

VANEMÉCUM

Enciclopedia Patogénica

PROYECTO FINAL

CICLO FORMATIVO DE GRADO SUPERIOR
DESARROLLO DE APLICACIONES WEB

Vanemecum

Guía de virus, bacterias y hongos

Alumno	Control de cambios	Curso	Centro	Año Académico
Vanessa Martínez Marín	Primera versión	2º D.A.W.	I.E.S. Arcipreste de Hita	2025/2026

Contenido

1. DEFINICIÓN DEL PROYECTO.....	4
1.1. Tipos de Empresas / Sectores Productivos	4
1.2. Descripción del problema	4
1.3. Objetivos. Especificación de Requisitos	5
1.3.1. Objetivos generales:	5
1.3.2. Objetivos específicos / personales:	5
1.3.3. Requisitos mínimos para utilizar el proyecto:	5
1.4. Herramientas de diseño	6
1.4.1. Herramientas usadas en el lado del cliente (front-end):	6
1.4.2. Herramientas usadas en el lado del servidor (back-end):.....	6
1.5. Estudio de viabilidad (técnica, económica, etc.)	7
2. PLANIFICACIÓN DEL PROYECTO	7
2.1. Secuenciación de las fases del proyecto	7
2.2. Planificación de recursos y tiempos	8
3. ANÁLISIS DEL PROYECTO	8
3.1. Casos de uso	8
3.2 Diagrama de actividades	9
3.3. Modelo de datos	10
3.3.1. Esquema Entidad / Relación	10
3.3.2. Modelo relacional y normalización	11

3.3.3. Diccionario de datos	12
4. DISEÑO DEL PROYECTO	16
4.1. <i>Diseño físico de la BD y estructura de los ficheros</i>	16
4.2. <i>Diseño de la estructura de clases y librerías (diagrama de clases)</i>	16
4.3. <i>Diseño de procedimientos especiales</i>	17
4.4. <i>Diseño de la interfaz gráfica</i>	19
5. DESARROLLO Y REALIZACIÓN DE PROYECTO	19
5.1. <i>Indicadores de calidad</i>	19
5.2. <i>Elaboración de una batería de pruebas para detectar errores</i>	19
5.2.1. REGISTRO:	20
5.2.2. INICIAR SESIÓN:	20
5.2.3. CONTACT:	21
5.2.4. FAVORITOS (MIS PATÓGENOS):	21
5.2.5. ADMIN – MENSAJES:	22
5.3. <i>Evaluación y solución de incidencias</i>	22
5.4. <i>Evaluación y seguimiento de las actividades</i>	23
6. IMPLANTACIÓN DEL PROYECTO	23
6.1. <i>Plan de implantación</i>	23
6.2. <i>Manual de instalación</i>	24
6.3. <i>Manual de usuario</i>	29
6.3.1. Usuario registrado (rol User)	29
6.3.2. Administrador (rol Admin)	37
7. CONCLUSIONES FINALES	42
7.1. <i>Evaluación del proyecto: objetivos conseguidos y no alcanzados</i>	42
7.1.1 <i>Objetivos conseguidos</i>	42
7.1.2 <i>Objetivos no alcanzados o pospuestos</i>	43
7.2. <i>Propuestas de mejora</i>	43

1. DEFINICIÓN DEL PROYECTO

Este proyecto se basa en una aplicación web denominada Vanemecum, que funciona como guía educativa de virus, bacterias y hongos. Permite a los usuarios consultar información sobre patógenos (microorganismos), sus síntomas asociados y tratamientos, y a los administradores gestionar todo el contenido, los mensajes de contacto y las alertas. La aplicación está desarrollada con Laravel (PHP) y MySQL.

1.1. Tipos de Empresas / Sectores Productivos

La aplicación está destinada al sector sanitario y educativo.

Puede ser utilizada por:

- Centros de salud y hospitales como referencia rápida (únicamente informativa, no diagnóstica).
- Universidades y centros de formación profesional, para apoyo docente en materias relacionadas con microbiología o enfermedades infecciosas.
- Cualquier empresa o entidad que necesite una guía de consulta sobre microorganismos en un formato web accesible y actualizable.

El diseño y la estructura pueden aplicarse a otros ámbitos que requieran catálogos con roles de usuario (consulta y administración).

1.2. Descripción del problema

Los usuarios necesitan acceder de forma rápida y fiable a información sobre virus, bacterias y hongos: descripción, síntomas habituales, tratamientos recomendados y fuentes de referencia.

Es imprescindible ofrecer esta información en un formato accesible desde navegador, ordenado (por tipo de patógeno, búsqueda por nombre) y mantenido por un administrador.

Por ello se ha desarrollado una aplicación web que actúa como intermediario entre el contenido técnico y el usuario: el usuario registrado consulta la guía, y el administrador crea, modifica y elimina patógenos, síntomas y tratamientos, y gestiona mensajes de contacto y alertas.

1.3. Objetivos. Especificación de Requisitos

1.3.1. Objetivos generales:

- Usuario (visitante registrado): Acceso a una página web clara y funcional para consultar la guía de patógenos (listado, ficha detallada con síntomas y tratamientos), guardar patógenos en “Mis patógenos” (favoritos), enviar mensajes de contacto (reportar error, sugerir patógeno, otro) y consultar tratamientos.
- Administrador: Acceso a un panel de administración para crear, modificar y eliminar patógenos (micros-db), síntomas y tratamientos, gestionar los mensajes de contacto (listado, detalle, marcar leído/no leído) y consultar el listado de alertas (patógenos con alerta activa).

1.3.2. Objetivos específicos / personales:

- Desarrollar una aplicación web completa con **Laravel** (PHP) y **MySQL**, utilizando buenas prácticas (migraciones, modelos, controladores, rutas, roles).
- Superar con buena nota el proyecto de fin de ciclo de Desarrollo de Aplicaciones Web.

1.3.3. Requisitos mínimos para utilizar el proyecto:

- **Conexión a internet:** para clonar el repositorio y, en producción, para acceder a la aplicación.
- **Sistemas operativos:** Windows 10 o posteriores, Linux o macOS. Se recomienda un entorno de desarrollo local (por ejemplo: XAMPP, Laravel Herd, o PHP/MySQL instalados por separado).
- **Hardware:** Procesador y RAM suficientes para ejecutar PHP, servidor web y MySQL (por ejemplo 2 GB RAM mínimo; 4 GB recomendado). Resolución mínima recomendada 1280×720 para una correcta visualización de la interfaz.
- **Programas instalados:**

- **PHP** 8.0.2 o superior (extensiones: PDO, MySQL, OpenSSL, Mbstring, etc.).
- **Composer** (gestor de dependencias PHP).
- **MySQL** (o MariaDB) para la base de datos.
- **Git** para clonar el repositorio.
- Navegador web actual (Chrome, Firefox, Edge, etc.).
- Opcional: **Node.js** y **npm** si se compilan assets con Vite/Tailwind.
- **Ficheros del proyecto:** “<https://github.com/virtualvmm/Vanemecum>”

1.4. Herramientas de diseño

1.4.1. Herramientas usadas en el lado del cliente (front-end):

- **Laravel Blade:** Motor de plantillas para generar las vistas HTML.
- **Tailwind CSS:** Framework CSS para el diseño de la interfaz (responsive, utilidades).
- **Vite** (opcional): Herramienta de construcción para compilar assets (CSS/JS).
- Recursos de iconos y estilo acorde a una aplicación de aspecto médico.

1.4.2. Herramientas usadas en el lado del servidor (back-end):

- **PHP 8.x:** Lenguaje de programación.
- **Laravel 9.x:** Framework PHP para rutas, controladores, modelos, autenticación, migraciones y vistas.
- **Laravel Breeze / Fortify:** Autenticación (registro, login, recuperación de contraseña, verificación de email).
- **MySQL:** Base de datos relacional. El diseño se realiza mediante **migraciones** de Laravel.
- **Composer:** Gestión de dependencias PHP.

1.5. Estudio de viabilidad (técnica, económica, etc.)

- **Viabilidad económica:** El proyecto es viable económicamente como aplicación educativa o interna: la parte de administración permite a la organización mantener el contenido sin depender constantemente de soporte externo. Solo sería necesario soporte externo en caso de fallos en el código o despliegue en producción.
- **Viabilidad técnica:** En el momento de planificación se dispone de los requisitos necesarios (PHP, Composer, MySQL, Laravel). El proyecto es viable técnicamente y permite aprender y aplicar el stack Laravel + MySQL de forma completa (modelos, relaciones, roles, formularios, correo).
- **Viabilidad legal:** La aplicación está destinada a uso informativo/educativo. Los datos mostrados (patógenos, síntomas, tratamientos) deben proceder de fuentes fiables y no sustituyen el criterio médico. No se almacenan datos sanitarios sensibles de pacientes.

2. PLANIFICACIÓN DEL PROYECTO

2.1. Secuenciación de las fases del proyecto

El proyecto se ha dividido en fases coherentes con el ciclo de vida del software:

- **Fase de análisis:** Definición de entidades (patógenos, tipos de patógeno, síntomas, tratamientos, tipos de tratamiento, usuarios, roles, mensajes de contacto, favoritos), relaciones entre ellas y flujos (guía pública, usuario autenticado, administración).
- **Fase de planificación:** División en dos grandes bloques: **funciones de usuario** (consulta guía, favoritos, contacto, perfil) y **funciones de administración** (CRUD patógenos, síntomas, tratamientos; mensajes; alertas). Asignación de prioridad a la base de datos, autenticación y roles, y después a cada módulo.
- **Fase de desarrollo y gestión** Implementación en Laravel (migraciones, modelos, controladores, rutas, vistas), con seguimiento en repositorio Git. Pruebas manuales

de cada flujo y corrección de incidencias (por ejemplo, el comportamiento “marcar como no leído” en la pantalla de detalle del mensaje).

2.2. Planificación de recursos y tiempos

- **Recursos:** Equipo de desarrollo, ordenador con PHP/MySQL/Composer, editor de código (por ejemplo VS Code o Cursor), repositorio Git (GitHub).
- **Tiempos:** Dependen del plan del ciclo formativo. Las tareas se han ido completando de forma iterativa (primero estructura y BD, después autenticación y roles, luego módulos de contenido y por último administración y ajustes).

