

## Diagrama entidad relación

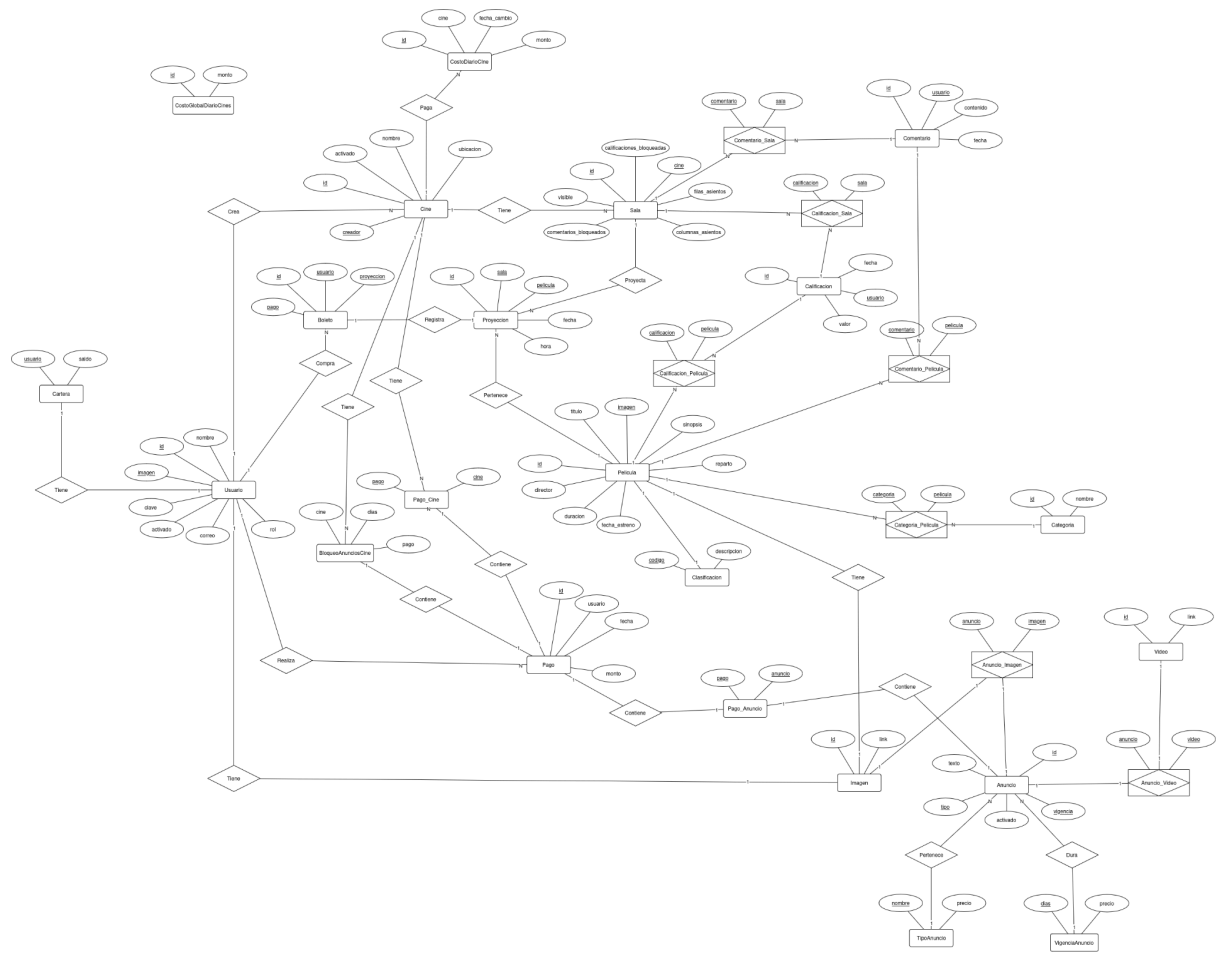
