

GESTION ESCOLAR

```
CREATE DATABASE IF NOT EXISTS GestionEscolar;  
USE GestionEscolar;
```

```
CREATE TABLE Alumnos (  
    id_alumno INT PRIMARY KEY AUTO_INCREMENT,  
    nombre VARCHAR(100),  
    correo VARCHAR(100) UNIQUE  
);
```

```
CREATE TABLE Profesores (  
    id_profesor INT PRIMARY KEY AUTO_INCREMENT,  
    nombre VARCHAR(100),  
    correo VARCHAR(100) UNIQUE  
);
```

```
CREATE TABLE Cursos (  
    id_curso INT PRIMARY KEY AUTO_INCREMENT,  
    nombre VARCHAR(100),  
    id_profesor INT,  
    FOREIGN KEY (id_profesor) REFERENCES Profesores(id_profesor)  
);
```

```
CREATE TABLE Inscripciones (  
    id_inscripcion INT PRIMARY KEY AUTO_INCREMENT,  
    id_alumno INT,  
    id_curso INT,  
    fecha_inscripcion DATE,  
    FOREIGN KEY (id_alumno) REFERENCES Alumnos(id_alumno),  
    FOREIGN KEY (id_curso) REFERENCES Cursos(id_curso)  
);
```

```
CREATE TABLE Calificaciones (  
    id_calificacion INT PRIMARY KEY AUTO_INCREMENT,  
    id_inscripcion INT,  
    nota DECIMAL(4,2),
```

```
FOREIGN KEY (id_inscripcion) REFERENCES Inscripciones(id_inscripcion)
);
```

APP DE MENSAJES

```
CREATE DATABASE IF NOT EXISTS Mensajeria;
USE Mensajeria;
```

```
CREATE TABLE Usuarios (
    id_usuario INT PRIMARY KEY AUTO_INCREMENT,
    nombre VARCHAR(100),
    correo VARCHAR(100) UNIQUE
);
```

```
CREATE TABLE Mensajes (
    id_mensaje INT PRIMARY KEY AUTO_INCREMENT,
    id_emisor INT,
    id_receptor INT,
    contenido TEXT,
    fecha_envio DATETIME,
    FOREIGN KEY (id_emisor) REFERENCES Usuarios(id_usuario),
    FOREIGN KEY (id_receptor) REFERENCES Usuarios(id_usuario)
);
```

```
CREATE TABLE Adjuntos (
    id_adjunto INT PRIMARY KEY AUTO_INCREMENT,
    id_mensaje INT,
    archivo VARCHAR(255),
    FOREIGN KEY (id_mensaje) REFERENCES Mensajes(id_mensaje)
);
```

STREAMING DE MUSICA

```
CREATE DATABASE IF NOT EXISTS StreamingMusica;
USE StreamingMusica;
```

```
CREATE TABLE Usuarios (
    id_usuario INT PRIMARY KEY AUTO_INCREMENT,
    nombre VARCHAR(100),
    correo VARCHAR(100) UNIQUE
```

);

```
CREATE TABLE Artistas (  
    id_artista INT PRIMARY KEY AUTO_INCREMENT,  
    nombre VARCHAR(100)  
);
```

```
CREATE TABLE Canciones (  
    id_cancion INT PRIMARY KEY AUTO_INCREMENT,  
    titulo VARCHAR(100),  
    id_artista INT,  
    FOREIGN KEY (id_artista) REFERENCES Artistas(id_artista)  
);
```

```
CREATE TABLE Playlists (  
    id_playlist INT PRIMARY KEY AUTO_INCREMENT,  
    id_usuario INT,  
    nombre VARCHAR(100),  
    FOREIGN KEY (id_usuario) REFERENCES Usuarios(id_usuario)  
);
```

```
CREATE TABLE PlaylistCanciones (  
    id_playlist INT,  
    id_cancion INT,  
    PRIMARY KEY (id_playlist, id_cancion),  
    FOREIGN KEY (id_playlist) REFERENCES Playlists(id_playlist),  
    FOREIGN KEY (id_cancion) REFERENCES Canciones(id_cancion)  
);
```

```
CREATE TABLE HistorialReproduccion (  
    id_historial INT PRIMARY KEY AUTO_INCREMENT,  
    id_usuario INT,  
    id_cancion INT,  
    fecha_reproduccion DATETIME,  
    FOREIGN KEY (id_usuario) REFERENCES Usuarios(id_usuario),  
    FOREIGN KEY (id_cancion) REFERENCES Canciones(id_cancion)  
);
```