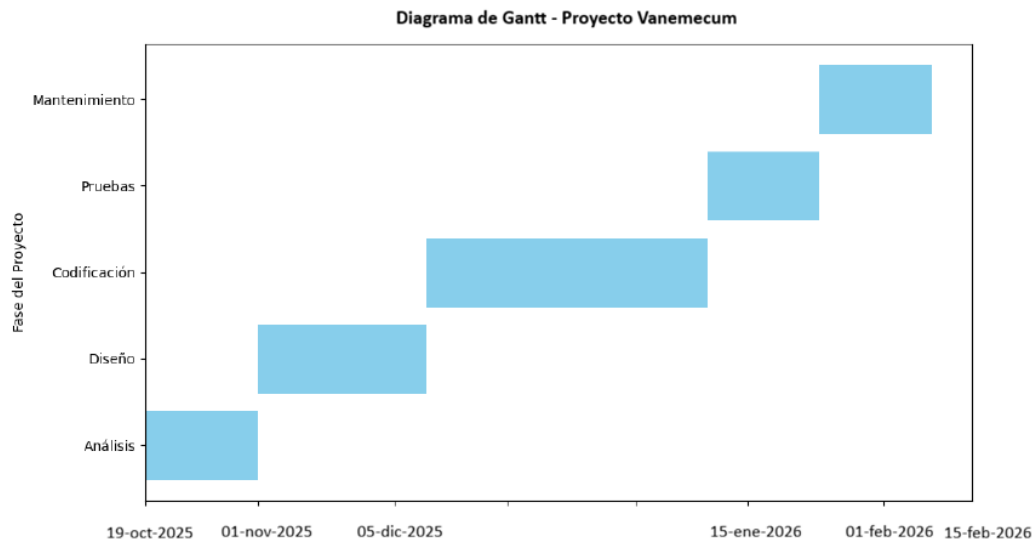


Figura 1 - Tabla de Gantt

3. ANÁLISIS DEL PROYECTO

3.1. Casos de uso

Los casos de uso se definen como las **acciones** que los distintos actores pueden realizar con la aplicación. Se distinguen tres tipos de actor:

- **Usuario registrado (rol User):** Acceder a página de Bienvenida, página de Tratamientos (consulta), página de Guía rápida (listado y ficha de patógeno),

página de Mis patógenos (añadir/quitar favoritos), página de Contacto (enviar mensaje), Perfil (editar datos, cerrar sesión).

- **Administrador (rol Admin):** Acceder a página de Bienvenida, página de Micros-DB (CRUD patógenos), página de Alertas (listado de patógenos con alerta activa), página de Tratamientos (CRUD tratamiento), página de Guía rápida (CRUD patógeno), página de Mis patógenos (añadir/quitar favoritos), página de Mensajes (listado, detalle, marcar leído/no leído), Perfil (editar datos, cerrar sesión).

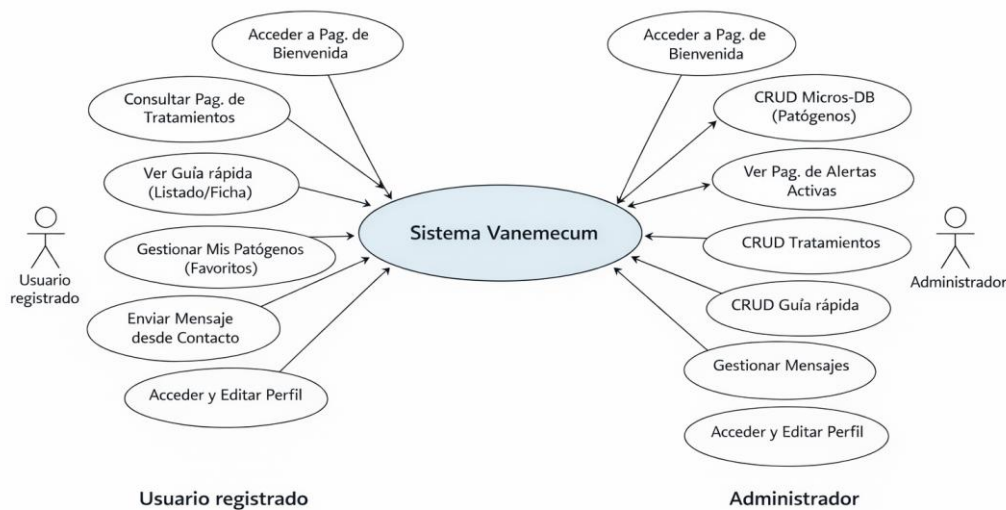


Figura 2 - Casos de uso

3.2 Diagrama de actividades

En el diagrama de actividades o flujo, describe el proceso de autenticación y la navegación principal diferenciada por rol (Usuario / Admin) en la aplicación Vanemecum.

Proyecto Final DAW – Vanemecum

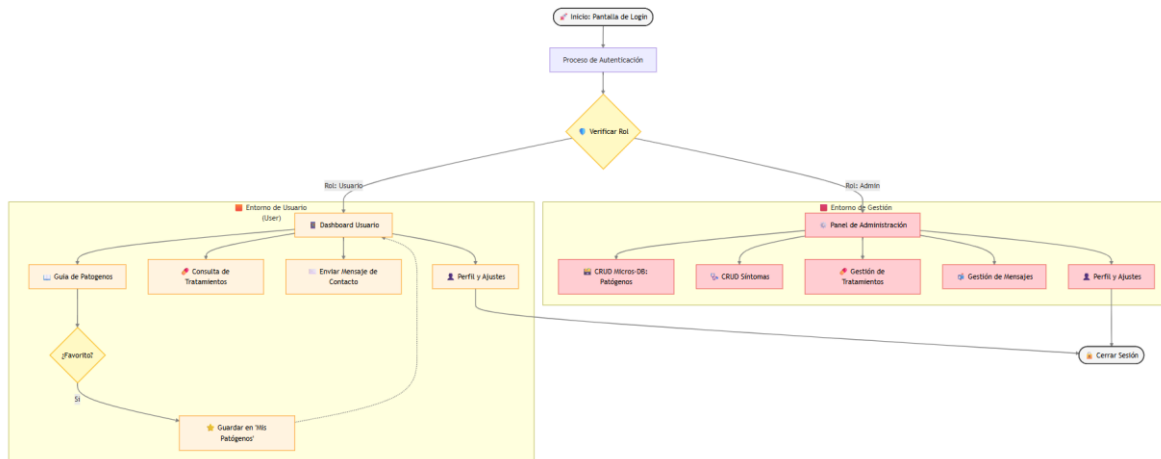


Figura 3 - Diagrama de actividades

3.3. Modelo de datos

3.3.1. Esquema Entidad / Relación

- Las entidades principales son:
 - Users
 - Roles
 - tipo_patogenos
 - fuentes_informacion
 - patógenos
 - sintomas
 - tipo_tratamientos
 - tratamientos
 - contact_messages
- Sus relaciones son:
 - users N:M roles (*role_user*).
 - users N:M patogenos (*patogeno_user*, favoritos).
 - patogenos N:1 tipo_patogenos y N:1 fuentes_informacion.
 - patogenos N:M sintomas (*patogeno_sintoma*) y N:M tratamientos (*patogeno_tratamiento*).

- tratamientos N:1 tipo_tratamientos; contact_messages N:1 users (user_id opcional).

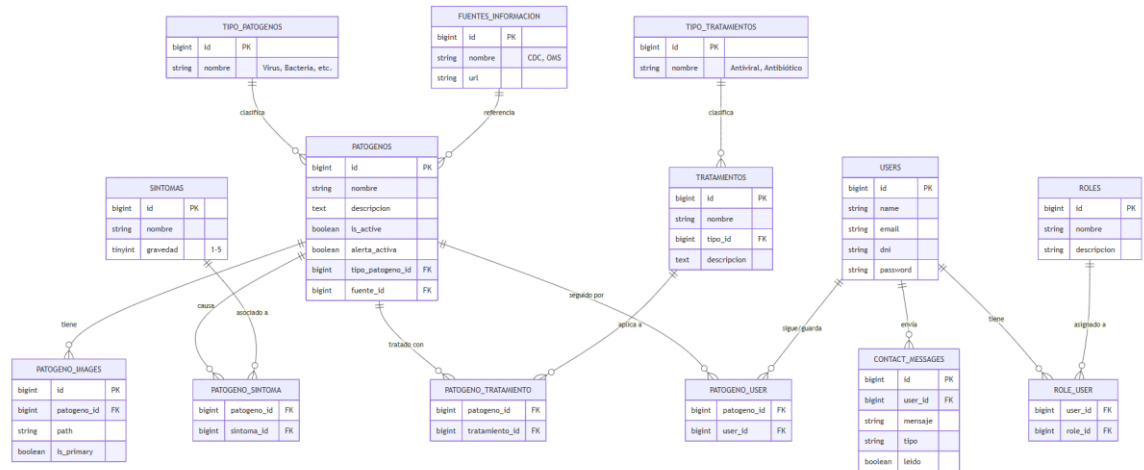


Figura 4 - Esquema E/R

3.3.2. Modelo relacional y normalización

El modelo relacional se ha implementado mediante **migraciones** de Laravel. Las tablas cumplen:

- **1FN:**
 - Valores atómicos, sin grupos repetitivos
 - Claves primarias definidas (id o claves compuestas en tablas pivote).
- **2FN:**
 - No hay dependencias parciales
 - En las tablas pivote los atributos dependen del par completo (`patogeno_id`, `sintoma_id`), etc.
- **3FN:**
 - No hay dependencias transitivas
 - El nombre del tipo de patógeno no se guarda en *patogenos*, sino en *tipo_patogenos* (patogenos solo almacena `tipo_patogeno_id`)
 - Igual para *tipo_tratamientos* y *fuentes_informacion*.

