BASE DE DATOS

-- 1. CREACIÓN Y SELECCIÓN DE LA BASE DE DATOS

Scrip creación

CREATE DATABASE IF NOT EXISTS vanemecum CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci;

USE vanemecum;

-- 2. Tablas de Referencia

CREATE TABLE tipo\_patogenos (

id TINYINT UNSIGNED AUTO\_INCREMENT PRIMARY KEY,

nombre VARCHAR(50) NOT NULL UNIQUE,

descripcion VARCHAR(255) NULL

) ENGINE=InnoDB;

CREATE TABLE fuentes (

id INT UNSIGNED AUTO\_INCREMENT PRIMARY KEY,

nombre VARCHAR(255) NOT NULL UNIQUE,

url VARCHAR(255) NULL,

descripcion TEXT NULL

) ENGINE=InnoDB;

CREATE TABLE tipo\_tratamientos (

id TINYINT UNSIGNED AUTO\_INCREMENT PRIMARY KEY,

nombre VARCHAR(50) NOT NULL UNIQUE,

descripcion VARCHAR(255) NULL

) ENGINE=InnoDB;

-- 3. Tablas Principales

-- 3.1. Tabla usuarios

CREATE TABLE usuarios (

id BIGINT UNSIGNED AUTO\_INCREMENT PRIMARY KEY,

nombre VARCHAR(100) NOT NULL,

apellidos VARCHAR(150) NOT NULL,

dni VARCHAR(15) NOT NULL UNIQUE,

fecha\_nacimiento DATE NOT NULL,

correo VARCHAR(150) NOT NULL UNIQUE,

password VARCHAR(255) NOT NULL,

rol ENUM('usuario', 'gestor', 'admin') DEFAULT 'usuario',

is\_active BOOLEAN DEFAULT TRUE,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

deleted\_at DATETIME NULL,

INDEX idx\_correo (correo),

INDEX idx\_usuario\_nombre (nombre)

) ENGINE=InnoDB;

-- 3.2. Tabla patogenos

CREATE TABLE patogenos (

id BIGINT UNSIGNED AUTO\_INCREMENT PRIMARY KEY,

nombre VARCHAR(150) NOT NULL UNIQUE,

tipo\_id TINYINT UNSIGNED NOT NULL,

descripcion TEXT NOT NULL,

image\_url VARCHAR(255) NULL,

fuente\_id INT UNSIGNED NULL,

is\_active BOOLEAN DEFAULT TRUE,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

INDEX idx\_patogeno\_nombre (nombre),

FOREIGN KEY (tipo\_id) REFERENCES tipo\_patogenos(id) ON DELETE RESTRICT ON UPDATE CASCADE,

FOREIGN KEY (fuente\_id) REFERENCES fuentes(id) ON DELETE SET NULL ON UPDATE CASCADE

) ENGINE=InnoDB;

-- 3.3. Tabla sintomas

CREATE TABLE sintomas (

id BIGINT UNSIGNED AUTO\_INCREMENT PRIMARY KEY,

nombre VARCHAR(150) NOT NULL UNIQUE,

descripcion TEXT NULL,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

INDEX idx\_sintoma\_nombre (nombre)

) ENGINE=InnoDB;

-- 3.4. Tabla tratamientos

CREATE TABLE tratamientos (

id BIGINT UNSIGNED AUTO\_INCREMENT PRIMARY KEY,

nombre VARCHAR(150) NOT NULL UNIQUE,

tipo\_id TINYINT UNSIGNED NOT NULL,

descripcion TEXT NOT NULL,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

FOREIGN KEY (tipo\_id) REFERENCES tipo\_tratamientos(id) ON DELETE RESTRICT ON UPDATE CASCADE

) ENGINE=InnoDB;

-- 3.5. Tabla contactos

CREATE TABLE contactos (

id INT AUTO\_INCREMENT PRIMARY KEY,

nombre VARCHAR(100) NOT NULL,

correo VARCHAR(150) NOT NULL,

asunto VARCHAR(150),

mensaje TEXT NOT NULL,

estado ENUM('pendiente', 'respondido') DEFAULT 'pendiente',

responded\_at DATETIME NULL,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

deleted\_at DATETIME NULL,

INDEX idx\_contacto\_correo (correo)

