

**Instituto Nacional de México Campu Mexicali**



Equipo:

**Donovan Hernandez Hernandez**

**Angel Hiram Velazquez Dueñas**

**Genesis Edel Ortiz Perez**

**unidad 1**

**Fundamentos de base de datos**

**Problemáticas Diagramas de Venn**

**Prof. Jose Ramon Bogarin Valenzuela**

**Lunes,24 de febrero del 2024**

## 2. Plataforma de Comercio Electrónico

### 1. Entidades principales del sistema:

- Usuarios
- Pedidos
- Productos

### 2. Atributos Clave del Sistema

- Usuario: ID\_usuario(PK), nombre completo, correo electrónico.
- Pedido: ID\_pedido(PK), ID\_producto(FK), nombre.
- Producto: ID\_producto(PK), nombre, tipo, precio, disponible.

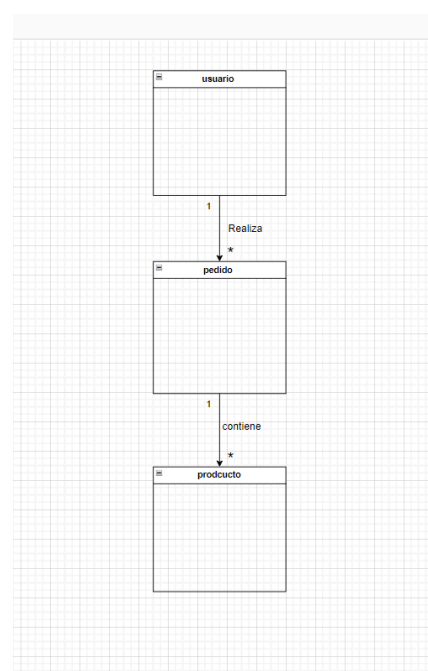
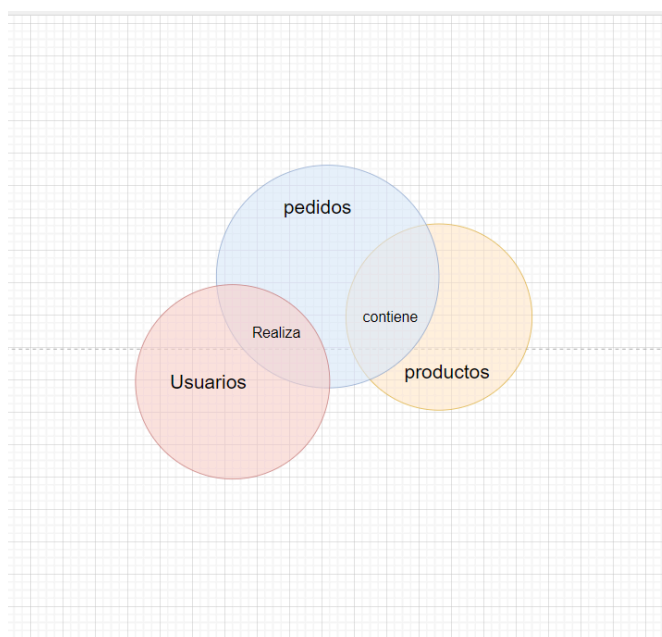
### 3. Relaciones entre entidades

- Muchos usuarios puede solicitar muchos pedidos y muchos productos (relación de M:M)
- Muchos productos pueden tener un solo pedido (relacion de M:1)
- Muchos pedidos pueden tener muchos usuarios (relacion de M:M)

### 4. Claves primarias para cada entidad

- Usuario: ID\_usuario
- Producto: ID\_producto
- Pedido: ID\_pedido (compuesta)

### 5. Diagrama de Venn



Consulta Historial de Consultas

```

1 SELECT * FROM public."Pedido"
2 ORDER BY "ID_pedido" ASC

```

Data Output Mensajes Notificaciones

ID_pedido	ID_producto	nombre
[PK] integer	integer	text

Consulta Historial de Consultas

```

1 SELECT * FROM public."Producto"
2 ORDER BY "ID_producto" ASC

```

Data Output Mensajes Notificaciones

ID_producto	nombre	tipo	precio	Disponibilidad
[PK] integer	text	text	integer	character varying

Consulta Historial de Consultas

```

1 SELECT * FROM public."Usuario"
2 ORDER BY "ID_usuario" ASC

```

Data Output Mensajes Notificaciones

ID_usuario	Nombre	Correo
[PK] integer	text	character varying

### 3. Sistema de Gestión Escolar

1. Las entidades principales del sistema son:

- ❖ Alumnos
- ❖ Cursos
- ❖ Inscripciones
- ❖ Profesores
- ❖ Calificaciones

2. Atributos Clave del Sistema:

- Alumnos: ID\_Alumno(PK), nombre, apellido, correo, fecha\_nacimiento.
- Cursos: ID\_curso(PK), nombre, descripción, créditos.