Proyecto Final DAW – Vanemecum

Tabla	1FN	2FN	3FN	BCNF	Justificación técnica
users	✓	✓	✓	✓	Atributos atómicos. Todos dependen de <code>id</code> . No dependencias transitivas.
roles	✓	✓	✓	✓	Tabla simple con clave primaria <code>id</code> . Sin redundancias.
role_user	✓	✓	✓	✓	Tabla pivote. Clave compuesta (<code>role_id</code> , <code>user_id</code>). Sin atributos adicionales dependientes parciales.
patogenos	✓	✓	✓	✓	Atributos describen únicamente al patógeno. Sin dependencias transitivas.
sintomas	✓	✓	✓	✓	Tabla independiente. Atributos atómicos.
tratamientos	✓	✓	✓	✓	Relación directa con tipo (FK). No hay datos redundantes.
patogeno_sintoma	✓	✓	✓	✓	Tabla N:M pura. Solo claves foráneas.
patogeno_tratamiento	✓	✓	✓	✓	Correcta resolución de relación N:M.
diagnosticos	✓	✓	✓	✓	Atributos dependen totalmente de su clave primaria.
alarmas	✓	✓	✓	✓	Datos dependientes solo de su PK.
fuentes_informacion	✓	✓	✓	✓	Entidad independiente. No redundancias.
contact_messages	✓	✓	⚠	⚠	Contiene <code>user_name</code> y <code>user_email</code> aunque existe <code>user_id</code> . Posible dependencia transitiva.
noticias	✓	✓	✓	✓	Datos propios de la noticia. Sin redundancias estructurales.

Figura 5 - Normalización

3.3.3. Diccionario de datos

Agrupando los datos por categorías:

- **USUARIO (users):** id, name, email, email_verified_at, password, remember_token, dni, telefono, direccion, created_at, updated_at.

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	Tipo de medio		
id (Primaria)	bigint(20)	No						
name	varchar(255)	No						
email	varchar(255)	No						
dni	varchar(20)	Si	NULL					
telefono	varchar(30)	Si	NULL					
direccion	varchar(255)	Si	NULL					
email_verified_at	timestamp	Si	NULL					
password	varchar(255)	No						
remember_token	varchar(100)	Si	NULL					
created_at	timestamp	Si	NULL					
updated_at	timestamp	Si	NULL					
Índices								
Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Si	No	id	2	A	No	
users_email_unique	BTREE	Si	No	email	2	A	No	

Figura 6 - Diccionario de datos (users)

Proyecto Final DAW – Vanemecum

- **ROL (roles):** id, nombre, descripcion, created_at, updated_at.

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	Tipo de medio		
id (Primaria)	bigint(20)	No						
nombre	varchar(50)	No			Nombre del rol, ej. Admin, Usuario, Médico.			
descripcion	varchar(255)	SI	NULL		Descripción detallada del rol.			
created_at	timestamp	SI	NULL					
updated_at	timestamp	SI	NULL					
índices								
Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	SI	No	id	2	A	No	
roles_nombre_unico	BTREE	SI	No	nombre	2	A	No	

Figura 7 - Diccionario de datos (roles)

- **ROLE_USER (role_user):** user_id (FK users), role_id (FK roles); PK (user_id, role_id).

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	Tipo de medio		
user_id (Primaria)	bigint(20)	No		users -> id				
role_id (Primaria)	bigint(20)	No		roles -> id				
Índices								
Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Si	No	user_id	3	A	No	
				role_id	3	A	No	
role_user_role_id_foreign	BTREE	No	No	role_id	3	A	No	

Figura 8 - Diccionario de datos (role_user)

- **TIPO_PATOGENO (tipo_patogenos):** id, nombre, descripcion, created_at, updated_at.

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	Tipo de medio		
id (Primaria)	bigint(20)	No						
nombre	varchar(100)	No						
descripcion	text	Si	NULL					
created_at	timestamp	Si	NULL					
updated_at	timestamp	Si	NULL					
Índices								
Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Si	No	id	5	A	No	
tipo_patogenos_nombre_unico	BTREE	Si	No	nombre	5	A	No	

Figura 9 - Diccionario de datos (tipo_patogeno)

- **FUENTE (fuentes_informacion):** id, nombre, url, descripcion, created_at, updated_at.

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	Tipo de medio		
id (Primaria)	bigint(20)	No						
nombre	varchar(255)	No						
url	varchar(500)	Si	NULL					
descripcion	text	Si	NULL					
created_at	timestamp	Si	NULL					
updated_at	timestamp	Si	NULL					
Índices								
Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Si	No	id	2	A	No	
fuentes_informacion_nombre_unico	BTREE	Si	No	nombre	2	A	No	

Figura 10 - Diccionario de datos (fuentes_informacion)

- **PATOGENO (patogenos):** id, nombre, descripcion, tipo_patogeno_id (FK), fuente_id (FK, nullable), image_url, is_active, alerta_activa, alerta_texto, created_at, updated_at.

Proyecto Final DAW – Vanemecum

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	Tipo de medio
id (Primaria)	bigint(20)	No				
nombre	varchar(255)	No				
descripcion	text	No				
image_url	varchar(255)	Si	NULL			
is_active	tinyint(1)	No	1			
alerta_activa	tinyint(1)	No	0			
alerta_texto	varchar(500)	Si	NULL			
tipo_patogeno_id	bigint(20)	No		tipo_patogenos -> id		
fuentes_id	bigint(20)	Si	NULL	fuentes_informacion -> id		
created_at	timestamp	Si	NULL			
updated_at	timestamp	Si	NULL			

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Si	No	id	52	A	No	
patogenos_nombre_unique	BTREE	Si	No	nombre	52	A	No	
patogenos_tipo_patogeno_id_foreign	BTREE	No	No	tipo_patogeno_id	8	A	No	
patogenos_fuentes_id_fk	BTREE	No	No	fuentes_id	2	A	Si	
patogenos_is_active_nombre_index	BTREE	No	No	is_active	2	A	No	
				nombre	52	A	No	

Figura 11 - Diccionario de datos (patogenos)

- **SINTOMA (síntomas):** id, nombre, gravedad, descripción, created_at, updated_at.

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	Tipo de medio
id (Primaria)	bigint(20)	No				
nombre	varchar(100)	No				
gravedad	tinyint(4)	No	1		Nivel de gravedad del síntoma (1=Leve, 5=Crítico)	
descripcion	text	Si	NULL			
created_at	timestamp	Si	NULL			
updated_at	timestamp	Si	NULL			

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Si	No	id	39	A	No	
sintomas_nombre_unique	BTREE	Si	No	nombre	39	A	No	

Figura 12 - Diccionario de datos (síntomas)

- **PATOGENO_SINTOMA (patogeno_sintoma):** patogeno_id (FK), sintoma_id (FK); PK compuesta.

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	Tipo de medio
patogeno_id (Primaria)	bigint(20)	No		patogenos -> id		
sintoma_id (Primaria)	bigint(20)	No		sintomas -> id		

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Si	No	patogeno_id	115	A	No	
				sintoma_id	230	A	No	
patogeno_sintoma_unique	BTREE	Si	No	patogeno_id	115	A	No	
				sintoma_id	230	A	No	
patogeno_sintoma_sintoma_id_foreign	BTREE	No	No	sintoma_id	28	A	No	

Figura 13 - Diccionario de datos (patogeno_sintoma)

- **TIPO_TRATAMIENTO (tipo_tratamientos):** id, nombre, descripcion, created_at, updated_at.

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	Tipo de medio
id (Primaria)	bigint(20)	No				
nombre	varchar(100)	No				
descripcion	text	Si	NULL			
created_at	timestamp	Si	NULL			
updated_at	timestamp	Si	NULL			

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Si	No	id	4	A	No	
tipo_tratamientos_nombre_unique	BTREE	Si	No	nombre	4	A	No	

Figura 14 - Diccionario de datos (tipo_tratamientos)

- **TRATAMIENTO (tratamientos):** id, nombre, descripcion, duracion_dias, tipo_id (FK, nullable), created_at, updated_at.

Proyecto Final DAW – Vanemecum

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	Tipo de medio
id (Primaria)	bigint(20)	No				
nombre	varchar(150)	No				
tipo_id	bigint(20)	Si	NULL	tipo_tratamientos -> id		
descripcion	text	No				
duracion_dias	smallint(5)	Si	NULL		Duración estimada del tratamiento en días	
created_at	timestamp	Si	NULL			
updated_at	timestamp	Si	NULL			

Índices								
Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Si	No	id	30	A	No	
tratamientos_nombre_unico	BTREE	Si	No	nombre	30	A	No	
tratamientos_tipo_id_foreign	BTREE	No	No	tipo_id	10	A	Si	

Figura 15 - Diccionario de datos (tratamientos)

- **PATOGENO_TRATAMIENTO (patogeno_tratamiento):** patogeno_id (FK), tratamiento_id (FK); PK compuesta.

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	Tipo de medio
patogeno_id (Primaria)	bigint(20)	No		patogenos -> id		
tratamiento_id (Primaria)	bigint(20)	No		tratamientos -> id		

Índices								
Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Si	No	patogeno_id	130	A	No	
				tratamiento_id	130	A	No	
patogeno_tratamiento_tratamiento_id_foreign	BTREE	No	No	tratamiento_id	65	A	No	

Figura 16 - Diccionario de datos (patogeno_tratamiento)

- **PATOGENO_USER (patogeno_user, favoritos):** patogeno_id (FK), user_id (FK), created_at, updated_at; PK compuesta.

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	Tipo de medio
patogeno_id (Primaria)	bigint(20)	No		patogenos -> id		
user_id (Primaria)	bigint(20)	No		users -> id		
created_at	timestamp	Si	NULL			
updated_at	timestamp	Si	NULL			

Índices								
Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Si	No	patogeno_id	1	A	No	
				user_id	1	A	No	
patogeno_user_user_id_foreign	BTREE	No	No	user_id	1	A	No	

Figura 17 - Diccionario de datos (patogeno_user, favoritos)

- **CONTACT_MESSAGE (contact_messages):** id, user_id (FK, nullable), user_name, user_email, tipo, mensaje, leído, created_at, updated_at.

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	Tipo de medio
id (Primaria)	bigint(20)	No				
user_id	bigint(20)	Si	NULL	users -> id		
user_name	varchar(255)	No				
user_email	varchar(255)	No				
tipo	varchar(50)	No				
mensaje	text	No				
leído	tinyint(1)	No	0			
created_at	timestamp	Si	NULL			
updated_at	timestamp	Si	NULL			

Índices								
Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Si	No	id	2	A	No	
contact_messages_user_id_foreign	BTREE	No	No	user_id	2	A	Si	
contact_messages_leído_index	BTREE	No	No	leído	2	A	No	
contact_messages_created_at_index	BTREE	No	No	created_at	2	A	Si	

Figura 18 - Diccionario de datos (contact_messages)

4. DISEÑO DEL PROYECTO

4.1. Diseño físico de la BD y estructura de los ficheros

A continuación, se listan las tablas con sus columnas, tipo y claves (PK/FK) tal como están en la aplicación.