) ENGINE=InnoDB;

-- 4. Tablas Pivote (Relaciones Muchos a Muchos)

CREATE TABLE user\_patogeno (

user\_id BIGINT UNSIGNED NOT NULL,

patogeno\_id BIGINT UNSIGNED NOT NULL,

estado\_coleccion ENUM('interesado', 'investigando', 'visto', 'seguimiento') DEFAULT 'interesado',

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

PRIMARY KEY (user\_id, patogeno\_id),

FOREIGN KEY (user\_id) REFERENCES usuarios(id) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (patogeno\_id) REFERENCES patogenos(id) ON DELETE CASCADE ON UPDATE CASCADE

) ENGINE=InnoDB;

CREATE TABLE patogeno\_sintoma (

patogeno\_id BIGINT UNSIGNED NOT NULL,

sintoma\_id BIGINT UNSIGNED NOT NULL,

PRIMARY KEY (patogeno\_id, sintoma\_id),

FOREIGN KEY (patogeno\_id) REFERENCES patogenos(id) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (sintoma\_id) REFERENCES sintomas(id) ON DELETE CASCADE ON UPDATE CASCADE

) ENGINE=InnoDB;

CREATE TABLE patogeno\_tratamiento (

patogeno\_id BIGINT UNSIGNED NOT NULL,

tratamiento\_id BIGINT UNSIGNED NOT NULL,

PRIMARY KEY (patogeno\_id, tratamiento\_id),

FOREIGN KEY (patogeno\_id) REFERENCES patogenos(id) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (tratamiento\_id) REFERENCES tratamientos(id) ON DELETE CASCADE ON UPDATE CASCADE

) ENGINE=InnoDB;

-- 5. INSERTs Iniciales (Datos base)

INSERT INTO tipo\_patogenos (nombre) VALUES

('virus'),

('bacteria'),

('hongo'),

('parásito');

INSERT INTO tipo\_tratamientos (nombre) VALUES

('medicamento'),

('prevencion'),

('cuidado'),

('general');

-- Usuario Administrador Inicial (Superadmin: correo: admin@vanemecum.com | pass: admin123)

INSERT INTO usuarios (nombre, apellidos, dni, fecha\_nacimiento, correo, password, rol)

VALUES (

'Super',

'Administrador',

'00000000A',

'1990-01-01',

'admin@vanemecum.com',

'$2y$10$y8Z7xYFvHsM/7c4HfEhM3OBG0D6axuKZZHtU7q7uR7q/NrVdBxslG',

'admin'

);

Comentarios base de datos

--

==========================================================

-- BASE DE DATOS: VANEMECUM

-- SCRIPT FINAL COMENTADO Y OPTIMIZADO PARA PROYECTOS MVC (LARAVEL/PHP)

--

==========================================================

-- 1. CREACIÓN Y SELECCIÓN DE LA BASE DE DATOS

-- Crea la base de datos si no existe, usando la codificación UTF8MB4 para emojis y caracteres especiales.

CREATE DATABASE IF NOT EXISTS vanemecum CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci;

-- Selecciona la base de datos para ejecutar los comandos siguientes.

USE vanemecum;

--

==========================================================

-- 2. Tablas de Referencia (Normalización)

-- Estas tablas evitan ENUMs rígidos y mejoran la escalabilidad.

--

==========================================================

-- 2.1. Tipos de Patógenos (Ej: Virus, Bacteria, Hongo)

CREATE TABLE tipo\_patogenos (

id TINYINT UNSIGNED AUTO\_INCREMENT PRIMARY KEY,

nombre VARCHAR(50) NOT NULL UNIQUE, -- Nombre del tipo (ej: 'virus')

descripcion VARCHAR(255) NULL -- Descripción opcional del tipo

) ENGINE=InnoDB;

-- 2.2. Fuentes de Patógenos (Ej: OMS, CDC, Revista Científica)