- Inscripciones: ID\_inscripcion(PK), ID\_alumno(FK), ID\_curso(FK), fecha\_inscripcion.
- Profesores: ID\_Profesor(PK), Nombre, Teléfono, especialidad.
- Calificaciones: ID\_calificacion(PK), ID\_estudiante(FK), ID\_materia(FK), ID\_profesor, calificacion.

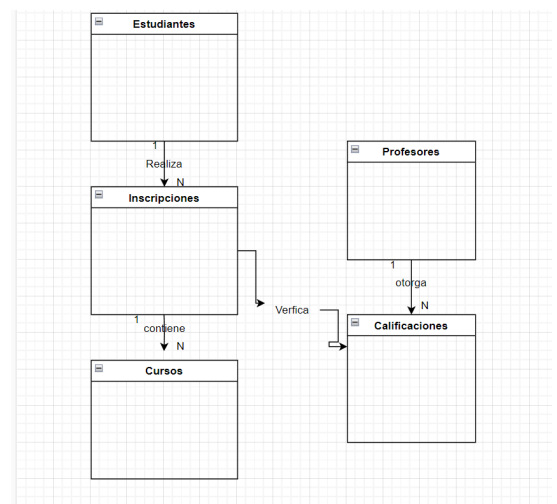
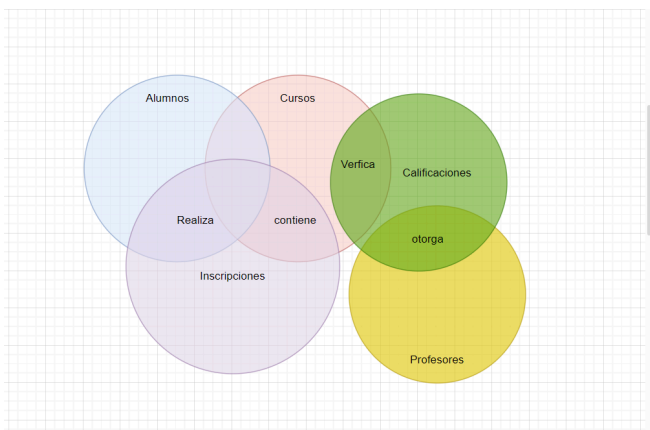
### 3. Relaciones entre entidades:

- Un alumno puede inscribirse en muchos cursos (la relación es 1:M entre alumnos e inscripciones).
- Un curso puede tener muchos alumnos inscritos (la relación es de 1:M entre cursos e inscripciones).
- un estudiante puede recibir muchas calificaciones, pero cada calificación pertenece a un solo estudiante (la relación es de N:1).
- Un profesor puede asignar muchas calificaciones, pero cada calificación es dada por un solo profesor.

### 4. Claves primarias para cada entidad:

- ☐ Alumno: ID\_alumno
- ☐ Cursos: ID\_Cursos
- ☐ Inscripciones: Esta tabla se usa para asociar a los alumnos con los cursos, en este caso la clave primaria es compuesta ya que es una combinación de ID\_alumno y ID\_curso
- ☐ profesores: ID\_profesor
- ☐ Calificaciones: ID\_calificacion

### 5. Diagrama de Venn y diagrama modelo E-R



Consulta

Historial de Consultas

1

SELECT \* FROM public."Profesor"

2

ORDER BY "ID\_Profesor" ASC

Data Output

Mensajes

Notificaciones

ID\_Profesor

[PK] integer

Nombre

text

Telefono

integer

Especialidad

text

Consulta

Historial de Consultas

1

SELECT \* FROM public."Inscripciones"

2

ORDER BY "ID\_curso" ASC

Data Output

Mensajes

Notificaciones

ID\_inscripcion

integer

Nombre

text

ID\_alumno

integer

Fecha\_inscripcion

date

ID\_curso

[PK] integer

Consulta

Historial de Consultas

1

SELECT \* FROM public."Curso"

2

ORDER BY "ID\_Curso" ASC

Data Output

Mensajes

Notificaciones

ID\_Curso

[PK] integer

Nombre

text

Creditos

integer

Consulta

Historial de Consultas

1

SELECT \* FROM public."Calificacion"

2

ORDER BY "ID\_calificacion" ASC

Data Output

Mensajes

Notificaciones

ID\_calificacion

[PK] integer

ID\_estudiante

integer

ID\_materia

integer

ID\_profesor

integer

Calificacion

integer

Consulta

Historial de Consultas

1

SELECT \* FROM public."Estudiante"

2

ORDER BY "ID" ASC

Data Output

Mensajes

Notificaciones

ID

[PK] integer

Nombre

text

Correo

character varying

Fecha\_Nacimiento

date

1

[null]

[null]

