

## Documentación del Proyecto Web: Zoonika Web

### 1. Información General del Proyecto

- **Nombre del Proyecto:** Zoonika Web
- **Descripción General:**

Zoonika Web es una aplicación web moderna diseñada para clientes de la clínica veterinaria Zoonika. A través de esta plataforma, los usuarios pueden visualizar información sobre los servicios ofrecidos, conocer al equipo de especialistas y dejar comentarios respecto a la atención recibida. El sistema incluye funcionalidades de registro, inicio de sesión y publicación de comentarios personalizados.
- **Propósito y Objetivos:**

El objetivo principal es fortalecer la relación entre los clientes y la clínica Zoonika mediante una plataforma web accesible, transparente y funcional. Se busca facilitar la interacción cliente-clínica, permitir retroalimentación directa y mejorar la presencia digital de la empresa.
- **Alcance:**

El proyecto abarca el desarrollo de las siguientes funcionalidades:

  - Registro e inicio de sesión de clientes.
  - Visualización de servicios ofrecidos por Zoonika.
  - Visualización del equipo de especialistas.
  - Publicación de comentarios por parte de clientes sobre los especialistas.
  - Visualización de una galería con fotos relacionadas a la clínica.

### 2. Requerimientos Funcionales

ID	Módulo / Característica	Descripción
F1	Registro de Usuarios	Permitir a los clientes crear una cuenta con datos personales y contraseña.
F2	Inicio de Sesión	Permitir a los clientes iniciar sesión para acceder a funciones exclusivas.
F3	Visualización de Servicios	Mostrar los servicios veterinarios que ofrece Zoonika.
F4	Visualización de Especialistas	Mostrar un listado con la información de los especialistas.
F5	Publicación de Comentarios	Permitir que los clientes registrados dejen comentarios a los especialistas.
F6	Visualización de Galería	Mostrar fotos o imágenes de la clínica y mascotas en la galería.

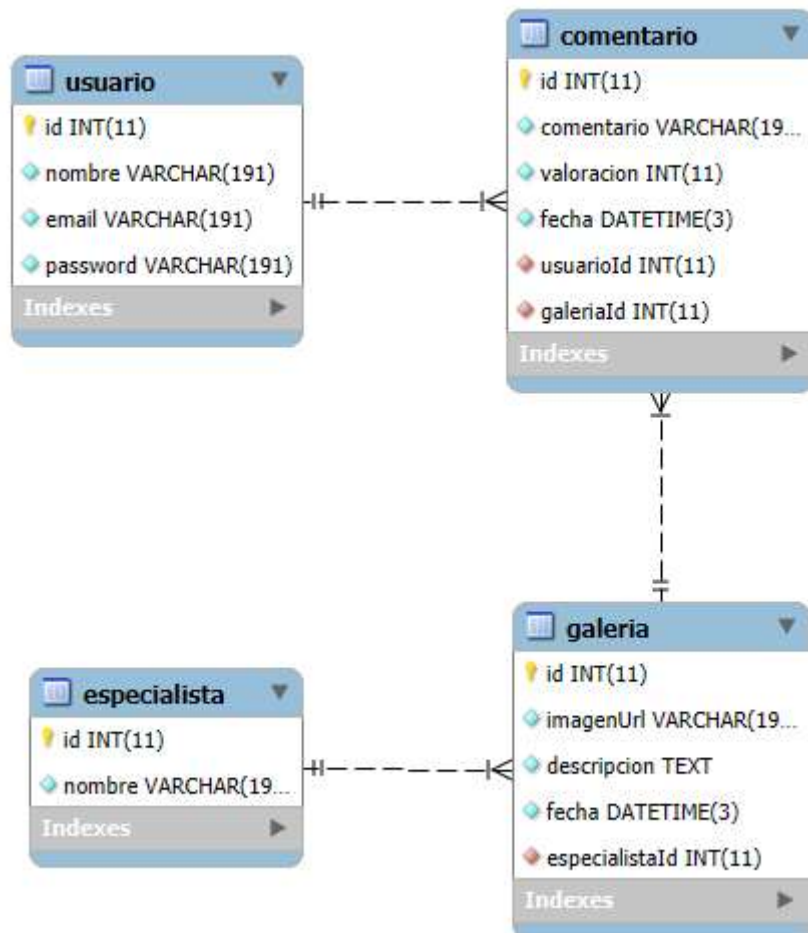
### 3. Diseño de Base de Datos

#### Modelo relacional (Prisma + MySQL):

La base de datos está compuesta por 4 tablas principales:

- Usuario: Contiene la información de los clientes (nombre, email, contraseña, etc.).
- Especialista: Datos del personal veterinario (nombre, especialidad, experiencia, etc.).
- Comentario: Almacena los comentarios hechos por los usuarios sobre un especialista determinado.
- galería: Enlaces a imágenes relacionadas con la clínica y su atención.

#### Diagrama ERD:



## 4. Diseño de Interfaz de Usuario (UI)

El diseño de Zoonika Web fue realizado en Figma y contempla las siguientes vistas clave:



## Vista de Inicio

- **Descripción:** Página principal donde se presenta la clínica, su lema y un resumen de sus servicios.
- **Flujo:** El usuario puede desplazarse hacia abajo para ver los servicios o navegar usando la barra superior.

## Especialistas

- **Descripción:** Muestra tarjetas con la información del staff veterinario.
- **Flujo:** Desde aquí se puede seleccionar a un especialista y ver comentarios publicados.

## Comentario

- **Descripción:** Los usuarios logueados pueden enviar su opinión.
- **Flujo:** Se accede al iniciar sesión y seleccionar un especialista.

## Login / Registro

- **Descripción:** Formulario para autenticarse o crear una cuenta nueva.
- **Flujo:** Redirecciona a la vista principal al completar el proceso.

## Galería

- **Descripción:** Muestra imágenes de mascotas, servicios y equipo de Zoonika.

## 5. Arquitectura de la Aplicación

### Componentes Principales:

- **Frontend (Cliente):**  
Desarrollado con **Next.js**, maneja las vistas, la navegación entre páginas y la interacción del usuario. Usa componentes reutilizables y estilos modernos para brindar una experiencia amigable.
- **Backend (Servidor):**  
Implementado con **Prisma ORM** conectado a **MySQL**. Se encarga de la lógica de negocio, validaciones, autenticación y manejo de comentarios. Las operaciones CRUD están disponibles mediante rutas protegidas.
- **Base de Datos:**  
MySQL estructurada mediante Prisma, donde se almacenan los datos de usuarios, especialistas, comentarios y fotos.

### Interacción entre Componentes:

- El usuario accede al frontend y realiza acciones (ver especialistas, iniciar sesión, dejar comentario).
- El frontend realiza solicitudes al backend mediante rutas API (REST).

- El backend procesa la lógica correspondiente y consulta o actualiza la base de datos.
- El resultado se devuelve al cliente para su visualización inmediata.

## 6. Tecnologías Clave Utilizadas

Capa	Tecnología
Frontend	Next.js
Estilos	CSS
Backend	Prisma ORM
Base de Datos	MySQL
Autenticación	
Despliegue	