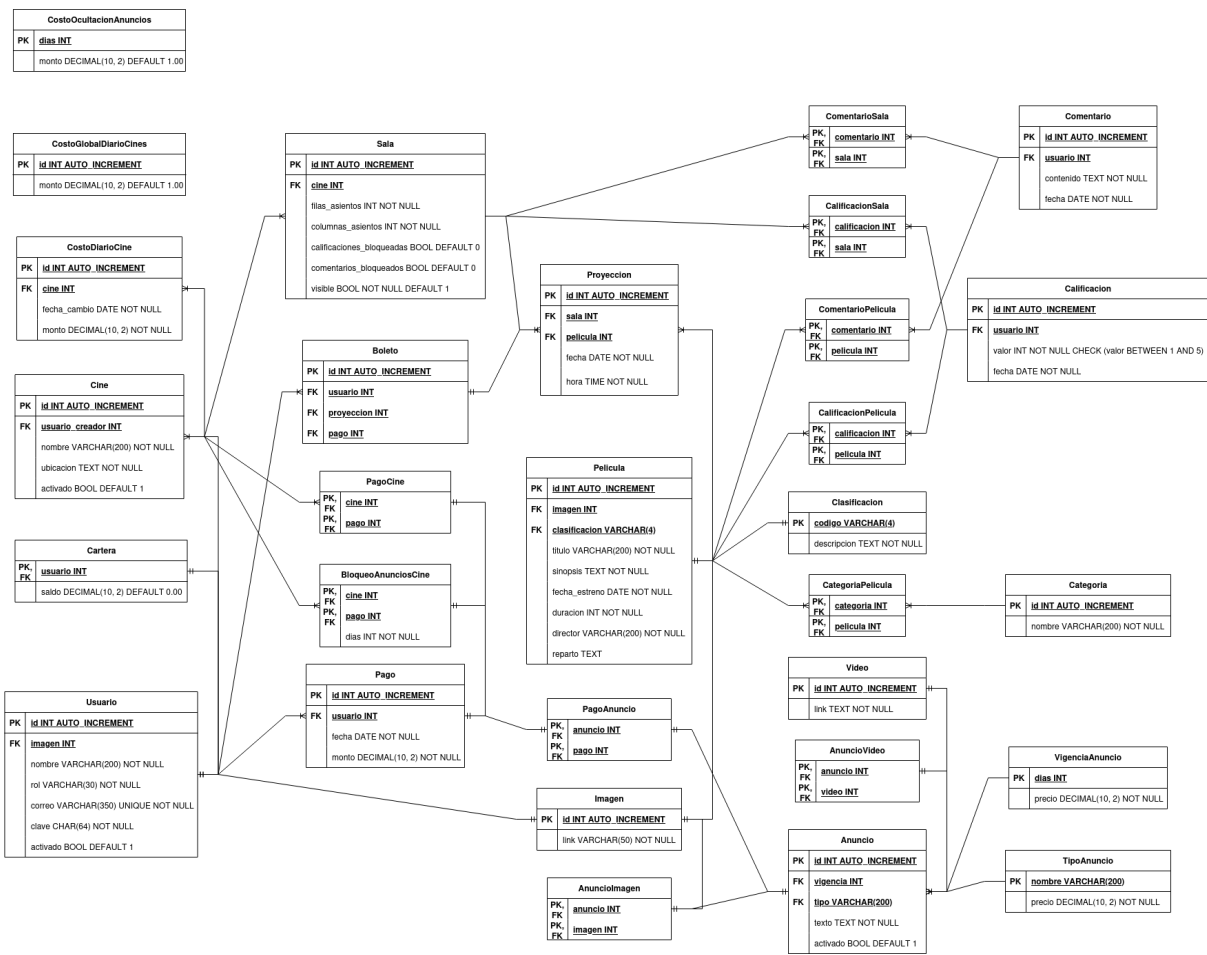