[null]

[null]

2

[null]

[null]

[null]

[null]

#### 4. Aplicación Mensajería

1. Las entidades principales de la aplicación son:

- ❖ Usuarios
- ❖ Mensajes
- ❖ Adjuntos

2. Atributos clave de cada entidad:

- ID\_usuario(PK), nombre\_usuario, Correo
- ID Mensaje(PK) ,contenido, fecha, tipo\_mensaje.
- ID\_Adjunto(PK) ,Nombre\_archivo, Tipo\_Archivo

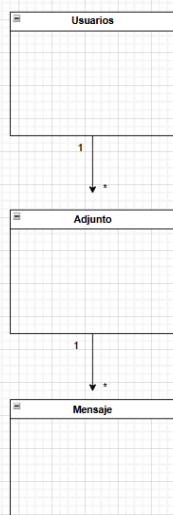
### 3. Relaciones entre entidades:

- Usuario(1)-->(N)Mensaje: Un usuario puede enviar varios mensajes, pero cada mensaje, solo es enviado por un usuario.
- Mensaje(1)-->(N)Adjunto: Un mensaje puede tener varios archivos adjuntos, pero un archivo adjunto pertenece a un solo mensaje.

4. Claves primarias de cada entidad:

- ☐ Usuario: ID\_usuario
- ☐ Mensaje: ID\_mensaje
- ☐ Adjunto: ID\_Adjunto

## 6. Diagrama de Venn y diagrama de modelo E-R



```
Consulta Historial de Consultas
1 SELECT * FROM public."Adjunto"
2 ORDER BY "ID_adjunto" ASC

Data Output Mensajes Notificaciones

ID_adjunto [PK] integer
Nombre_archivo text
Tipo_archivo bytea
```

Consulta	Historial de Consultas
1	SELECT * FROM public."Usuario"
2	ORDER BY "ID_usuario" ASC

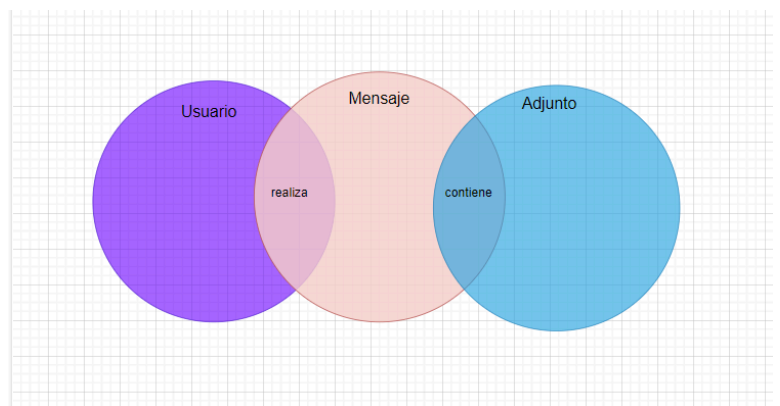
  

Data Output	Mensajes	Notificaciones
<div> <div>+</div> <div> <div>📄</div> <div>📋</div> <div>🗑️</div> <div>📊</div> <div>⬇️</div> <div>📈</div> </div> </div>		
ID_usuario [PK] integer	Nombre_usuario text	Correo character varying

Consulta	Historial de Consultas
1	SELECT * FROM public."Mensajes"
2	ORDER BY "id_Mensaje" ASC

Data Output	Mensajes	Notificaciones
<div> <div>+</div> <div> <div>📄</div> <div>📋</div> <div>🗑️</div> <div>📊</div> <div>⬇️</div> <div>📈</div> </div> </div>		
id_Mensaje [PK] integer	ID_Remitente integer	Fecha_envio date



## 5. Plataforma de streaming de música

Las entidades principales son:

- ❖ Usuarios
- ❖ Canciones
- ❖ Playlist
- ❖ Historial de reproducción

Atributos claves:

- Usuario: ID\_usuario(PK), nombre, Correo, contraseña.
- Canción: ID\_cancion(PK), Género, artista, fecha\_subida.