GESTION DE CITAS MEDICAS

```
CREATE DATABASE IF NOT EXISTS CitasMedicas;  
USE CitasMedicas;
```

```
CREATE TABLE Pacientes (  
    id_paciente INT PRIMARY KEY AUTO_INCREMENT,  
    nombre VARCHAR(100),  
    correo VARCHAR(100) UNIQUE  
);
```

```
CREATE TABLE Medicos (  
    id_medico INT PRIMARY KEY AUTO_INCREMENT,  
    nombre VARCHAR(100),  
    especialidad VARCHAR(100)  
);
```

```
CREATE TABLE Citas (  
    id_cita INT PRIMARY KEY AUTO_INCREMENT,  
    id_paciente INT,  
    id_medico INT,  
    fecha DATETIME,  
    estado ENUM('pendiente', 'atendida', 'cancelada'),  
    FOREIGN KEY (id_paciente) REFERENCES Pacientes(id_paciente),  
    FOREIGN KEY (id_medico) REFERENCES Medicos(id_medico)  
);
```

SISTEMA DE FACTURACION

```
CREATE DATABASE IF NOT EXISTS Facturacion;  
USE Facturacion;
```

```
CREATE TABLE Clientes (  
    id_cliente INT PRIMARY KEY AUTO_INCREMENT,  
    nombre VARCHAR(100),  
    correo VARCHAR(100) UNIQUE  
);
```

```
CREATE TABLE Productos (  
    id_producto INT PRIMARY KEY AUTO_INCREMENT,
```

```
    nombre VARCHAR(100),  
    precio DECIMAL(10,2)  
);
```

```
CREATE TABLE Facturas (  
    id_factura INT PRIMARY KEY AUTO_INCREMENT,  
    id_cliente INT,  
    fecha DATE,  
    FOREIGN KEY (id_cliente) REFERENCES Clientes(id_cliente)  
);
```

```
CREATE TABLE DetalleFactura (  
    id_detalle INT PRIMARY KEY AUTO_INCREMENT,  
    id_factura INT,  
    id_producto INT,  
    cantidad INT,  
    precio_unitario DECIMAL(10,2),  
    FOREIGN KEY (id_factura) REFERENCES Facturas(id_factura),  
    FOREIGN KEY (id_producto) REFERENCES Productos(id_producto)  
);
```

BIBLIOTECA DIGITAL

```
CREATE DATABASE IF NOT EXISTS BibliotecaDigital;  
USE BibliotecaDigital;
```

```
CREATE TABLE Usuarios (  
    id_usuario INT PRIMARY KEY AUTO_INCREMENT,  
    nombre VARCHAR(100),  
    correo VARCHAR(100) UNIQUE  
);
```

```
CREATE TABLE Libros (  
    id_libro INT PRIMARY KEY AUTO_INCREMENT,  
    titulo VARCHAR(100),  
    autor VARCHAR(100),  
    anio_publicacion INT  
);
```

```
CREATE TABLE Prestamos (  
    id_prestamo INT PRIMARY KEY AUTO_INCREMENT,  
    id_usuario INT,  
    id_libro INT,  
    fecha_prestamo DATE,  
    fecha_devolucion_esperada DATE,  
    FOREIGN KEY (id_usuario) REFERENCES Usuarios(id_usuario),  
    FOREIGN KEY (id_libro) REFERENCES Libros(id_libro)  
);
```