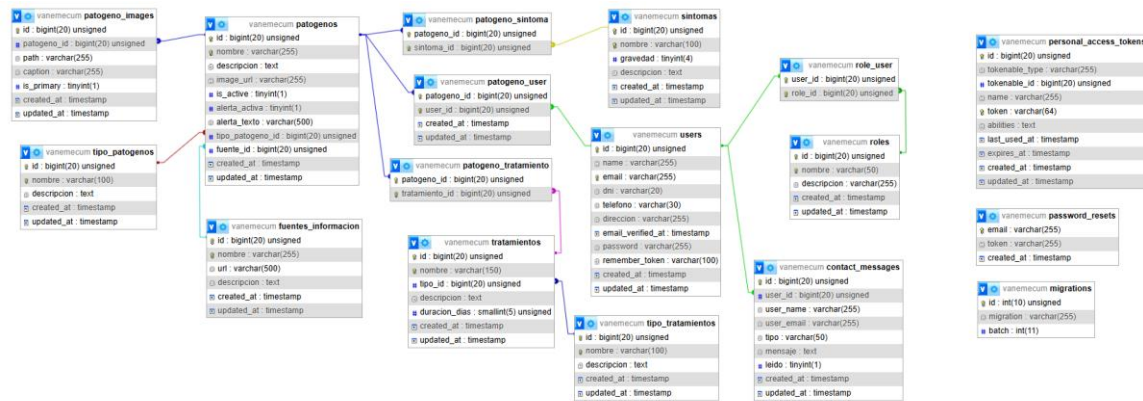


Figura 19 - Diseño físico de BD con tablas de Laravel

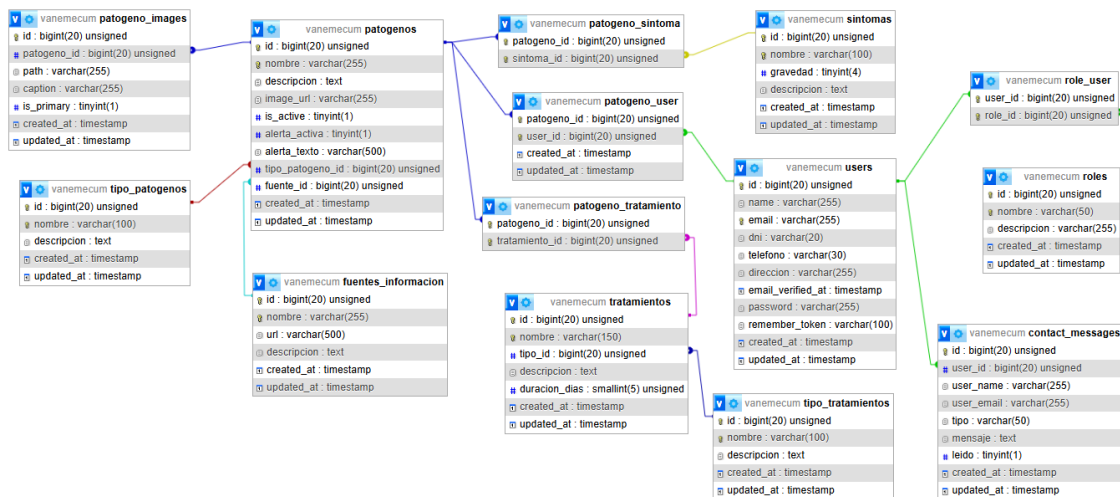


Figura 20 - Diseño físico de BD

4.2. Diseño de la estructura de clases y librerías (diagrama de clases)

- Los **modelos** de Laravel (Eloquent) representan las entidades: `User`, `Rol`, `Patogeno`, `TipoPatogeno`, `Fuente`, `Sintoma`, `Tratamiento`, `TipoTratamiento`, `ContactMessage`.

- Las relaciones se implementan con métodos `belongsTo`, `belongsToMany`, etc.
- Los **controladore** principales son: `GuiaController`, `PatogenoController`, `TratamientoController`, `FavoritoController`, `ContactController`, `ProfileController`
 - En el espacio de nombres Admin: `SintomaController`, `ContactMessageController`, `AlertaController`.

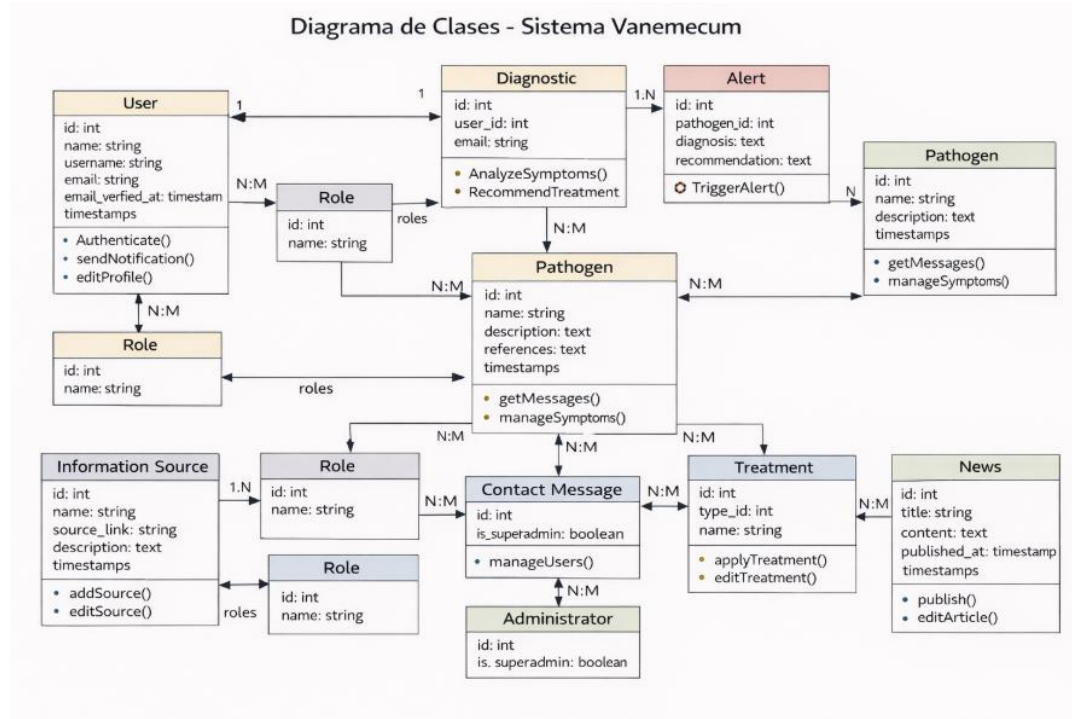


Figura 21 - Diseño de estructura de clases

4.3. Diseño de procedimientos especiales

En esta aplicación no se utilizan procedimientos almacenados en la base de datos, la lógica reside en los **controladores** y **modelos** de Laravel. Algunos ejemplos de “procedimientos” relevantes:

- **Consulta de patógenos para la guía:** En `GuiaController` se utiliza el modelo `Patogeno` con scopes como `activos()` y `filtradoGuia(\$search, \$tipoid)` para filtrar por nombre y tipo y ordenar por nombre.

```
// GuiaController::index() - fragmento
$query = $request->input('query');
$tipoId = $request->input('tipo');

$patogenos = Patogeno::with('tipo')
    ->activos()
    ->filtradoGuia($query, $tipoId)
    ->get();

$byTipo = $patogenos->groupBy(fn ($p) => $p->tipo->nombre ?? 'Otros');
```

Figura 22 - Consulta en GuiaController

- **Toggle leído en mensaje:** En `ContactMessageController::toggleLeido()` se actualiza el campo `leído` del mensaje y se redirige a la vista de detalle con el parámetro `from_toggle` para no volver a marcar como leído al cargar la pantalla.

```
// ContactMessageController::toggleLeido()
public function toggleLeido(ContactMessage $mensaje): RedirectResponse
{
    $mensaje->update(['leido' => ! $mensaje->leido]);
    $message = $mensaje->leido ? 'Mensaje marcado como leído.' : 'Mensaje marcado como no leído.';
    return redirect()->route('admin.mensajes.show', ['mensaje' => $mensaje, 'from_toggle' => 1])
        ->with('success', $message);
}

// ContactMessageController::show() - condición para no auto-marcar
$vieneDelToggle = request()->boolean('from_toggle');
if ($mensaje->exists && ! $mensaje->leido && ! $vieneDelToggle) {
    $mensaje->update(['leido' => true]);
}
```

Figura 23 - Toggle leído

- **Favoritos:** En `FavoritoController::store()` y `destroy()` se gestionan las filas de la tabla pivote `patogeno_user` mediante la relación `User::patogenos()`.

```
// FavoritoController::store() - añadir favorito
public function store(Request $request, Patogeno $patogeno)
{
    auth()->user()->patogenos()->syncWithoutDetaching([$patogeno->id]);
    return back()->with('success', __('Añadido a Mis patógenos.'));
}

// FavoritoController::destroy() - quitar favorito
public function destroy(Request $request, Patogeno $patogeno)
{
    auth()->user()->patogenos()->detach($patogeno->id);
    return back()->with('success', __('Quitado de Mis patógenos.'));
}
```

Figura 24 - Favoritos

4.4. Diseño de la interfaz gráfica

La interfaz se ha diseñado con vistas **Blade** y **Tailwind CSS**, con aspecto claro y orientado a uso médico/educativo. Incluye:

- Pantallas de login y registro (layout guest).
- Página de inicio (welcome), navegación común (logo, enlaces, menú usuario).
- Tratamientos (índice y ficha), Guía Rápida (índice y ficha de patógeno), Mis patógenos, Contactar, Usuario de pruebas (perfil y cerrar sesión).
- Panel admin: listados y formularios para Micros-DB, Síntomas, Tratamientos, Mensajes, Alertas.

5. DESARROLLO Y REALIZACIÓN DE PROYECTO

5.1. Indicadores de calidad

Se han realizado **pruebas manuales** por cada flujo principal de la aplicación: registro, login, recuperación de contraseña, consulta de guía y catálogo, búsqueda y filtros, favoritos (añadir y quitar), envío de formulario de contacto, y en el rol Admin: CRUD de patógenos, síntomas y tratamientos, listado y detalle de mensajes (incluyendo marcar como leído/no leído desde listado y desde la pantalla “Ver”), y listado de alertas. El objetivo es una aplicación **funcional, estable y controlada** ante entradas incorrectas (validación en formularios).