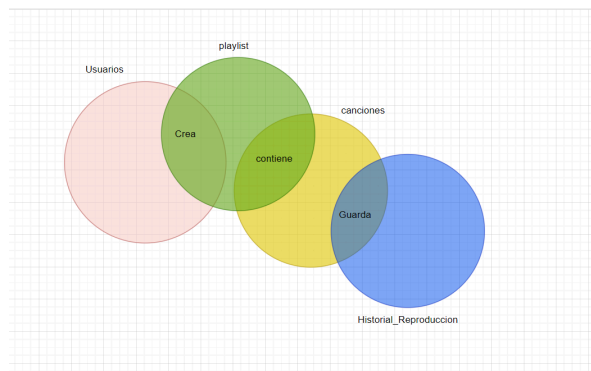
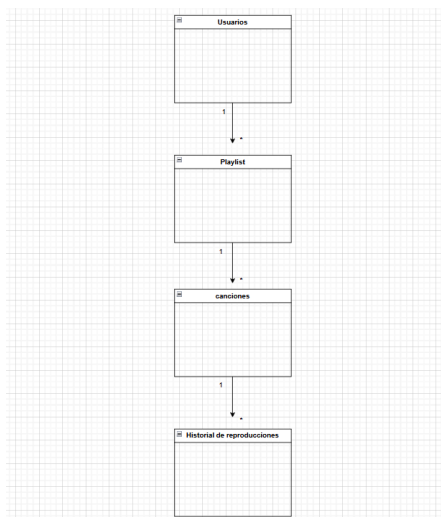
- Playlist: ID\_playlist(PK), Fecha\_creacion, nombre.
- Historial\_reproducciones: ID\_historial(PK), ID\_usuario(FK), ID\_cancion(FK), fecha\_hora.

### Relaciones entre entidades

- Usuarios - Playlists: 1 a N: Un usuario puede tener muchas playlist
- Playlists - Canciones: 1 a N: 1 playlist puede tener muchas canciones
- Usuarios - Historial Reproducción: 1 a N: un usuario puede tener muchos registros en su historial de canciones
- Canciones - Historial Reproducción: 1 a N: Cada registro en el Historial de Reproducción se refiere a una sola Canción

### Claves primarias de cada entidad:

- Usuario: ID\_usuario(PK)
- Playlist: ID\_playlist(PK)
- Canciones: ID\_canciones(PK)
- Historial de reproducciones: en este caso la combinación de ID canción((FK) y de ID Playlist (FK) actúan como una clave primaria compuesta





Consulta

Historial de Consultas

1

SELECT \* FROM public."Playlist"

2

ORDER BY "ID\_playlist" ASC

Data Output

Mensajes

Notificaciones

ID\_playlist

[PK] integer

Fecha\_Creacion

date

Nombre\_playlist

text

Consulta

Historial de Consultas

1

SELECT \* FROM public."Historial\_Reproducciones"

2

ORDER BY "ID\_Historial" ASC

Data Output

Mensajes

Notificaciones

ID\_Historial

[PK] integer

ID\_usuario

integer

ID\_cancion

integer

Fecha

date

Consulta

Historial de Consultas

1

SELECT \* FROM public."Usuario"

2

ORDER BY "ID\_usuario" ASC

Data Output

Mensajes

Notificaciones

ID\_usuario

[PK] integer

Nombre\_Usuario

text

Contraseña

character varying

Correo

character varying

Consulta

Historial de Consultas

1

SELECT \* FROM public."Cancion"

2

ORDER BY "ID\_cancion" ASC

Data Output

Mensajes

Notificaciones

ID\_cancion

[PK] integer

Genero

text

Artista

text

Fecha\_subida

date

## 6. Citas médicas

Entidades principales:

- Paciente
- Médico
- Cita

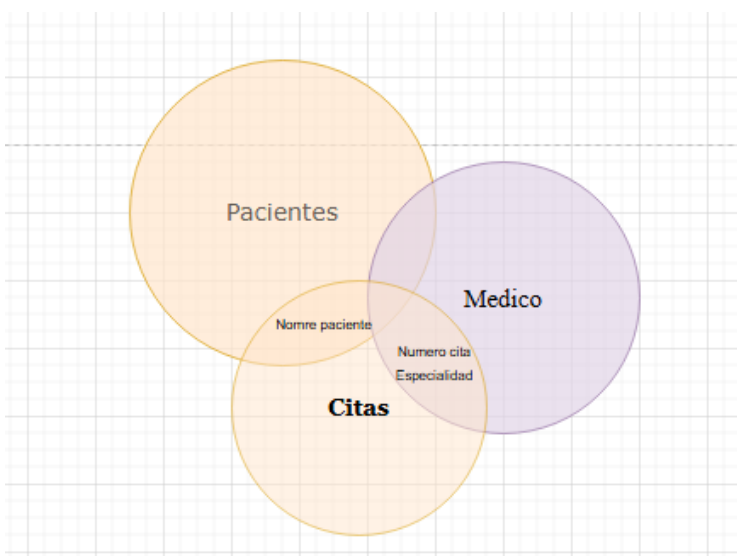
Atributos

1. Paciente: ID\_ paciente nombre completo del paciente, edad, número de seguro
2. Médico: ID\_médico nombre del médico, especialidad
3. Citas: ID\_citas, ID\_paciente(FK), ID\_medico(FK) número de cita nombre paciente nombre médico fecha

Relaciones

- Un paciente pueden tener una cita (1:1)
- Una cita puede tener un paciente(1:1)
- Muchos pacientes pueden tener muchos médicos(1:1)
- Muchos médicos pueden tener muchos pacientes (1:1)
- Cita relaciona médico con paciente