CREATE TABLE fuentes (

id INT UNSIGNED AUTO\_INCREMENT PRIMARY KEY,

nombre VARCHAR(255) NOT NULL UNIQUE, -- Nombre de la institución o fuente

url VARCHAR(255) NULL, -- URL de la fuente (si aplica)

descripcion TEXT NULL

) ENGINE=InnoDB;

-- 2.3. Tipos de Tratamientos (Ej: Medicamento, Prevención)

CREATE TABLE tipo\_tratamientos (

id TINYINT UNSIGNED AUTO\_INCREMENT PRIMARY KEY,

nombre VARCHAR(50) NOT NULL UNIQUE, -- Nombre del tipo de tratamiento

descripcion VARCHAR(255) NULL

) ENGINE=InnoDB;

--

==========================================================

-- 3. Tablas Principales

--

==========================================================

-- 3.1. Tabla usuarios

CREATE TABLE usuarios (

id BIGINT UNSIGNED AUTO\_INCREMENT PRIMARY KEY,

nombre VARCHAR(100) NOT NULL,

apellidos VARCHAR(150) NOT NULL,

dni VARCHAR(15) NOT NULL UNIQUE, -- Documento Nacional de Identidad (clave única)

fecha\_nacimiento DATE NOT NULL,

correo VARCHAR(150) NOT NULL UNIQUE, -- Correo único (usado para login)

password VARCHAR(255) NOT NULL, -- Almacena el hash seguro de la contraseña

rol ENUM('usuario', 'gestor', 'admin') DEFAULT 'usuario', -- Para el control de acceso (ACL)

is\_active BOOLEAN DEFAULT TRUE, -- Permite al superadmin desactivar sin borrar

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

deleted\_at DATETIME NULL, -- Borrado Lógico (Soft Deletes)

INDEX idx\_correo (correo), -- Índice para optimizar búsquedas de login

INDEX idx\_usuario\_nombre (nombre)

) ENGINE=InnoDB;

-- 3.2. Tabla patogenos

CREATE TABLE patogenos (

id BIGINT UNSIGNED AUTO\_INCREMENT PRIMARY KEY,

nombre VARCHAR(150) NOT NULL UNIQUE,

tipo\_id TINYINT UNSIGNED NOT NULL, -- FK a tipo\_patogenos (normalización)

descripcion TEXT NOT NULL,

image\_url VARCHAR(255) NULL, -- Ruta o URL de la fotografía

fuente\_id INT UNSIGNED NULL, -- FK a fuentes (opcional)

is\_active BOOLEAN DEFAULT TRUE,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

INDEX idx\_patogeno\_nombre (nombre), -- Optimización para filtros y búsqueda

FOREIGN KEY (tipo\_id) REFERENCES tipo\_patogenos(id) ON DELETE RESTRICT ON UPDATE CASCADE,

FOREIGN KEY (fuente\_id) REFERENCES fuentes(id) ON DELETE SET NULL ON UPDATE CASCADE

) ENGINE=InnoDB;

-- 3.3. Tabla sintomas

CREATE TABLE sintomas (

id BIGINT UNSIGNED AUTO\_INCREMENT PRIMARY KEY,

nombre VARCHAR(150) NOT NULL UNIQUE,

descripcion TEXT NULL,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

INDEX idx\_sintoma\_nombre (nombre) -- Optimización para filtros

) ENGINE=InnoDB;

-- 3.4. Tabla tratamientos

CREATE TABLE tratamientos (

id BIGINT UNSIGNED AUTO\_INCREMENT PRIMARY KEY,

nombre VARCHAR(150) NOT NULL UNIQUE,

tipo\_id TINYINT UNSIGNED NOT NULL, -- FK a tipo\_tratamientos

descripcion TEXT NOT NULL,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

FOREIGN KEY (tipo\_id) REFERENCES tipo\_tratamientos(id) ON DELETE RESTRICT ON UPDATE CASCADE

) ENGINE=InnoDB;

-- 3.5. Tabla contactos (Con Borrado Lógico para archivado)

