DOCUMENTACIÓN TÉCNICA Y FUNCIONAL

Sistema NovelasApp

Versión 1.0

Documento generado el: 01/04/2025 13:37:39

ÍNDICE DE CONTENIDOS

1. INTRODUCCIÓN

1.1 Descripción General

1.2 Propósito y Alcance

1.3 Objetivos del Sistema

2. DOCUMENTACIÓN FUNCIONAL

2.1 Módulo de Usuarios

2.2 Módulo de Novelas

2.3 Módulo de Capítulos

2.4 Módulo de Géneros

2.5 Módulo de Interacción

2.6 Módulo de Imágenes

2.7 Perfiles de Usuario

2.8 Flujos de Trabajo

3. DOCUMENTACIÓN TÉCNICA

3.1 Arquitectura del Sistema

3.2 Stack Tecnológico

3.3 Modelos y Base de Datos

3.4 Estructura del Código

3.5 API y Puntos de Entrada

3.6 Seguridad y Autenticación

3.7 Implementación de Características

4. ANEXOS

4.1 Glosario de Términos

4.2 Diagramas

1. INTRODUCCIÓN

1.1 Descripción General

NovelasApp es una plataforma web especializada para la creación, publicación y lectura de novelas en línea. Desarrollada con Laravel 10, el sistema permite a los escritores compartir sus obras por capítulos y a los lectores descubrir y seguir historias según sus géneros preferidos.

1.2 Propósito y Alcance

El propósito principal de NovelasApp es proporcionar un espacio donde:

• Los escritores puedan publicar sus novelas capítulo a capítulo

• Los lectores puedan descubrir y disfrutar de nuevas historias

• Se fomente la comunidad literaria mediante interacciones básicas

• Se mantenga un catálogo organizado por géneros literarios

1.3 Objetivos del Sistema

• Facilitar la creación y publicación de contenido literario

• Ofrecer una experiencia de lectura limpia y agradable

• Proporcionar herramientas para organizar y gestionar novelas

• Permitir la interacción básica entre usuarios y contenidos

• Establecer un sistema de categorización eficiente

2. DOCUMENTACIÓN FUNCIONAL

2.1 Módulo de Usuarios

2.1.1 Registro e Inicio de Sesión

El sistema utiliza Laravel Breeze para proporcionar un sistema de autenticación completo y seguro. Las funcionalidades principales incluyen:

• Registro de usuario con nombre, correo electrónico y contraseña

• Inicio de sesión con credenciales verificadas

• Recuperación de contraseña mediante correo electrónico

• Opción para mantener la sesión activa entre visitas ("Recordarme")

2.1.2 Gestión de Perfil

El usuario registrado puede gestionar su información personal a través de las siguientes opciones:

• Ver perfil con datos personales y estadísticas

• Editar información personal

• Cambiar contraseña

• Eliminar cuenta (dar de baja del sistema)

2.2 Módulo de Novelas

2.2.1 Gestión de Novelas

Los usuarios registrados pueden crear y administrar sus propias novelas a través de un conjunto completo de herramientas CRUD (Crear, Leer, Actualizar, Eliminar):

• Creación de novela con título, descripción, portada y géneros

• Listado de novelas personales

• Edición de datos de la novela

• Eliminación de novelas completas

• Visualización de estadísticas básicas (lecturas, interacciones)

2.2.2 Exploración de Novelas

El sistema proporciona diferentes formas de descubrir el contenido disponible:

• Catálogo general de novelas

• Filtrado por género literario

• Búsqueda por texto

• Sección de novelas destacadas

2.3 Módulo de Capítulos

2.3.1 Gestión de Capítulos

Cada novela puede contener múltiples capítulos que son gestionados por su autor:

• Creación de capítulos con editor de texto enriquecido

• Listado de capítulos por novela

• Edición de capítulos existentes

• Eliminación de capítulos

• Sistema de ordenación automática por número de capítulo

3. DOCUMENTACIÓN TÉCNICA

3.1 Arquitectura del Sistema

NovelasApp sigue una arquitectura Modelo-Vista-Controlador (MVC) implementada a través del framework Laravel 10. Esta arquitectura separa la lógica de negocio de la interfaz de usuario, facilitando el desarrollo y mantenimiento.