### Diagrama de Venn



ConsultaHistorial de Consultas

1

2

SELECT \* FROM public."Paciente"

ORDER BY "ID\_paciente" ASC

Data OutputMensajesNotificaciones

Consulta

Historial de Consultas

1

▼

SELECT \* FROM public.medico

2


ORDER BY "ID\_medico" ASC

Data Output


Mensajes

Notificaciones


≡+





▼




▼











ID\_medico

[PK] integer




Nombre

text



especialidad

text



Consulta

Historial de Consultas

1

SELECT \* FROM public."Citas"

2

ORDER BY "ID\_citas" ASC

Data Output

Mensajes

Notificaciones

≡+

▼

▼

ID\_citas

ID\_paciente

ID\_medico

Nombre\_paciente

fecha

nombre\_medico

[PK] integer

integer

integer

text

date

text

## 7. Biblioteca Virtual

### 1. Entidades Principales:

- Usuarios.
- Libros.
- Préstamos.
- Administración.

### 2. Atributos Clave del Sistema

- Usuario: ID\_Usuario (PK), Nombre, Tipo\_Usuario (Estándar / Administrador).
- Libros: ID\_Libro (PK), Título, Autor, Estado (Disponible / Prestado).
- Préstamos: ID\_Préstamo (PK), ID\_Usuario (FK), ID\_Libro (FK), Fecha\_Préstamo, Fecha\_Devolución\_Esperada.
- Administración: ID\_Usuario (FK) (*Debe ser administrador*), ID\_Libro (FK), Acción (Agregar, Eliminar, Editar), Fecha\_Acción.

### 3. Relaciones entre entidades

- Un Usuario puede realizar muchos Préstamos.
- Un Préstamo pertenece a un solo Usuario.
- El préstamo está asociado a un solo libro.

- Un libro puede tener muchos préstamos a lo largo del tiempo.
- Solo los Usuarios Administradores pueden administrar libros.
- Un administrador puede gestionar muchos libros.

#### 4. Claves primarias para cada entidad

- Usuarios: ID\_Usuario.
- Libros: ID\_Libro.
- Préstamos (Entidad débil o relación): ID\_Préstamo.
- Administración (Relación): ID\_Usuario, ID\_Libro, Fecha\_Acción.

#### 5. Diagrama de Venn y Diagrama Modelo E-R

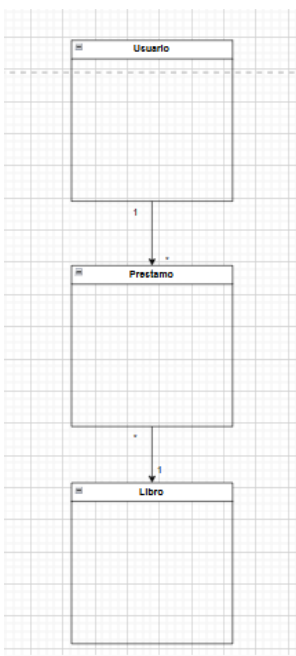
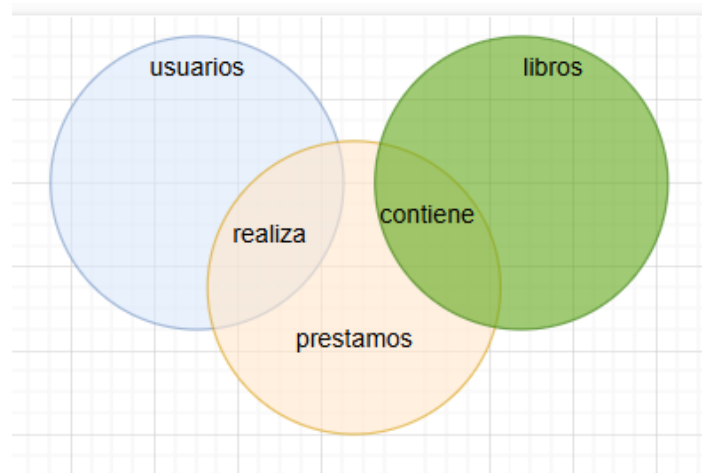
```
-- Crear tabla Usuarios
CREATE TABLE Usuarios (
  ID_Usuario SERIAL PRIMARY KEY,
  Nombre VARCHAR(100) NOT NULL,
  Tipo_Usuario VARCHAR(20) CHECK (Tipo_Usuario IN ('Estandar', 'Administrador'))
);

-- Crear tabla Libros
CREATE TABLE Libros (
  ID_Libro SERIAL PRIMARY KEY,
  Titulo VARCHAR(255) NOT NULL,
  Autor VARCHAR(255) NOT NULL,
  Estado VARCHAR(20) CHECK (Estado IN ('Disponible', 'Prestado')) DEFAULT 'Disponible'
);
```

```
-- Crear tabla Préstamos
CREATE TABLE Prestamos (
  ID_Prestamo SERIAL PRIMARY KEY,
  ID_Usuario INT NOT NULL,
  ID_Libro INT NOT NULL,
  Fecha_Prestamo DATE DEFAULT CURRENT_DATE,
  Fecha_Devolucion_Esperada DATE NOT NULL,
  FOREIGN KEY (ID_Usuario) REFERENCES Usuarios(ID_Usuario),
  FOREIGN KEY (ID_Libro) REFERENCES Libros(ID_Libro)
);

-- Crear tabla Administración (solo administradores pueden administrar libros)
CREATE TABLE Administracion (
  ID_Usuario INT NOT NULL,
  ID_Libro INT NOT NULL,
  Accion VARCHAR(50) CHECK (Accion IN ('Agregar', 'Eliminar', 'Editar')),
  Accion TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (ID_Usuario, ID_Libro, Fecha_Accion),
  FOREIGN KEY (ID_Usuario) REFERENCES Usuarios(ID_Usuario),
  FOREIGN KEY (ID_Libro) REFERENCES Libros(ID_Libro),
  CHECK (
    (SELECT Tipo_Usuario FROM Usuarios WHERE ID_Usuario = Administracion.ID_Usuario)
  )
);
```

```
-- Crear tabla Administración (solo administradores pueden administrar libros)
CREATE TABLE Administracion (
  ID_Usuario INT NOT NULL,
  ID_Libro INT NOT NULL,
  Accion VARCHAR(50) CHECK (Accion IN ('Agregar', 'Eliminar', 'Editar')),
  Accion TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (ID_Usuario, ID_Libro, Fecha_Accion),
  FOREIGN KEY (ID_Usuario) REFERENCES Usuarios(ID_Usuario),
  FOREIGN KEY (ID_Libro) REFERENCES Libros(ID_Libro),
  CHECK (
    (SELECT Tipo_Usuario FROM Usuarios WHERE ID_Usuario = Administracion.ID_Usuario)
  )
);
```