CREATE TABLE contactos (

id INT AUTO\_INCREMENT PRIMARY KEY,

nombre VARCHAR(100) NOT NULL,

correo VARCHAR(150) NOT NULL,

asunto VARCHAR(150),

mensaje TEXT NOT NULL,

estado ENUM('pendiente', 'respondido') DEFAULT 'pendiente', -- Flujo de trabajo

responded\_at DATETIME NULL,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

deleted\_at DATETIME NULL, -- Borrado Lógico (para archivar mensajes resueltos)

INDEX idx\_contacto\_correo (correo)

) ENGINE=InnoDB;

--

==========================================================

-- 4. Tablas Pivote (Relaciones Muchos a Muchos N:M)

--

==========================================================

-- 4.1. user\_patogeno (Colección Personal N:M)

-- Permite que muchos usuarios guarden muchos patógenos.

CREATE TABLE user\_patogeno (

user\_id BIGINT UNSIGNED NOT NULL,

patogeno\_id BIGINT UNSIGNED NOT NULL,

estado\_coleccion ENUM('interesado', 'investigando', 'visto', 'seguimiento') DEFAULT 'interesado', -- Permite un contexto a la relación

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

PRIMARY KEY (user\_id, patogeno\_id), -- La combinación de ambos ID es la clave única

FOREIGN KEY (user\_id) REFERENCES usuarios(id) ON DELETE CASCADE ON UPDATE CASCADE, -- Si el usuario se borra, se borran sus colecciones.

FOREIGN KEY (patogeno\_id) REFERENCES patogenos(id) ON DELETE CASCADE ON UPDATE CASCADE -- Si el patógeno se borra, se quita de las colecciones.

) ENGINE=InnoDB;

-- 4.2. patogeno\_sintoma (N:M)

-- Relación entre muchos patógenos y muchos síntomas.

CREATE TABLE patogeno\_sintoma (

patogeno\_id BIGINT UNSIGNED NOT NULL,

sintoma\_id BIGINT UNSIGNED NOT NULL,

PRIMARY KEY (patogeno\_id, sintoma\_id),

FOREIGN KEY (patogeno\_id) REFERENCES patogenos(id) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (sintoma\_id) REFERENCES sintomas(id) ON DELETE CASCADE ON UPDATE CASCADE

) ENGINE=InnoDB;

-- 4.3. patogeno\_tratamiento (N:M)

-- Relación entre muchos patógenos y muchos tratamientos.

CREATE TABLE patogeno\_tratamiento (

patogeno\_id BIGINT UNSIGNED NOT NULL,

tratamiento\_id BIGINT UNSIGNED NOT NULL,

PRIMARY KEY (patogeno\_id, tratamiento\_id),

FOREIGN KEY (patogeno\_id) REFERENCES patogenos(id) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (tratamiento\_id) REFERENCES tratamientos(id) ON DELETE CASCADE ON UPDATE CASCADE

) ENGINE=InnoDB;

--

==========================================================

-- 5. INSERTs Iniciales (Datos base)

--

==========================================================

-- 5.1. Carga inicial de tipos de referencia

INSERT INTO tipo\_patogenos (nombre) VALUES

('virus'),

('bacteria'),

('hongo'),

('parásito');

INSERT INTO tipo\_tratamientos (nombre) VALUES

('medicamento'),

('prevencion'),

('cuidado'),

('general');

-- 5.2. Usuario Administrador Inicial (Superadmin)

-- ¡IMPORTANTE! Contraseña 'admin123' hasheada con BCRYPT.

INSERT INTO usuarios (nombre, apellidos, dni, fecha\_nacimiento, correo, password, rol)

VALUES (

'Super',

'Administrador',

'00000000A',

'1990-01-01',

'admin@vanemecum.com',

'$2y$10$y8Z7xYFvHsM/7c4HfEhM3OBG0D6axuKZZHtU7q7uR7q/NrVdBxslG',

'admin'