5.2. Elaboración de una batería de pruebas para detectar errores

A continuación, se describe una **batería de pruebas de caja negra** representativa. En la columna “Resultado” se indica OK si el comportamiento coincide con el esperado y KO en caso contrario.

5.2.1. REGISTRO:

Nombre función	Parámetros (ejemplo)	Valor obtenido	Valor esperado	Resultado	Observaciones
Registrar usuario	name, email, password válidos	Usuario creado, redirección	Registro correcto	OK	
Registrar usuario	email ya existente	Mensaje error / no validado	No permitir duplicado	OK	
Registrar usuario	Campos vacíos	Mensaje validación	No validado	OK	
Registrar usuario	Contraseña débil (ej. "123")	Mensaje validación	No validado	OK	Reglas de contraseña (mínimo, etc.)

Figura 25 - Prueba caja negra (registro)

5.2.2. INICIAR SESIÓN:

Nombre función	Parámetros	Valor obtenido	Valor esperado	Resultado	Observaciones
Login	email y password correctos	Acceso al dashboard	Validado	OK	
Login	email inexistente o password incorrecto	Mensaje error	No validado	OK	

Nombre función	Parámetros	Valor obtenido	Valor esperado	Resultado	Observaciones
Login	Campos vacíos	Mensaje validación	No validado	OK	

Figura 26 - Prueba caja negra (iniciar sesión)

5.2.3. CONTACT:

Nombre función	Parámetros	Valor obtenido	Valor esperado	Resultado	Observaciones
Enviar mensaje	Nombre, email, tipo, mensaje válidos	Mensaje enviado / guardado	Envío correcto	OK	Se guarda en contact_messages y se envía correo (o log si MAIL_MAILER=log)
Enviar mensaje	Campos obligatorios vacíos	Mensaje validación	No enviado	OK	

Figura 27 - Prueba caja negra (contact)

5.2.4. FAVORITOS (MIS PATÓGENOS):

Nombre función	Parámetros	Valor obtenido	Valor esperado	Resultado	Observaciones
Añadir favorito	Usuario autenticado, patógeno válido	Patógeno en Mis patógenos	Añadido	OK	Desde ficha de patógeno

Nombre función	Parámetros	Valor obtenido	Valor esperado	Resultado	Observaciones
Quitar favorito	Usuario con favorito	Patógeno quitado de la lista	Eliminado	OK	

Figura 28 - Prueba caja negra (favoritos)

5.2.5. ADMIN – MENSAJES:

Nombre función	Parámetros	Valor obtenido	Valor esperado	Resultado	Observaciones
Marcar como no leído desde listado	Admin, mensaje leído	Estado no leído	Toggle correcto	OK	
Marcar como no leído desde pantalla “Ver”	Admin, mensaje leído	Estado no leído al volver	Toggle correcto	OK	Corregido con redirección y parámetro from_toggle

Figura 29 - Prueba caja negra (admin - mensajes)

5.3. Evaluación y solución de incidencias

- **Incidencia resuelta (mensajes):** En la pantalla “Ver” de un mensaje de contacto, el botón “Marcar como no leído” no tenía efecto: al ejecutar el toggle, la aplicación redirigía de nuevo a la misma pantalla y el método `show()` volvía a marcar el mensaje como leído automáticamente.
 - **Solución:** En `toggleLeido()` se redirige a la ruta de detalle añadiendo el parámetro de consulta `from_toggle=1`; en `show()` solo se auto-marca

como leído cuando la petición no trae ese parámetro, de modo que el estado “no leído” se mantiene.

- Otras incidencias menores (responsive, textos, permisos) se han ido corrigiendo durante el desarrollo.

5.4. Evaluación y seguimiento de las actividades

Se ha llevado un seguimiento del avance mediante el repositorio Git (commits descriptivos) y pruebas manuales tras cada cambio relevante. En caso de detectar nuevos fallos (por ejemplo, en navegadores concretos o en móvil), se documentan y corrigen en la medida de lo posible.

6. IMPLANTACIÓN DEL PROYECTO

6.1. Plan de implantación

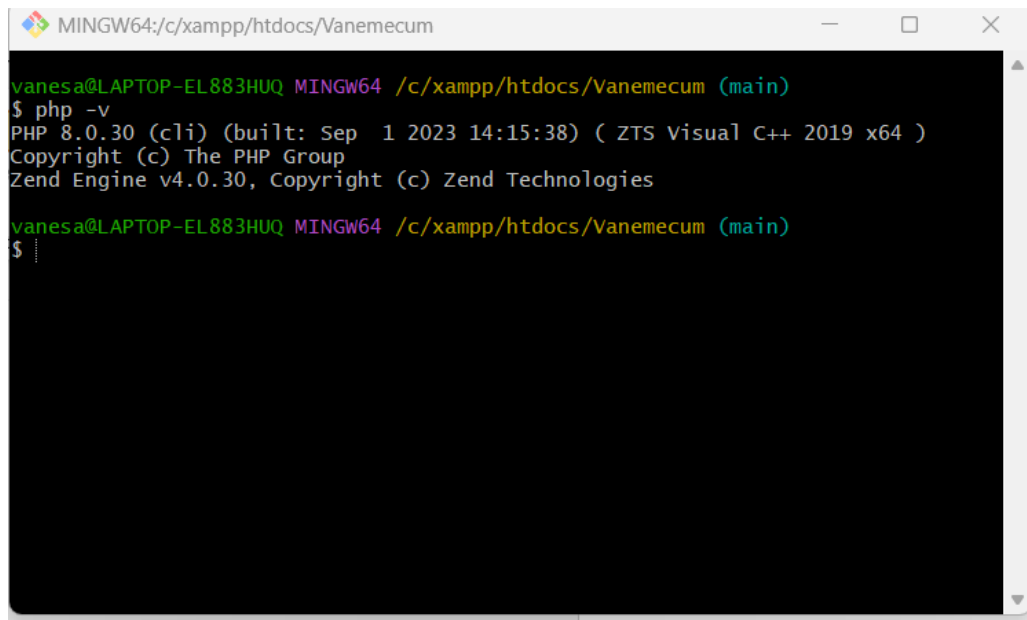
Para implantar la aplicación en un servidor (o en un equipo local):

1. Disponer de un entorno con **PHP 8.x**, **Composer**, **MySQL** y (opcionalmente) **Node/npm** para compilar assets.
2. Clonar el repositorio, instalar dependencias con Composer y configurar el archivo `.env` (base de datos, APP_KEY, correo, etc.).
3. Ejecutar migraciones y seeders para crear las tablas y datos iniciales (roles, tipos, etc.).
4. Configurar el servidor web (Apache/Nginx) con el document root en la carpeta `public` del proyecto, o usar `php artisan serve` en desarrollo.
5. Comprobar que la aplicación responde correctamente y que el administrador puede acceder al panel.

La implantación en un entorno nuevo puede requerir aproximadamente **medio día a un día** si ya están instalados PHP, Composer y MySQL.

6.2. Manual de instalación

- **Requisitos:**
 - XAMPP: Servidor Apache y MySQL para PHP.
 - PHP 8.1 o superior: Incluido en XAMPP, verificar con `php -v`.
 - Composer: Gestor de dependencias de PHP. Verifica con ``composer -V``.
 - Git: Control de versiones, verificar con `git --version`.
 - Opcional: Node.js y npm.
- **Pasos:**
 - **Clonar el repositorio**
 - **Abre Git Bash y ejecuta:** `cd C:/xampp/htdocs`
`git clone https://github.com/virtualvmm/Vanemecum.git`
`cd Vanemecum`
- **Instalar dependencias PHP**
 - **Instala Laravel y las dependencias del proyecto:** `composer install`



```
MINGW64:/c:/xampp/htdocs/Vanemecum
vanesa@LAPTOP-EL883HUQ MINGW64 /c:/xampp/htdocs/Vanemecum (main)
$ php -v
PHP 8.0.30 (cli) (built: Sep 1 2023 14:15:38) ( ZTS Visual C++ 2019 x64 )
Copyright (c) The PHP Group
Zend Engine v4.0.30, Copyright (c) Zend Technologies
vanesa@LAPTOP-EL883HUQ MINGW64 /c:/xampp/htdocs/Vanemecum (main)
$
```

Figura 30 - Instalar dependencias PHP (1)

Proyecto Final DAW – Vanemecum

```
vanesa@LAPTOP-EL883HUQ MINGW64 /c/xampp/htdocs/Vanemecum (main)
$ composer -V
Composer version 2.7.7 2024-06-10 22:11:12
PHP version 8.0.30 (C:\xampp\php\php.exe)
Run the "diagnose" command to get more detailed diagnostics output.

vanesa@LAPTOP-EL883HUQ MINGW64 /c/xampp/htdocs/Vanemecum (main)
$
```

Figura 31 - Instalar dependencias PHP (2)

```
vanesa@LAPTOP-EL883HUQ MINGW64 /c/xampp/htdocs/Vanemecum (main)
$ git -- version
unknown option: --
usage: git [-v | --version] [-h | --help] [-C <path>] [-c <name>=<value>]
          [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
          [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--no-lazy
          --fetch]
          [--no-optional-locks] [--no-advice] [--bare] [--git-dir=<path>]
          [--work-tree=<path>] [--namespace=<name>] [--config-env=<name>=<envva
          r>]
          <command> [<args>]

vanesa@LAPTOP-EL883HUQ MINGW64 /c/xampp/htdocs/Vanemecum (main)
$
```

Figura 32 - Instalar dependencias PHP (3)

```
bash: cd: /c/xampp/htdocs: No such file or directory

vanesa@LAPTOP-EL883HUQ MINGW64 /c/xampp/htdocs/Vanemecum
$ cd /c/xampp/htdocs

vanesa@LAPTOP-EL883HUQ MINGW64 /c/xampp/htdocs
$ composer create-project laravel/laravel VanemecumLara
Creating a "laravel/laravel" project at "./VanemecumLara
Cannot use laravel/laravel's latest version v12.7.1 as
by your platform.
Installing laravel/laravel (v9.5.2)
- Downloading laravel/laravel (v9.5.2)
- Installing laravel/laravel (v9.5.2): Extracting and
Created project in C:\xampp\htdocs\VanemecumLaravel
> @php -r "file_exists('.env') || copy('.env.example',
Loading composer repositories with package information

vanesa@LAPTOP-EL883HUQ MINGW64 /c/xampp/htdocs
$ AC

vanesa@LAPTOP-EL883HUQ MINGW64 /c/xampp/htdocs
```