GESTION DE PROYECTOS

```
CREATE DATABASE IF NOT EXISTS GestionProyectos;  
USE GestionProyectos;
```

```
CREATE TABLE Empresas (  
    id_empresa INT PRIMARY KEY AUTO_INCREMENT,  
    nombre VARCHAR(100)  
);
```

```
CREATE TABLE Usuarios (  
    id_usuario INT PRIMARY KEY AUTO_INCREMENT,  
    nombre VARCHAR(100),  
    correo VARCHAR(100) UNIQUE  
);
```

```
CREATE TABLE Proyectos (  
    id_proyecto INT PRIMARY KEY AUTO_INCREMENT,  
    id_empresa INT,  
    nombre VARCHAR(100),  
    fecha_inicio DATE,  
    fecha_fin DATE,  
    FOREIGN KEY (id_empresa) REFERENCES Empresas(id_empresa)  
);
```

```
CREATE TABLE Tareas (  
    id_tarea INT PRIMARY KEY AUTO_INCREMENT,  
    id_proyecto INT,  
    descripcion VARCHAR(255),
```

```
    responsable INT,  
    estado ENUM('pendiente', 'en progreso', 'completada'),  
    FOREIGN KEY (id_proyecto) REFERENCES Proyectos(id_proyecto),  
    FOREIGN KEY (responsable) REFERENCES Usuarios(id_usuario)  
);
```

```
CREATE TABLE Avances (  
    id_avance INT PRIMARY KEY AUTO_INCREMENT,  
    id_tarea INT,  
    descripcion TEXT,  
    fecha DATE,  
    FOREIGN KEY (id_tarea) REFERENCES Tareas(id_tarea)  
);
```

RED SOCIAL

```
CREATE DATABASE IF NOT EXISTS RedSocial;  
USE RedSocial;
```

```
CREATE TABLE Usuarios (  
    id_usuario INT PRIMARY KEY AUTO_INCREMENT,  
    nombre VARCHAR(100),  
    correo VARCHAR(100) UNIQUE  
);
```

```
CREATE TABLE Publicaciones (  
    id_publicacion INT PRIMARY KEY AUTO_INCREMENT,  
    id_usuario INT,  
    contenido TEXT,  
    fecha_publicacion DATETIME,  
    FOREIGN KEY (id_usuario) REFERENCES Usuarios(id_usuario)  
);
```

```
CREATE TABLE Comentarios (  
    id_comentario INT PRIMARY KEY AUTO_INCREMENT,  
    id_publicacion INT,  
    id_usuario INT,  
    contenido TEXT,  
    fecha_comentario DATETIME,
```

```
FOREIGN KEY (id_publicacion) REFERENCES Publicaciones(id_publicacion),  
FOREIGN KEY (id_usuario) REFERENCES Usuarios(id_usuario)  
);
```

```
CREATE TABLE Reacciones (  
    id_reaccion INT PRIMARY KEY AUTO_INCREMENT,  
    id_publicacion INT,  
    id_usuario INT,  
    tipo ENUM('me gusta', 'me encanta', 'me enoja', 'me entristece'),  
    fecha_reaccion DATETIME,  
    FOREIGN KEY (id_publicacion) REFERENCES Publicaciones(id_publicacion),  
    FOREIGN KEY (id_usuario) REFERENCES Usuarios(id_usuario)  
);
```

```
CREATE TABLE Seguidores (  
    id_seguidor INT,  
    id_seguido INT,  
    PRIMARY KEY (id_seguidor, id_seguido),  
    FOREIGN KEY (id_seguidor) REFERENCES Usuarios(id_usuario),  
    FOREIGN KEY (id_seguido) REFERENCES Usuarios(id_usuario)  
);
```