Diagrama de tablas



## Mapeo físico de la base de datos

-- Esquema

```
CREATE DATABASE ManejadorCines;
```

```
USE ManejadorCines;
```

```
CREATE TABLE Imagen (  
    id INT AUTO_INCREMENT,  
    link VARCHAR(50) NOT NULL,  
    PRIMARY KEY (id)  
);
```

```
CREATE TABLE Usuario (  
    id INT AUTO_INCREMENT,  
    imagen INT DEFAULT 1,  
    nombre VARCHAR(200) NOT NULL,  
    rol VARCHAR(30) NOT NULL,  
    correo VARCHAR(350) UNIQUE NOT NULL,  
    clave CHAR(64) NOT NULL,  
    activado BOOL DEFAULT 1,  
    PRIMARY KEY (id),  
    CONSTRAINT fk_imagen_usuario FOREIGN KEY (imagen) REFERENCES Imagen (id)  
);
```

```
CREATE TABLE Cartera (  
    usuario INT,  
    saldo DECIMAL(10, 2) DEFAULT 0.00,  
    PRIMARY KEY (usuario),  
    CONSTRAINT fk_usuario_cartera FOREIGN KEY (usuario) REFERENCES Usuario (id)  
);
```

```
CREATE TABLE Cine (  
    id INT AUTO_INCREMENT,  
    usuario_creador INT,  
    nombre VARCHAR(200) NOT NULL,  
    ubicacion TEXT NOT NULL,  
    activado BOOL DEFAULT 1,  
    PRIMARY KEY (id),  
    CONSTRAINT fk_usuario_cine FOREIGN KEY (usuario_creador) REFERENCES Usuario  
(id)  
);
```

```
CREATE TABLE CostoDiarioCine (  
    id INT AUTO_INCREMENT,  
    cine INT,  
    fecha_cambio DATE NOT NULL,  
    monto DECIMAL(10, 2) NOT NULL,  
    PRIMARY KEY (id),  
    CONSTRAINT fk_cine_costodiariocine FOREIGN KEY (cine) REFERENCES Cine (id) ON  
DELETE CASCADE
```

);

```
CREATE TABLE CostoGlobalDiarioCines (  
    id INT AUTO_INCREMENT,  
    monto DECIMAL(10, 2) DEFAULT 1.00,  
    PRIMARY KEY (id)
```

);

```
CREATE TABLE CostoOcultacionAnuncios (  
    dias INT,  
    monto DECIMAL(10, 2) DEFAULT 1.00,  
    PRIMARY KEY (dias)
```

);

```
CREATE TABLE Sala (  
    id INT AUTO_INCREMENT,  
    cine INT,  
    filas_asientos INT NOT NULL,  
    columnas_asientos INT NOT NULL,  
    calificaciones_bloqueadas BOOL DEFAULT 0,  
    comentarios_bloqueados BOOL DEFAULT 0,  
    visible BOOL NOT NULL DEFAULT 1,  
    PRIMARY KEY (id),  
    CONSTRAINT fk_cine_sala FOREIGN KEY (cine) REFERENCES Cine (id) ON DELETE  
CASCADE
```

);

```
CREATE TABLE Pago (  
    id INT AUTO_INCREMENT,  
    usuario INT,  
    fecha DATE NOT NULL,  
    monto DECIMAL(10, 2) NOT NULL,  
    PRIMARY KEY (id),  
    CONSTRAINT fk_usuario_pago FOREIGN KEY (usuario) REFERENCES Usuario (id)
```

);

```
CREATE TABLE Clasificacion (  
    codigo VARCHAR(4),  
    descripcion TEXT NOT NULL,  
    PRIMARY KEY (codigo)
```