Figura 33 - Instalar dependencias PHP (4)

```
thrown in C:\xampp\htdocs\VanemecumLaravel\artisan on line 18
vanesa@LAPTOP-EL883HUQ MINGW64 /c/xampp/htdocs/VanemecumLaravel
$ AC
vanesa@LAPTOP-EL883HUQ MINGW64 /c/xampp/htdocs/VanemecumLaravel
$ cd /c/xampp/htdocs/VanemecumLaravel
vanesa@LAPTOP-EL883HUQ MINGW64 /c/xampp/htdocs/VanemecumLaravel
$ composer install
No composer.lock file present. Updating dependencies to latest instead of installing from lock file. See https://getcomposer.org/install for more information.
Loading composer repositories with package information
```

Figura 34 - Instalar dependencias PHP (5)

- **Configurar entorno**

- Copiar el archivo ``.env.example`` a ``.env``.
- Ejecutar: ``php artisan key:generate``
- Editar ``.env`` y configurar:

DB_CONNECTION=mysql

DB_HOST=127.0.0.1

DB_PORT=3306

DB_DATABASE=vanemecum

DB_USERNAME=root

DB_PASSWORD=

```
vanesa@LAPTOP-EL883HUQ MINGW64 /c/xampp/htdocs/VanemecumLaravel
$ php artisan key:generate

  INFO  Application key set successfully.

vanesa@LAPTOP-EL883HUQ MINGW64 /c/xampp/htdocs/VanemecumLaravel
$
```

Figura 35 - Configurar entorno (1)

```
APP_NAME=Laravel
APP_ENV=local
APP_KEY=base64:LYWFf0ZhydJHb/vU//dkZ0aXmp5GQHyySXXlimtthIc=
APP_DEBUG=true
APP_URL=http://localhost

LOG_CHANNEL=stack
LOG_DEPRECATIONS_CHANNEL=null
LOG_LEVEL=debug

DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=laravel
DB_USERNAME=root
DB_PASSWORD=

BROADCAST_DRIVER=log
CACHE_DRIVER=file
FILESYSTEM_DISK=local
QUEUE_CONNECTION=sync
SESSION_DRIVER=file
SESSION_LIFETIME=120

MEMCACHED_HOST=127.0.0.1

REDIS_HOST=127.0.0.1
REDIS_PASSWORD=null
REDIS_PORT=6379

MAIL_MAILER=smtp
MAIL_HOST=mailpit
MAIL_PORT=1025
MAIL_USERNAME=null
MAIL_PASSWORD=null
MAIL_ENCRYPTION=null
MAIL_FROM_ADDRESS="hello@example.com"
```

Figura 36 - Configurar entorno (2)

- **Crear base de datos y ejecutar migraciones**
 - Crear la base de datos en MySQL con el nombre indicado en ``DB_DATABASE``.
 - **Abre Git Bash y ejecuta:**
php artisan migrate
php artisan db:seed

```
mysql -u root
CREATE DATABASE laravel;
EXIT;
```

Proyecto Final DAW – Vanemecum

```
MINGW64:/c:/xampp/htdocs/VanemecumLaravel
vanesa@LAPTOP-EL883HUQ MINGW64 /c:/xampp/htdocs/VanemecumLaravel
$ mysql -u root
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 26
Server version: 10.4.32-MariaDB mariadb.org binary distribution
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
MariaDB [(none)]> |
```

Figura 37 - Crear BD y ejecutar migraciones (1)

```
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
MariaDB [(none)]> CREATE DATABASE laravel;
Query OK, 1 row affected (0.003 sec)
MariaDB [(none)]> |
```

Figura 38 - Crear BD y ejecutar migraciones (2)

```
vanesa@LAPTOP-EL883HUQ MINGW64 /c:/xampp/htdocs/VanemecumLaravel
$ php artisan migrate

  INFO  Preparing database.

Creating migration table ..... 30ms DONE

  INFO  Running migrations.

2014_10_12_000000_create_users_table ..... 38ms DONE
2014_10_12_100000_create_password_resets_table ..... 45ms DONE
2019_08_19_000000_create_failed_jobs_table ..... 25ms DONE
2019_12_14_000001_create_personal_access_tokens_table ..... 37ms DONE

vanesa@LAPTOP-EL883HUQ MINGW64 /c:/xampp/htdocs/VanemecumLaravel
$ |
```

Figura 39 - Crear BD y ejecutar migraciones (3)

- **Levantar el servidor de desarrollo**
 - **Abre Git Bash y ejecuta:**
php artisan serve
Acceder en el navegador a <http://127.0.0.1:8000>

```
anessa@LAPTOP-EL883HUQ MINGW64 /c/xampp/htdocs/VanemecumLaravel
$ php artisan serve

[INFO] Server running on [http://127.0.0.1:8000].

Press Ctrl+C to stop the server
```

Figura 40 - Levantar servidor

- **Assets (si se usa Vite/Tailwind compilado)**

- **Abre Git Bash y ejecuta:**

- npm install

- npm run build


6.3. Manual de usuario

6.3.1. Usuario registrado (rol User)

Pantalla de Acceso y Login: Pantalla de Acceso a la aplicación, creación de usuarios nuevos y restablecimiento de contraseña.



Figura 41 - Pantalla de acceso y login (1)



Registrarse

Nombre

Correo electrónico

DNI

Teléfono

Dirección

Contraseña

Confirmar contraseña

[Crear cuenta](#)

[¿Ya tienes cuenta? Iniciar sesión](#)

Consulta de patógenos y tratamientos

Figura 42 - Pantalla de acceso y login (2)



Iniciar sesión

Correo electrónico

Contraseña

☐ Recordarme [¿Olvidaste tu contraseña?](#)

Entrar

[¿No tienes cuenta? Registrarse](#)

Consulta de patógenos y tratamientos

Figura 43 - Pantalla de acceso y login (3)



¿Olvidaste tu contraseña? Indica tu correo electrónico y te enviaremos un enlace para restablecerla.

Correo Electrónico

ENVIAR ENLACE PARA RESTABLECER CONTRASEÑA

Consulta de patógenos y tratamientos

Figura 44 - Pantalla de acceso y login (4)

Proyecto Final DAW – Vanemecum

Tras **iniciar sesión** tiene acceso a:

Dashboard: Pantalla principal tras el login.

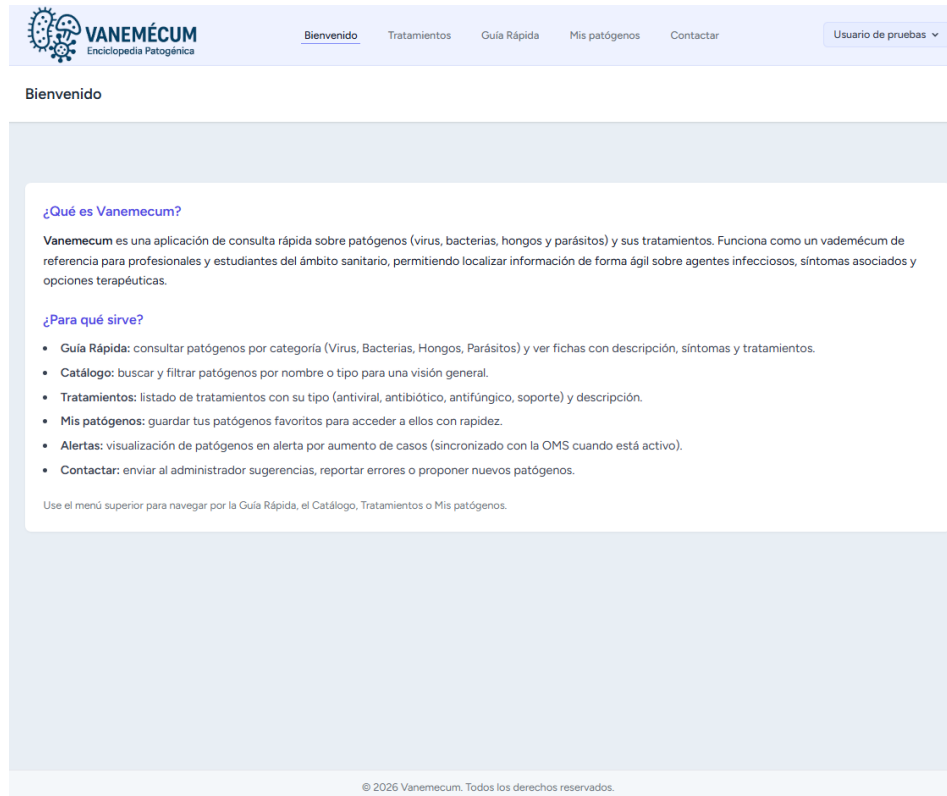


Figura 45 - Dashboard

Guía Rápida: Listado de patógenos con búsqueda por nombre y filtro por tipo (virus, bacteria, hongo). Al hacer clic en un patógeno se abre la **ficha detallada** (descripción, imagen, síntomas, tratamientos, fuente, alerta si existe). En la ficha puede pulsar

“Guardar en Mis patógenos para añadirlo a favoritos.

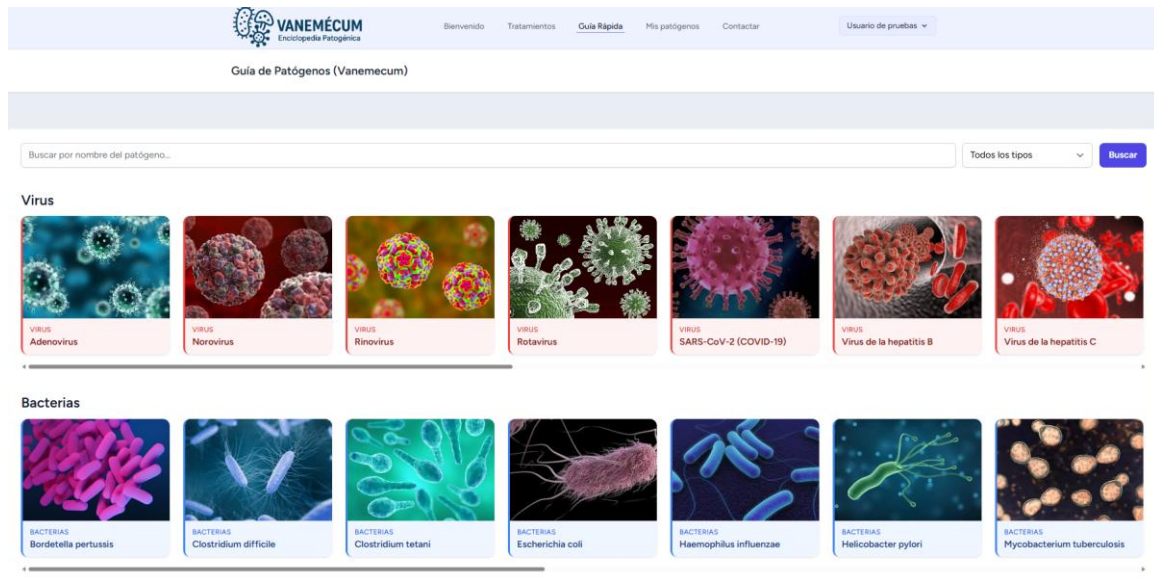



Figura 46 - Guía Rápida (1)

[← Volver al listado](#)[♥ Guardar en Mis patógenos](#)



Bacterias

Escherichia coli

(Bacteria Gram negativa. Infecciones urinarias, gastroenteritis, sepsis. Cepas productoras de toxina...)