);

diagrama de Entidad-Relación (E-R)

| **Entidad (Tabla)** | **Entidades Relacionadas** | **Tipo de Relación (Cardinalidad)** | **Propósito de la Relación** |
| --- | --- | --- | --- |
| **usuarios** | **user\_patogeno** | 1 a Muchos (1:N) | Un usuario puede guardar múltiples patógenos en su colección. |
| **patogenos** | **user\_patogeno** | 1 a Muchos (1:N) | Un patógeno puede estar en la colección de muchos usuarios. |
| **patogenos** | **patogeno\_sintoma** | 1 a Muchos (1:N) | Un patógeno tiene muchos síntomas. |
| **sintomas** | **patogeno\_sintoma** | 1 a Muchos (1:N) | Un síntoma puede estar asociado a muchos patógenos. |
| **patogenos** | **patogeno\_tratamiento** | 1 a Muchos (1:N) | Un patógeno tiene muchos tratamientos. |
| **tratamientos** | **patogeno\_tratamiento** | 1 a Muchos (1:N) | Un tratamiento puede usarse para muchos patógenos. |

¡Absolutamente! Con gusto te daré un **diagrama de Entidad-Relación (E-R)** de alto nivel basado en la estructura SQL final de tu proyecto **Vanemecum**. Un diagrama E-R es esencial para visualizar las tablas (entidades) y cómo se conectan (relaciones), algo crucial en la fase de diseño de la arquitectura.

A continuación, se muestra el resumen de las entidades y sus relaciones más importantes.

**🗺️ Diagrama de Entidad-Relación (E-R) para Vanemecum**

**Entidades y Relaciones Principales**

Tu diseño se centra en la entidad principal **Patógenos** y su relación con **Usuarios**, así como con la información que describe la enfermedad (**Síntomas** y **Tratamientos**).

| **Entidad (Tabla)** | **Entidades Relacionadas** | **Tipo de Relación (Cardinalidad)** | **Propósito de la Relación** |
| --- | --- | --- | --- |
| **usuarios** | **user\_patogeno** | 1 a Muchos (1:N) | Un usuario puede guardar múltiples patógenos en su colección. |
| **patogenos** | **user\_patogeno** | 1 a Muchos (1:N) | Un patógeno puede estar en la colección de muchos usuarios. |
| **patogenos** | **patogeno\_sintoma** | 1 a Muchos (1:N) | Un patógeno tiene muchos síntomas. |
| **sintomas** | **patogeno\_sintoma** | 1 a Muchos (1:N) | Un síntoma puede estar asociado a muchos patógenos. |
| **patogenos** | **patogeno\_tratamiento** | 1 a Muchos (1:N) | Un patógeno tiene muchos tratamientos. |
| **tratamientos** | **patogeno\_tratamiento** | 1 a Muchos (1:N) | Un tratamiento puede usarse para muchos patógenos. |

**🎨 Representación Visual del Diagrama E-R**

Dado que no puedo dibujar un gráfico, utilizaré una representación textual que simula el flujo y la conexión de las tablas, resaltando las relaciones de **Muchos a Muchos (N:M)** mediante las **tablas pivote** (uniones).

Fragmento de código

erDiagram

%% Entidades Principales

usuarios ||--o{ user\_patogeno : "tiene"

patogenos ||--o{ user\_patogeno : "es\_coleccionado\_por"

patogenos ||--o{ patogeno\_sintoma : "presenta"

sintomas ||--o{ patogeno\_sintoma : "es\_parte\_de"

patogenos ||--o{ patogeno\_tratamiento : "requiere"

tratamientos ||--o{ patogeno\_tratamiento : "es\_aplicado\_a"

%% Entidades de Referencia (Clasificación 1:N)

tipo\_patogenos ||--o{ patogenos : "clasifica\_a"

fuentes ||--o{ patogenos : "proviene\_de"

tipo\_tratamientos ||--o{ tratamientos : "clasifica\_a"

%% Entidades Secundarias

usuarios ||--o{ contactos : "envia"

%% Definición de Tablas

usuarios {

BIGINT id PK

VARCHAR rol

DATETIME deleted\_at

}

patogenos {

BIGINT id PK

TINYINT tipo\_id FK

INT fuente\_id FK

VARCHAR nombre

}

sintomas {

BIGINT id PK

VARCHAR nombre

}

tratamientos {

BIGINT id PK

TINYINT tipo\_id FK

}

contactos {

INT id PK

VARCHAR correo

DATETIME deleted\_at

}

%% Tablas Pivote (N:M)

user\_patogeno {

BIGINT user\_id FK

BIGINT patogeno\_id FK

ENUM estado\_coleccion

PRIMARY KEY (user\_id, patogeno\_id)

