FROM Calificaciones c
98 INNER JOIN Estudiantes e ON c.id_estudiante = e.id_estudiante
99 GROUP BY e.nombre, e.apellido;
100
```

Data Output Messages Notifications

	nombre character varying (50)	apellido character varying (50)	promedio_calificacion numeric
1	Jorge	Jimenez	84.00
2	Beatriz	Hernandez	87.00
3	María	Lopez	87.00
4	Luis	Rodriguez	82.00
5	Pedro	Fernandez	80.50
6	Laura	Sánchez	84.50
7	Juan	Torres	85.50
8	Carlos	Gomez	87.33
9	Ana	Martinez	84.00
10	Marta	Diaz	91.00