## 8. Sistema de Gestión de Proyectos

### 1. Entidades Principales:

- Empresa
- Proyecto
- Tarea
- Usuarios

### 2. Atributos Clave del Sistema

- Empresa: id\_empresa, nombre, ubicación.
- Proyecto: id\_proyecto, nombre, fecha\_inicio, fecha\_fin, id\_empresa.
- Tarea: id\_tarea, descripcion, estado, fecha\_inicio, fecha\_fin, id\_proyecto.
- usuario: id\_usuario, nombre, cargo, email.

### 3. Relaciones entre entidades

- Una empresa puede registrar múltiples proyectos. (1:N)
- Un proyecto tiene múltiples tareas. (1:N)
- Una tarea puede tener múltiples responsables, y un responsable puede encargarse de múltiples tareas. (N:M)
- Cada tarea tiene múltiples avances registrados. (1:N)

### 4. Claves primarias para cada entidad

- Empresa: id\_empresa (PK)
- Proyecto: id\_proyecto (PK), id\_empresa (FK)
- Tarea: id\_tarea (PK), id\_proyecto (FK)
- Responsable: id\_responsable (PK)
- Avance: id\_avance (PK), id\_tarea (FK)
- Tarea\_Responsable (Tabla intermedia para la relación N:M entre Tarea y Responsable): id\_tarea (FK), id\_responsable (FK), (PK compuesta)

## 5. Diagrama de Venn

```

16 CREATE TABLE Empresa (
17     id_empresa SERIAL PRIMARY KEY,
18     nombre VARCHAR(100) NOT NULL,
19     ubicacion VARCHAR(255)
20 );
21
22 CREATE TABLE Proyecto (
23     id_proyecto SERIAL PRIMARY KEY,
24     nombre VARCHAR(100) NOT NULL,
25     fecha_inicio DATE,
26     fecha_fin DATE,
27     id_empresa INT NOT NULL,
28     FOREIGN KEY (id_empresa) REFERENCES Empresa(id_empresa) ON DELETE CASCADE
29 );
30