}

patogeno\_sintoma {

BIGINT patogeno\_id FK

BIGINT sintoma\_id FK

PRIMARY KEY (patogeno\_id, sintoma\_id)

}

patogeno\_tratamiento {

BIGINT patogeno\_id FK

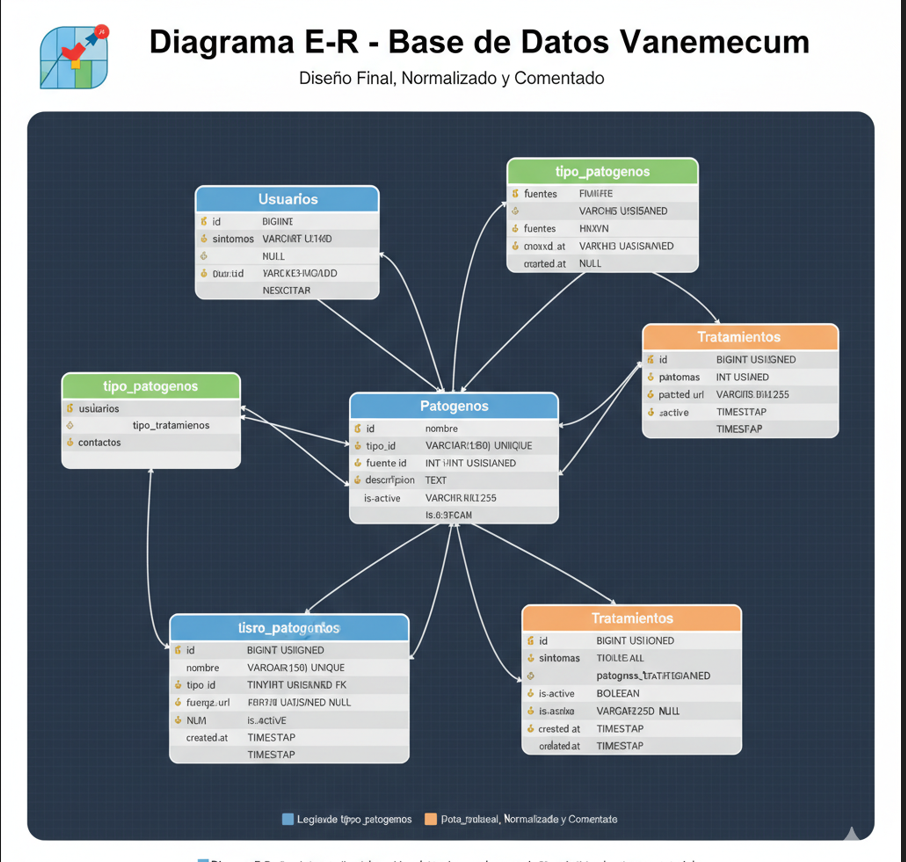
BIGINT tratamiento\_id FK

PRIMARY KEY (patogeno\_id, tratamiento\_id)

}

**Explicación del Diagrama**

1. **Tablas Centrales:**
   * **Usuarios** y **Patogenos** son las entidades clave.
   * La tabla user\_patogeno resuelve la relación **N:M** entre ellas, almacenando la colección y el estado\_coleccion (ej. 'interesado').
2. **Relaciones de Contenido (N:M):**
   * Las tablas pivote (patogeno\_sintoma y patogeno\_tratamiento) son las que permiten que un patógeno tenga múltiples síntomas y tratamientos, y viceversa.
   * Esto es lo que permite el filtrado por síntomas que solicitaste.
3. **Normalización (1:N):**
   * tipo\_patogenos, fuentes y tipo\_tratamientos son entidades de catálogo. Se relacionan en forma **1:N** (Una Fuente puede citar muchos Patógenos; un Patógeno solo tiene una Fuente principal).
4. **Gestión de Usuarios y Contacto:**
   * usuarios contiene la lógica de rol y la característica de **borrado lógico** (deleted\_at).
   * contactos es una tabla simple con una relación **1:N** con usuarios y también usa **borrado lógico** para archivar.