Información General

Bacteria Gram negativa. Infecciones urinarias, gastroenteritis, sepsis. Cepas productoras de toxina Shiga (EHEC) causan diarrea sanguinolenta.

Síntomas Comunes

- Elevación de la temperatura corporal por encima de 38 °C.
- Deposiciones líquidas o frecuentes.
- Malestar digestivo y vómitos.
- Dolor en la zona del abdomen.

Tratamiento y Farmacología

Tratamientos Asociados

Hidratación y reposoAmoxicilinaAzitromicina

Fuente de Información

OMS
<https://www.who.int/>
Organización Mundial de la Salud.

© 2026 Vanemecum. Todos los derechos reservados.

Figura 47 - Guía Rápida (2)

Mis patógenos: Listado de patógenos guardados; puede eliminar cada uno.

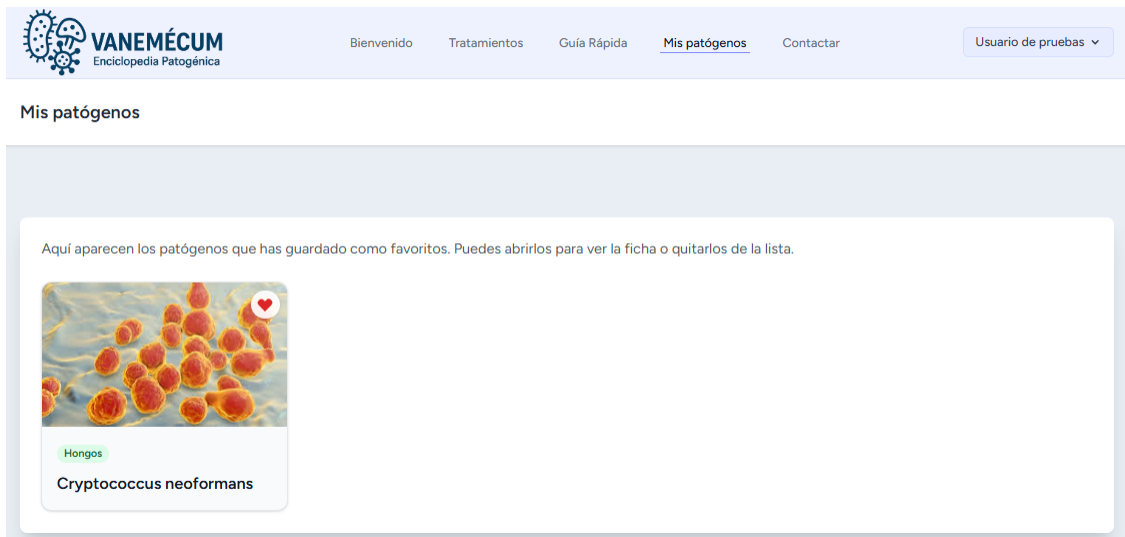


Figura 48 - Mis patógenos

Tratamientos: Listado y ficha de tratamientos (solo consulta).

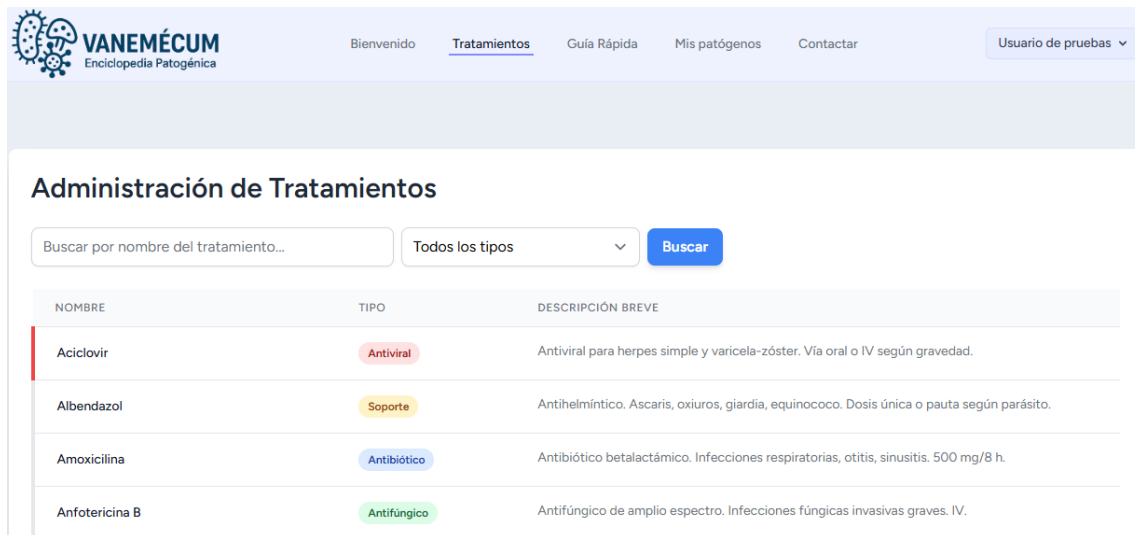
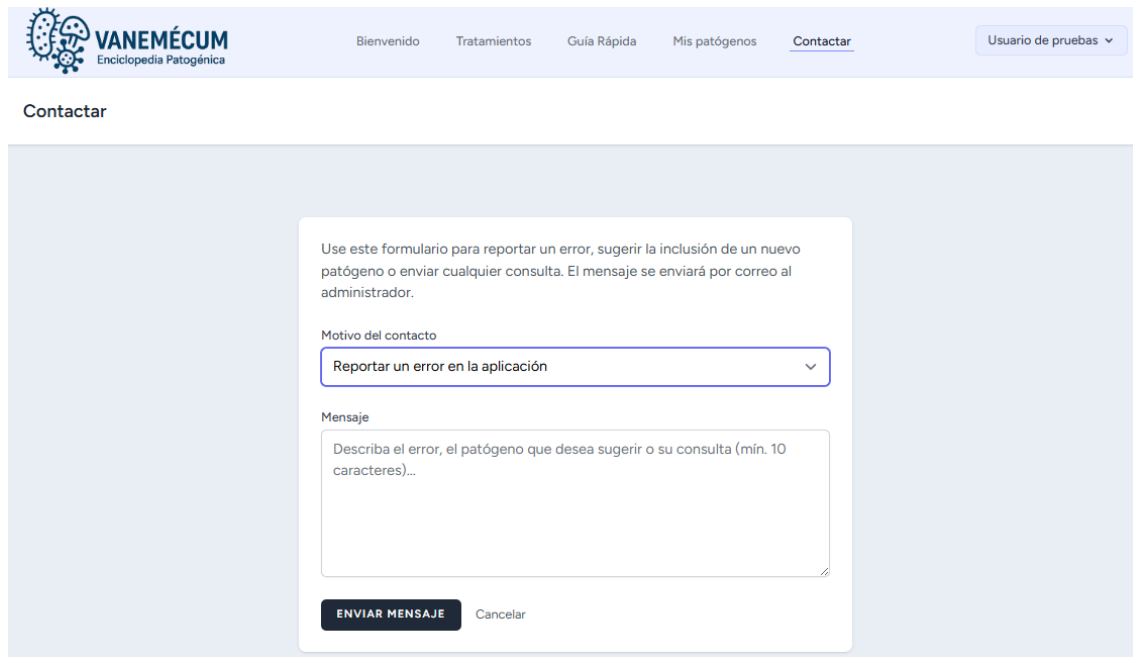


Figura 49 - Tratamientos

Contacto: Formulario para enviar un mensaje (reportar error, sugerir patógeno, otro); se pide nombre, email, tipo y mensaje.

Proyecto Final DAW – Vanemecum



Vanemecum
Enciclopedia Patogénica

Bienvenido Tratamientos Guía Rápida Mis patógenos Contactar Usuario de pruebas ▾

Contactar

Use este formulario para reportar un error, sugerir la inclusión de un nuevo patógeno o enviar cualquier consulta. El mensaje se enviará por correo al administrador.

Motivo del contacto

Reportar un error en la aplicación ▾

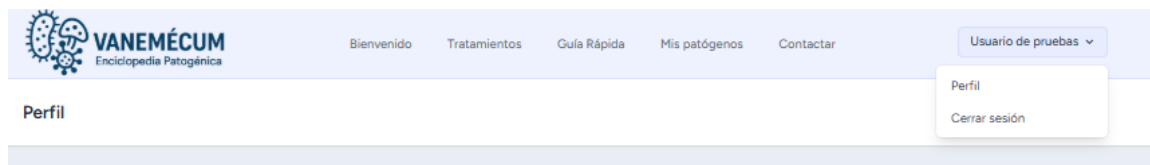
Mensaje

Describe el error, el patógeno que desea sugerir o su consulta (min. 10 caracteres)...

ENVIAR MENSAJE Cancelar

Figura 50 - Contacto

Perfil: editar datos de perfil y cerrar sesión (en el menú de navegación).



Vanemecum
Enciclopedia Patogénica


Bienvenido Tratamientos Guía Rápida Mis patógenos Contactar Usuario de pruebas ▾

Perfil

Perfil
Cerrar sesión

Figura 51 - Perfil (1)

Proyecto Final DAW – Vanemecum



BienvenidoTratamientosGuía RápidaMis patógenosContactar

Usuario de pruebas ▾

Perfil

Información del perfil

Actualice la información de su perfil y su dirección de correo electrónico.