3.1.1 Componentes Principales

• Capa de Presentación: Blade templates y TailwindCSS

• Capa de Negocio: Controladores organizados por funcionalidad

• Capa de Datos: Modelos Eloquent con relaciones definidas

• Capa de Persistencia: Base de datos MySQL

3.2 Stack Tecnológico

El sistema está construido utilizando las siguientes tecnologías:

• Backend: PHP 8.1, Laravel 10

• Frontend: HTML5, CSS3, JavaScript, TailwindCSS, Alpine.js

• Base de Datos: MySQL

• Autenticación: Laravel Breeze

• Herramientas Adicionales: TinyMCE (editor WYSIWYG)

3.3 Modelos y Base de Datos

3.3.1 Entidades Principales

• User: Almacena información de usuarios y credenciales

• Novela: Contiene información sobre las novelas publicadas

• Capitulo: Guarda el contenido de cada capítulo de las novelas

• Genero: Categorías para clasificar las novelas

• CapituloImagen: Imágenes asociadas a cada capítulo

3.3.2 Relaciones Principales

• Usuario → Novelas: Un usuario puede tener muchas novelas (1:N)

• Novela → Capítulos: Una novela contiene muchos capítulos (1:N)

• Novela ↔ Géneros: Relación muchos a muchos (N:M)

• Capítulo → Imágenes: Un capítulo puede tener múltiples imágenes (1:N)

3.4 Estructura del Código

novelas-app/  
├── app/  
│ ├── Http/  
│ │ ├── Controllers/ # Controladores de la aplicación  
│ │ ├── Middleware/ # Middleware de autenticación y permisos  
│ │ └── Requests/ # Validación de formularios  
│ ├── Models/ # Modelos Eloquent  
│ └── Providers/ # Proveedores de servicios  
├── config/ # Configuración de la aplicación  
├── database/  
│ ├── migrations/ # Migraciones de la base de datos  
│ └── seeders/ # Datos iniciales  
├── public/ # Archivos accesibles públicamente  
│ └── storage/ # Imágenes y archivos subidos  
├── resources/  
│ ├── css/ # Estilos CSS/TailwindCSS  
│ ├── js/ # Código JavaScript  
│ └── views/ # Vistas Blade  
├── routes/ # Definición de rutas  
│ └── web.php # Rutas principales  
└── storage/ # Almacenamiento de la aplicación

3.5 API y Puntos de Entrada

3.5.1 Rutas Principales

El sistema organiza las rutas en grupos según su accesibilidad y funcionalidad:

• Rutas Públicas: Accesibles sin autenticación

• Rutas Protegidas: Requieren autenticación

• Rutas Administrativas: Restringidas a administradores

Es importante destacar que las rutas de capítulos están anidadas dentro de las rutas de novelas, reflejando la relación jerárquica entre estos recursos. Por ejemplo:

/novelas/{novela}/capitulos

3.5.2 Middlewares

Los principales middleware utilizados incluyen:

• auth: Verifica la autenticación del usuario

• verified: Confirma que el email del usuario ha sido verificado

• admin: Restringe el acceso a usuarios con rol de administrador

3.6 Seguridad y Autenticación

El sistema implementa varias capas de seguridad para proteger los datos y el acceso:

• Autenticación mediante Laravel Breeze

• Protección CSRF en formularios

• Validación de datos de entrada

• Almacenamiento seguro de contraseñas con bcrypt

• Sesiones con tiempo de expiración configurable

4. ANEXOS

4.1 Glosario de Términos

|  |  |
| --- | --- |
| Término | Definición |
| Laravel | Framework de PHP utilizado para el desarrollo de la aplicación. |
| MVC | Patrón de diseño Modelo-Vista-Controlador implementado en la arquitectura. |
| Middleware | Mecanismo de filtrado de peticiones HTTP en Laravel. |
| Eloquent | ORM (Object-Relational Mapping) utilizado para interactuar con la base de datos. |
| Blade | Motor de plantillas de Laravel utilizado para las vistas. |
| TailwindCSS | Framework CSS utilizado para el diseño de la interfaz de usuario. |
| MySQL | Sistema de gestión de base de datos relacional utilizado para almacenar los datos. |
| Breeze | Paquete de Laravel que proporciona implementación de autenticación. |