```

```

CREATE TABLE Tarea (
    id_tarea SERIAL PRIMARY KEY,
    descripcion TEXT NOT NULL,
    estado VARCHAR(50) CHECK (estado IN ('Pendiente', 'En Progreso', 'Completada')),
    fecha_inicio DATE,
    fecha_fin DATE,
    id_proyecto INT NOT NULL,
    FOREIGN KEY (id_proyecto) REFERENCES Proyecto(id_proyecto) ON DELETE CASCADE
);

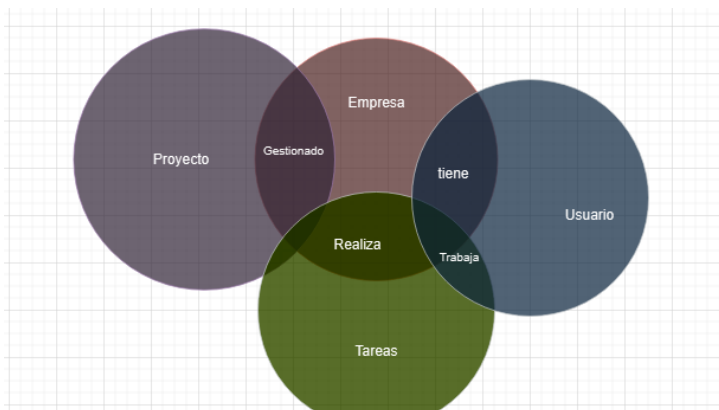
CREATE TABLE Responsable (
    id_responsable SERIAL PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    cargo VARCHAR(100),
    email VARCHAR(100) UNIQUE
);

```

```

48 CREATE TABLE Avance (
49     id_avance SERIAL PRIMARY KEY,
50     descripcion TEXT NOT NULL,
51     fecha TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
52     id_tarea INT NOT NULL,
53     FOREIGN KEY (id_tarea) REFERENCES Tarea(id_tarea) ON DELETE CASCADE
54 );
55
56 CREATE TABLE Tarea_Responsable (
57     id_tarea INT NOT NULL,
58     id_responsable INT NOT NULL,
59     PRIMARY KEY (id_tarea, id_responsable),
60     FOREIGN KEY (id_tarea) REFERENCES Tarea(id_tarea) ON DELETE CASCADE,
61     FOREIGN KEY (id_responsable) REFERENCES Responsable(id_responsable) ON DELETE CASCADE
62 );
63
64

```



## 9. Red social

Entidades:

usuarios

reacciones

publicaciones

comentarios

Atributos clave:

Usuario ID\_usuario(PK), nombre, apellido, Número, amigos

publicaciones: ID\_publicaciones(PK), Id\_comentarios(fk) id\_reacciones(fk)

Autor, número de reacciones números comentarios, texto comentario

Reacciones: ID\_reaccion, número de reacciones, tipo de reacción

Comentarios: ID\_ Comentarios, número de comentarios ,ID\_usuario(FK)

Relaciones

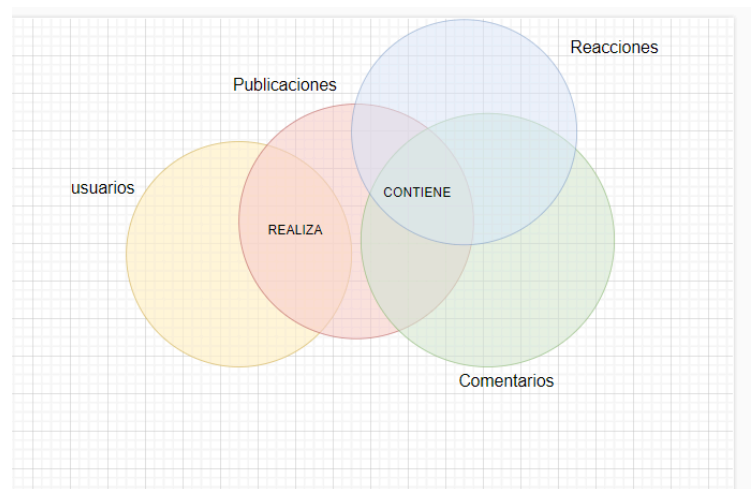
Los usuarios se relacionan con las publicaciones (M:M)

Las publicaciones pueden tener muchos usuarios (M:M)

Y las reacciones y comentarios se relacionan con Publicaciones

Consulta	Historial de Consultas
1	SELECT * FROM public."Comentarios"
2	ORDER BY "ID_comentario" ASC

Data Output	Mensajes	Notificaciones
ID_comentario [PK] integer	#_comentarios integer	ID_usuario integer



Consulta	Historial de Consultas
1	SELECT * FROM public."Publicaciones"
2	ORDER BY "ID_publicacion" ASC

Data Output	Mensajes	Notificaciones
ID_publicacion [PK] integer	ID_comentarios integer	ID_Reacciones integer
Autor text	#_comentarios integer	#_reacciones integer

Consulta	Historial de Consultas
1	SELECT * FROM public."Usuario"
2	ORDER BY "ID_usuario" ASC

Data Output	Mensajes	Notificaciones
ID_usuario [PK] integer	Nombre text	Apellido text
Número_telefono integer	#_Amigos integer	

10.