);

```
CREATE TABLE Pelicula (  
    id INT AUTO_INCREMENT,  
    imagen INT,  
    clasificacion VARCHAR(4),  
    titulo VARCHAR(200) NOT NULL,  
    sinopsis TEXT NOT NULL,  
    fecha_estreno DATE NOT NULL,  
    duracion INT NOT NULL,  
    director VARCHAR(200) NOT NULL,  
    reparto TEXT,
```

```

PRIMARY KEY (id),
CONSTRAINT fk_imagen_pelicula FOREIGN KEY (imagen) REFERENCES Imagen (id),
CONSTRAINT fk_clasificacion_pelicula FOREIGN KEY (clasificacion) REFERENCES
Clasificacion (codigo)
);

```

```

CREATE TABLE Proyeccion (
    id INT AUTO_INCREMENT,
    sala INT,
    pelicula INT,
    fecha DATE NOT NULL,
    hora TIME NOT NULL,
    precio DECIMAL(10, 2) DEFAULT(30.00),
    PRIMARY KEY (id),
    CONSTRAINT fk_sala_proyeccion FOREIGN KEY (sala) REFERENCES Sala (id),
    CONSTRAINT fk_pelicula_proyeccion FOREIGN KEY (pelicula) REFERENCES Pelicula (id)
);

```

```

CREATE TABLE Boleto (
    id INT AUTO_INCREMENT,
    usuario INT,
    proyeccion INT,
    pago INT,
    PRIMARY KEY (id),
    CONSTRAINT fk_usuario_boleto FOREIGN KEY (usuario) REFERENCES Usuario (id),
    CONSTRAINT fk_proyeccion_boleto FOREIGN KEY (proyeccion) REFERENCES Proyeccion
(id),
    CONSTRAINT fk_pago_boleto FOREIGN KEY (pago) REFERENCES Pago (id)
);

```

```

CREATE TABLE PagoCine (
    cine INT,
    pago INT,
    PRIMARY KEY (cine, pago),
    CONSTRAINT fk_cine_pagocine FOREIGN KEY (cine) REFERENCES Cine (id),
    CONSTRAINT fk_pago_pagocine FOREIGN KEY (pago) REFERENCES Pago (id)
);

```

```

CREATE TABLE BloqueoAnunciosCine (
    cine INT,
    pago INT,
    dias INT NOT NULL,
    PRIMARY KEY (cine, pago),
    CONSTRAINT fk_cine_bloqueoanuncioscine FOREIGN KEY (cine) REFERENCES Cine (id),
    CONSTRAINT fk_pago_bloqueoanuncioscine FOREIGN KEY (pago) REFERENCES Pago
(id)
);

```

```

CREATE TABLE Comentario (
    id INT AUTO_INCREMENT,
    usuario INT,
    contenido TEXT NOT NULL,

```

```

        fecha DATE NOT NULL,
        PRIMARY KEY (id),
        CONSTRAINT fk_usuario_comentario FOREIGN KEY (usuario) REFERENCES Usuario (id)
    );

```

```

CREATE TABLE Calificacion (
    id INT AUTO_INCREMENT,
    usuario INT,
    valor INT NOT NULL CHECK (valor BETWEEN 1 AND 5),
    fecha DATE NOT NULL,
    PRIMARY KEY (id),
    CONSTRAINT fk_usuario_calificacion FOREIGN KEY (usuario) REFERENCES Usuario (id)
);

```

```

CREATE TABLE ComentarioSala (
    comentario INT,
    sala INT,
    PRIMARY KEY (comentario, sala),
    CONSTRAINT fk_comentario_comentariosala FOREIGN KEY (comentario) REFERENCES
Comentario (id) ON DELETE CASCADE,
    CONSTRAINT fk_sala_comentariosala FOREIGN KEY (sala) REFERENCES Sala (id)
);

```

```

CREATE TABLE CalificacionSala (
    calificacion INT,
    sala INT,
    PRIMARY KEY (calificacion, sala),
    CONSTRAINT fk_calificacion_calificacionsala FOREIGN KEY (calificacion) REFERENCES
Calificacion (id),
    CONSTRAINT fk_sala_calificacionsala FOREIGN KEY (sala) REFERENCES Sala (id)
);