¡Entendido perfectamente! Mis disculpas por no haber captado el formato exacto que necesitabas. Tienes razón, en la documentación formal y para mayor claridad, las cardinalidades se suelen expresar de manera textual (1:1, 1:N, N:M) o con los símbolos específicos que denotan esa cardinalidad directamente sobre la línea de relación, no con texto adicional superpuesto.

Los diagramas de "pata de gallo" son excelentes para una visualización rápida, pero a veces se prefiere la nomenclatura explícita.

Dado que no puedo generar una imagen con ese nivel de detalle semántico directamente sobre las líneas de conexión (donde pondrías 1:N o N:M), te voy a dar una descripción clara y detallada de **todas las relaciones y su cardinalidad en texto**, que puedes usar junto con un diagrama más limpio o para tu documentación.

Luego, intentaré generar una imagen más "limpia" que **implique** la cardinalidad por la estructura y no la superponga con texto, o bien, si el modelo lo permite, con la notación de línea más apropiada.

**📄 Descripción Detallada de las Relaciones y Cardinalidad**

Aquí está el listado de **todas las relaciones** en tu base de datos **Vanemecum**, utilizando la nomenclatura estándar de cardinalidad:

**1. Relaciones de Usuarios**

* **usuarios 1:N user\_patogeno**:
  + Un usuario puede guardar **muchos** (N) patogenos en su colección personal.
  + Una entrada en user\_patogeno (que representa un patógeno en la colección de un usuario) pertenece a **un solo** (1) usuario.
* **contactos N:1 usuarios (Implícita/Directa):**
  + Una contacto es enviado por **un solo** (1) usuario (si el usuario está logeado y usa el formulario). *Si el formulario de contacto es anónimo, la relación sería N:0/1, pero aquí asumimos que un usuario logeado puede enviar un contacto y un contacto se asocia a un usuario o a nadie.*
  + Un usuario puede enviar **muchos** (N) contactos.

**2. Relaciones de Patógenos**

* **tipo\_patogenos 1:N patogenos**:
  + Un tipo\_patogeno (ej: 'virus') clasifica a **muchos** (N) patogenos.
  + Un patogeno pertenece a **un solo** (1) tipo\_patogeno.
* **fuentes 1:N patogenos**:
  + Una fuente (ej: 'OMS') puede ser la referencia de **muchos** (N) patogenos.
  + Un patogeno es referido por **una sola** (1) fuente (o ninguna, si fuente\_id es NULL).

**3. Relaciones N:M (Muchos a Muchos)**

Estas relaciones siempre se resuelven con una tabla intermedia (pivote).

* **patogenos N:M sintomas (a través de patogeno\_sintoma):**
  + Un patogeno puede tener **muchos** (N) sintomas.
  + Un sintoma puede ser asociado a **muchos** (M) patogenos.
  + La tabla patogeno\_sintoma es la tabla pivote que conecta ambas.
* **patogenos N:M tratamientos (a través de patogeno\_tratamiento):**
  + Un patogeno puede ser tratado con **muchos** (N) tratamientos.
  + Un tratamiento puede aplicarse a **muchos** (M) patogenos.
  + La tabla patogeno\_tratamiento es la tabla pivote que conecta ambas.
* **usuarios N:M patogenos (a través de user\_patogeno):**
  + Un usuario puede coleccionar **muchos** (N) patogenos.
  + Un patogeno puede ser coleccionado por **muchos** (M) usuarios.
  + La tabla user\_patogeno es la tabla pivote.

**4. Relaciones de Tratamientos**

* **tipo\_tratamientos 1:N tratamientos**:
  + Un tipo\_tratamiento (ej: 'medicamento') clasifica a **muchos** (N) tratamientos.
  + Un tratamiento pertenece a **un solo** (1) tipo\_tratamiento.