Consulta	Historial de Consultas
1	SELECT * FROM public.reacciones
2	ORDER BY "#_reacciones" ASC

Data Output	Mensajes	Notificaciones
#_reacciones [PK] integer	Tipo_reaccion text	

## Sistema de Facturación

1. Entidades Principales:

- Cliente
- Factura
- Producto
- Detalle\_Factura
- Impuesto
- Descuento
- Factura\_Impuestos
- Factura\_Descuentos

## 2. Atributos Clave del Sistema

- Clientes: ID\_Cliente (PK), Nombre, Dirección, Teléfono, Correo.
- Factura: ID\_Factura (PK), Fecha, Total, ID\_Cliente (FK).
- Producto: ID\_Producto (PK), Nombre, Precio, Stock.
- Detalle\_Factura: ID\_Factura (FK), ID\_Producto (FK), Cantidad, Subtotal
- Impuestos: ID\_Impuesto (PK), Tipo (IVA, otros), Tasa.
- Descuento: ID\_Descuento (PK), Tipo (porcentaje, fijo), Monto.
- Factura\_Impuestos: ID\_Factura (FK), ID\_Impuesto (FK), Monto\_Impuesto.
- Factura\_Descuentos: ID\_Factura (FK), ID\_Descuento (FK), Monto\_Descuento.

## 3. Relaciones entre entidades

- Cliente - Factura: Un cliente puede tener muchas facturas (1:N).
- Factura - Detalle\_Factura: Una factura tiene muchos productos a través de los detalles (1:N).
- Producto - Detalle\_Factura: Un producto puede estar en muchas facturas (N:M).
- Factura - Impuesto: Una factura puede tener uno o más impuestos aplicados (N:M).
- Factura - Descuento: Una factura puede tener uno o más descuentos aplicados (N:M).

## 4. Claves primarias para cada entidad

- Clientes: ID\_Cliente.
- Factura: ID\_Factura.
- Producto: ID\_Producto.
- Detalle\_Factura: Combinación de (ID\_Factura, ID\_Producto).
- Impuestos: ID\_Impuesto.
- Descuento: ID\_Descuento.
- Factura\_Impuestos: Combinación de (ID\_Factura, ID\_Impuesto).
- Factura\_Descuentos: Combinación de (ID\_Factura, ID\_Descuento).



## 5. Diagrama de Venn y Diagrama Modelo E-R

```
10
17 -- Crear la tabla Cliente
18 CREATE TABLE Cliente (
19     ID_Cliente SERIAL PRIMARY KEY,
20     Nombre VARCHAR(100),
21     Direccion VARCHAR(255),
22     Telefono VARCHAR(20),
23     Correo VARCHAR(100)
24 );
25
26 -- Crear la tabla Producto
27 CREATE TABLE Producto (
28     ID_Producto SERIAL PRIMARY KEY,
29     Nombre VARCHAR(100),
30     Precio DECIMAL(10, 2),
31     Stock INT
32 );
33
34 -- Crear la tabla Factura
35 CREATE TABLE Factura (
36     ID_Factura SERIAL PRIMARY KEY,
37     Fecha DATE,
38     Total DECIMAL(10, 2),
39     ID_Cliente INT,
40     FOREIGN KEY (ID_Cliente) REFERENCES Cliente(ID_Cliente)
41 );
42
43 -- Crear la tabla Detalle_Factura
44 CREATE TABLE Detalle_Factura (
45     ID_Factura INT,
46     ID_Producto INT,
47     Cantidad INT,
48     Subtotal DECIMAL(10, 2),
49     PRIMARY KEY (ID_Factura, ID_Producto),
50     FOREIGN KEY (ID_Factura) REFERENCES Factura(ID_Factura),
51     FOREIGN KEY (ID_Producto) REFERENCES Producto(ID_Producto)
52 );
53
54
55
56 );
57
58 -- Crear la tabla Factura_Impuesto (relación Factura - Impuesto)
59 CREATE TABLE Factura_Impuesto (
60     ID_Factura INT,
61     ID_Impuesto INT,
62     Monto_Impuesto DECIMAL(10, 2),
63     PRIMARY KEY (ID_Factura, ID_Impuesto),
64     FOREIGN KEY (ID_Factura) REFERENCES Factura(ID_Factura),
65     FOREIGN KEY (ID_Impuesto) REFERENCES Impuesto(ID_Impuesto)
66 );
67
68 -- Crear la tabla Factura_Descuento (relación Factura - Descuento)
69 CREATE TABLE Factura_Descuento (
70     ID_Factura INT,
71     ID_Descuento INT,
72     Monto_Descuento DECIMAL(10, 2),
73     PRIMARY KEY (ID_Factura, ID_Descuento),
74     FOREIGN KEY (ID_Factura) REFERENCES Factura(ID_Factura),
75     FOREIGN KEY (ID_Descuento) REFERENCES Descuento(ID_Descuento)
76 );
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```