```

```

CREATE TABLE ComentarioPelicula (
    comentario INT,
    pelicula INT,
    PRIMARY KEY (comentario, pelicula),
    CONSTRAINT fk_comentario_comentariopelicula FOREIGN KEY (comentario)
REFERENCES Comentario (id) ON DELETE CASCADE,
    CONSTRAINT fk_pelicula_comentariopelicula FOREIGN KEY (pelicula) REFERENCES
Pelicula (id)
);

```

```

CREATE TABLE CalificacionPelicula (
    calificacion INT,
    pelicula INT,
    PRIMARY KEY (calificacion, pelicula),
    CONSTRAINT fk_calificacion_calificacionpelicula FOREIGN KEY (calificacion)
REFERENCES Calificacion (id),
    CONSTRAINT fk_pelicula_calificacionpelicula FOREIGN KEY (pelicula) REFERENCES
Pelicula (id)
);

```

```
CREATE TABLE Categoria (
    id INT AUTO_INCREMENT,
    nombre VARCHAR(200) NOT NULL,
    PRIMARY KEY (id)
);
```

```
CREATE TABLE CategoriaPelicula (
    categoria INT,
    pelicula INT,
    PRIMARY KEY (categoria, pelicula),
    CONSTRAINT fk_categoria_categoriapelicula FOREIGN KEY (categoria) REFERENCES
Categoria (id),
    CONSTRAINT fk_pelicula_categoriapelicula FOREIGN KEY (pelicula) REFERENCES
Pelicula (id)
);
```

```
CREATE TABLE Video (
    id INT AUTO_INCREMENT,
    link TEXT NOT NULL,
    PRIMARY KEY (id)
);
```

```
CREATE TABLE VigenciaAnuncio (
    dias INT,
    precio DECIMAL(10, 2) NOT NULL,
    PRIMARY KEY (dias)
);
```

```
CREATE TABLE TipoAnuncio (
    nombre VARCHAR(200),
    precio DECIMAL(10, 2) NOT NULL,
    PRIMARY KEY (nombre)
);
```

```
CREATE TABLE Anuncio (
    id INT AUTO_INCREMENT,
    vigencia INT,
    tipo VARCHAR(200),
    texto TEXT NOT NULL,
    activado BOOL DEFAULT 1,
    PRIMARY KEY (id),
    CONSTRAINT fk_vigencia_anuncio FOREIGN KEY (vigencia) REFERENCES
VigenciaAnuncio (dias),
    CONSTRAINT fk_tipo_anuncio FOREIGN KEY (tipo) REFERENCES TipoAnuncio (nombre)
);
```

```
CREATE TABLE PagoAnuncio (
    anuncio INT,
    pago INT,
    PRIMARY KEY (anuncio, pago),
    CONSTRAINT fk_anuncio_pagoanuncio FOREIGN KEY (anuncio) REFERENCES Anuncio
(id),
```



```

        CONSTRAINT fk_pago_pagoanuncio FOREIGN KEY (pago) REFERENCES Pago (id)
    );

    CREATE TABLE AnuncioImagen (
        anuncio INT,
        imagen INT,
        PRIMARY KEY (anuncio, imagen),
        CONSTRAINT fk_anuncio_anuncioimagen FOREIGN KEY (anuncio) REFERENCES Anuncio
    (id),
        CONSTRAINT fk_imagen_anuncioimagen FOREIGN KEY (imagen) REFERENCES Imagen
    (id)
    );

```

```

    CREATE TABLE AnuncioVideo (
        anuncio INT,
        video INT,
        PRIMARY KEY (anuncio, video),
        CONSTRAINT fk_anuncio_anunciovideo FOREIGN KEY (anuncio) REFERENCES Anuncio
    (id),
        CONSTRAINT fk_video_anunciovideo FOREIGN KEY (video) REFERENCES Video (id)
    );