Nombre

Usuario de pruebas

Correo electrónico

usuario@example.com

GUARDAR

Actualizar contraseña

Asegúrese de que su cuenta use una contraseña larga y aleatoria para mantenerse segura.

Contraseña actual

Nueva contraseña

Confirmar contraseña

GUARDAR

Eliminar cuenta

Una vez eliminada su cuenta, todos sus recursos y datos se eliminarán permanentemente.

Antes de eliminar la cuenta, descargue los datos o la información que desee conservar.

ELIMINAR CUENTA


© 2026 Vanemecum. Todos los derechos reservados.

Figura 52 - Perfil (2)

6.3.2. Administrador (rol Admin)

Además de todo lo anterior:

Micros-DB: listado de patógenos; botones para crear, editar y eliminar. Formulario con nombre, descripción, tipo, fuente, imagen, estado activo.

BienvenidoMicro-DB (Patógenos)AlertasTratamientosGuía RápidaMis patógenosMensajesadmin ▾

Administración de Patógenos

Lista de Patógenos

Crear Nuevo Patógeno

Buscar por nombre o descripción... Buscar





IMAGEN	NOMBRE	TIPO	DESCRIPCIÓN	ACTIVO	ACCIONES
	Adenovirus	Virus	Virus ADN. Infecciones respiratorias, conjuntivitis, ...	SI	Editar Eliminar
	Ancylostoma duodenale	Parásitos	Nematodo. Anquilostomiasis. Anemia por pérdida ...	SI	Editar Eliminar
	Ascaris lumbricoides	Parásitos	Nematodo. Ascariasis: infección intestinal muy pre...	SI	Editar Eliminar

Figura 53 - Micro-DB (Patógenos) (1)

Proyecto Final DAW – Vanemecum

**VANEMECUM**
Enciclopedia Patológica

Bienvenido

[Micro-DB \(Patógenos\)](#)

[Alertas](#)

[Tratamientos](#)

[Guía Rápida](#)

[Mis patógenos](#)

[Mensajes](#)

admin ▾

Editar Patógeno: [Ancylostoma duodenale](#)

[← Volver a la Administración](#)

Información Básica

Nombre Científico / Identificador

Ancylostoma duodenale

Tipo de Patógeno


Parásitos ▾

Fuente Principal de Referencia

OMS ▾

Detalles y Archivos

Imagen Actualizar (JPG/PNG)



Seleccionar archivo

Ningún archivo seleccionado

Dejar vacío para mantener la imagen actual.

Descripción Detallada

Nematodo, Anquilostomiasis, Anemia por pérdida de sangre intestinal.
Penetración cutánea de larvas.

☒ Activo (Visible en la Guía)

☐ Alerta epidemiológica por aumento de casos

Marque esta casilla si este patógeno está actualmente en alerta por aumento de incidencias.

Mensaje breve de alerta (opcional)

Ej.: Aumento significativo de casos en la última semana.

Relaciones: Síntomas y Tratamientos

Síntomas Asociados

amet nic
aut voluptas
Cefalea
Congestión nasal
Diarrea
dignissimos ea (generalizado)
Dolor abdominal
Dolor de garganta
Dolor muscular

Mantén pulsado Ctrl/Cmd para seleccionar múltiples.

Tratamientos Aplicables

Aciclovir
Albendazol
Amoxicilina
Anfotericina B
Antimoniales pentavalentes
Arteméter con lumefantrina
Aзитromicina
Benzidazol
Ceftriaxona

Mantén pulsado Ctrl/Cmd para seleccionar múltiples.

ACTUALIZAR PATÓGENO

Figura 54 - Micro-DB (Patógenos) (2)

Tratamientos: listado con columna “Acciones” (solo Admin); crear, editar y eliminar tratamientos (nombre, descripción, duración, tipo).

Página 39 de 43 Vanemecum

Proyecto Final DAW – Vanemecum

Administración de Tratamientos [+ Crear Nuevo Tratamiento](#)

Buscar por nombre del tratamiento... Todos los tipos Buscar

NOMBRE	TIPO	DESCRIPCIÓN BREVE	ACCIONES
Aciclovir	Antiviral	Antiviral para herpes simple y varicela-zóster. Vía oral o IV según gravedad.	Editar Eliminar
Albendazol	Soporte	Antihelmíntico. Ascaris, oxiuros, giardia, equinococo. Dosis única o pauta según parásito.	Editar Eliminar
Amoxicilina	Antibiótico	Antibiótico betalactámico. Infecciones respiratorias, otitis, sinusitis. 500 mg/8 h.	Editar Eliminar
Anfotericina B	Antifúngico	Antifúngico de amplio espectro. Infecciones fúngicas invasivas graves. IV.	Editar Eliminar
Antimoniales pentavalentes	Soporte	Antimoniato de meglumina. Leishmaniasis visceral y cutánea. Esquema según forma clínica.	Editar Eliminar

Figura 55 - Tratamientos (1)

Editar Tratamiento [← Volver al Listado](#)

Nombre del Tratamiento:

Tipo de Tratamiento: Antiviral

Nota: Si el Tipo no aparece aquí, debe crearlo en su sección de administración auxiliar.


Descripción:

[Actualizar Tratamiento](#)

Figura 56 - Tratamientos (2)

Mensajes: listado de mensajes de contacto con filtro “solo no leídos”; al hacer clic en “Ver” se abre el detalle; desde listado o desde detalle se puede **marcar como leíd o no leído**.

Proyecto Final DAW – Vanemecum

**VANEMÉCUM**
Enciclopedia Patogénica


BienvenidoMicro-DB (Patógenos)AlertasTratamientosGuía RápidaMis patógenos**Mensajes 1**admin ▾

Mensajes de contacto

TodosNo leídos

FECHA	USUARIO	MOTIVO	MENSAJE	ACCIONES
13/02/2026 21:10	Usuario de pruebas usuario@example.com	Otra consulta o sugerencia	Te has esforzado mucho en tu trabajo y estoy impresi...	Ver Marcar no leído
10/02/2026 23:02	Usuario de pruebas usuario@example.com	Otra consulta o sugerencia	fSDFGFERGFGSRTGSRTGTRG	Ver Marcar leído

Figura 57 - Mensajes (1)

**VANEMÉCUM**
Enciclopedia Patogénica

BienvenidoMicro-DB (Patógenos)AlertasTratamientosGuía RápidaMis patógenos**Mensajes**admin ▾

Mensaje de contacto

FECHA
10/02/2026 23:02

USUARIO
Usuario de pruebas
usuario@example.com

MOTIVO
Otra consulta o sugerencia

MENSAJE
fSDFGFERGFGSRTGSRTGTRG

[← Volver al listado](#) Marcar como no leído

Figura 58 - Mensajes (2)

Alertas: listado de patógenos con alerta activa; búsqueda por nombre.

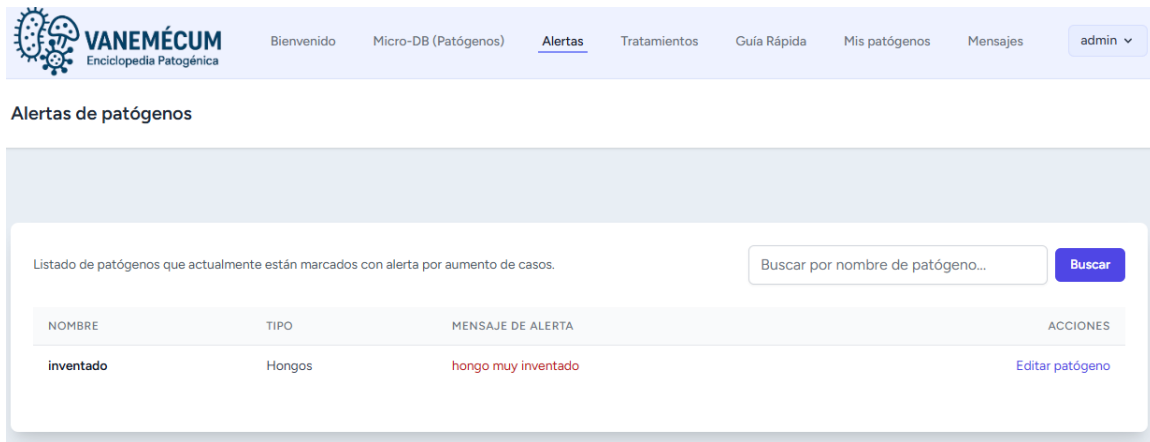


Figura 59 - Alertas

7. CONCLUSIONES FINALES

7.1. Evaluación del proyecto: objetivos conseguidos y no alcanzados

7.1.1 Objetivos conseguidos

Los objetivos principales del proyecto han sido alcanzados satisfactoriamente. Se ha desarrollado una aplicación web completa que funciona como guía de virus, bacterias y hongos, permitiendo:

- Consulta pública de información.
- Acceso a funcionalidades adicionales para usuarios registrados (guía personalizada, favoritos y formulario de contacto).
- Gestión integral mediante un panel de administración.

El sistema incorpora:

- Autenticación y control de roles (Administrador y Usuario).
- Sistema de búsqueda y filtros avanzados tanto en la guía como en el panel de alertas.
- Integridad referencial y normalización en la base de datos, garantizando coherencia y consistencia de la información.

7.1.2 Objetivos no alcanzados o pospuestos

Algunos objetivos inicialmente planteados han sido pospuestos para futuras versiones, principalmente por limitaciones de tiempo o alcance. Entre ellos se encuentran:

- Implementación de exportación de datos en formatos CSV y PDF.
- Integración automática con APIs externas de alertas sanitarias.
- Desarrollo de un módulo de valoraciones y comentarios por patógeno.

Estos elementos no afectan al funcionamiento principal del sistema, pero representan oportunidades claras de mejora y ampliación futura.

7.2. Propuestas de mejora

- Implementación de funcionalidad de **exportación de listados** (patógenos y tratamientos) en formatos CSV y PDF.
- Incorporación de un **informe de incidencias** o historial de cambios accesible desde el panel de administración.
- Mejora de la **accesibilidad** (uso de atributos ARIA, mejora de contraste) y ampliación de las **pruebas automatizadas** (PHPUnit y tests de integración).
- Desarrollo de una posible **API REST** para permitir la integración con aplicaciones externas o móviles.