```

-- Datos iniciales

```
INSERT INTO Imagen (link) VALUES ("1.png");
```

```

INSERT INTO Usuario
    (imagen, nombre, rol, correo, clave)
VALUES
    (1, 'adminsistema1', 'ADMINISTRADOR_SISTEMA', 'admin@mail.com',
    '5feceb66ffc86f38d952786c6d696c79c2dbc239dd4e91b46729d73a27fb57e9'),
    (1, 'admindines1', 'ADMINISTRADOR_CINES', 'admindine@mail.com',
    '5feceb66ffc86f38d952786c6d696c79c2dbc239dd4e91b46729d73a27fb57e9'),
    (1, 'cliente1', 'CLIENTE', 'cliente@mail.com',
    '5feceb66ffc86f38d952786c6d696c79c2dbc239dd4e91b46729d73a27fb57e9'),
    (1, 'anunciante1', 'ANUNCIANTE', 'anunciante@mail.com',
    '5feceb66ffc86f38d952786c6d696c79c2dbc239dd4e91b46729d73a27fb57e9');

```

```

INSERT INTO Cartera
    (usuario)

```

```

VALUES
    (1),
    (2),
    (3),
    (4);

```

```

INSERT INTO Clasificacion
    (codigo, descripcion)
VALUES
    ('A', 'Apto para todo público'),
    ('B', 'Apto para todo público, especialmente para mayores de 6 años'),
    ('B-12', 'Apto para mayores de 12 años'),

```

('B-15', 'Apto para mayores de 15 años'),  
('C', 'Mayores de 18 años');

INSERT INTO Categoria  
(nombre)

VALUES

('Acción'),  
('Aventura'),  
('Comedia'),  
('Drama'),  
('Terror'),  
('Ciencia Ficción'),  
('Fantasía'),  
('Romance'),  
('Misterio'),  
('Suspenso'),  
('Animación'),  
('Documental'),  
('Musical'),  
('Crimen'),  
('Histórica'),  
('Bélica'),  
('Deportes'),  
('Superhéroes'),  
('Thriller Psicológico');

INSERT INTO VigenciaAnuncio  
(días, precio)

VALUES

(1, 1.00),  
(3, 2.00),  
(7, 3.00),  
(14, 4.00);

INSERT INTO TipoAnuncio  
(nombre, precio)

VALUES

('TEXTO', 2.00),  
('TEXTO\_IMAGEN', 4.00),  
('TEXTO\_VIDEO', 8.00);

INSERT INTO CostoGlobalDiarioCines  
(monto)

VALUES

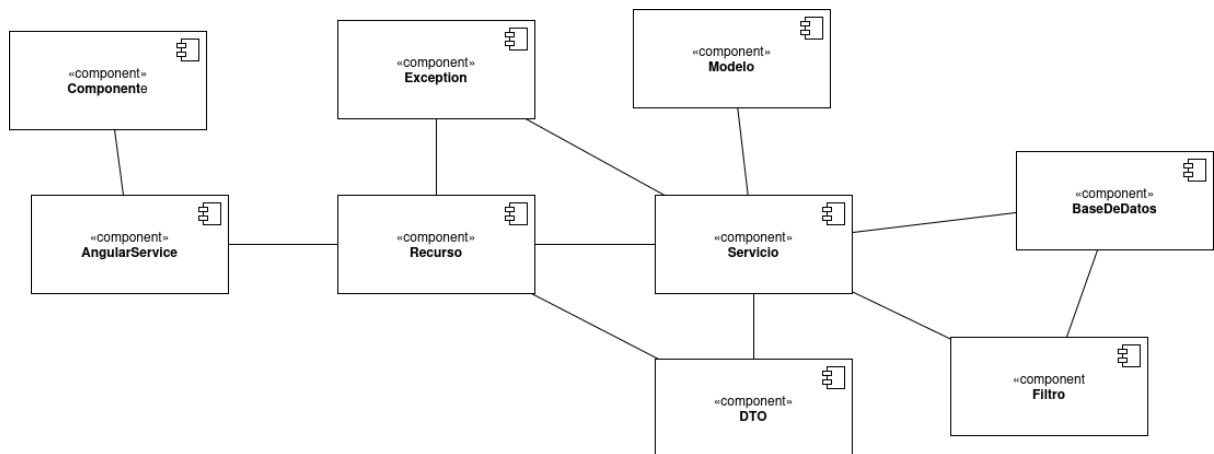
(1);

INSERT INTO CostoOcultacionAnuncios  
(días, monto)

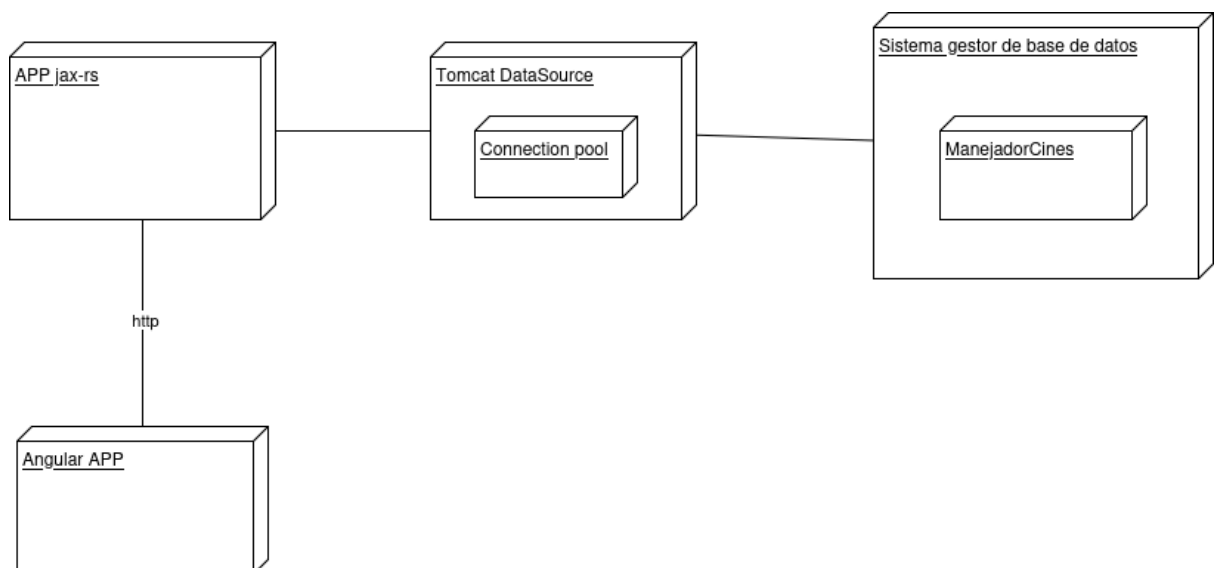
VALUES

(7, 10),  
(30, 35),;

## Diagrama de componentes



## Diagrama de despliegue



## Instrucciones de instalación

Instalar estos programas:

1. Java Development Kit (JDK) versión 21
2. Maven versión 3.8 o superior
3. Node.js y npm
4. Angular CLI
5. MySQL Server versión 8.0 o superior
6. Apache Tomcat versión 11
7. Git

Descargar el proyecto desde el repositorio usando git.

Ejecutar el archivo mapeo\_fisico.sql en la base de datos MySQL.

Ingresar credenciales en el archivo config.properties.

Compilar el archivo con tomcat, creando el archivo .war.

Al iniciar tomcat, la aplicación en local estará corriendo en el puerto 8080.

Para iniciar el frontend, ejecutar ng serve en la raíz del proyecto angular.

La aplicación estará corriendo en el puerto local 4200.