

SIREH - Sistema de Registro Hotelero

SIREH es una plataforma web desarrollada para gestionar hoteles, habitaciones, tipos de acomodaciones y usuarios. Permite validar la capacidad hotelera, administrar las reglas de acomodación, y visualizar todo desde una interfaz moderna y responsiva.

Tecnologías Utilizadas

Backend:

- ✓ Laravel 12 - Framework PHP para construir APIs RESTful.
- ✓ PostgreSQL - Sistema de base de datos relacional.
- ✓ PHPUnit - Pruebas automatizadas.
- ✓ Migrations y Seeders - Para creación y población de la base de datos para usuarios.

Frontend:

- ✓ React 18 - Biblioteca para interfaces de usuario
- ✓ Vite - Empaquetador rápido para proyectos Frontend
- ✓ Bootstrap 5 - Framework CSS para diseño responsive
- ✓ React Router - Enrutamiento de vistas
- ✓ Axios - Cliente HTTP para consumo de API
- ✓ React Toastify - Notificaciones elegantes y usables

Funcionalidades Principales

- Registro y gestión de hoteles (nombre, dirección, ciudad, gerente, etc.)
- Registro de habitaciones por hotel, validando que no excedan la capacidad permitida.
- Asignación de acomodaciones y tipos de habitación según reglas definidas.
- Gestión de usuarios y roles (administrador, gerente, etc.).
- Listado de combinaciones permitidas tipo de habitación + acomodación.
- Diseño adaptable para portátiles de 13" y 15", compatible con pantallas móviles.

Requisitos Previos

- PHP 8.2+
- Composer
- PostgreSQL
- Node.js 18+
- NPM- Vite

Instalación y Ejecución

1. Clonación del repositorio

Se utilizará la para Git Clone, con el fin de poder copiar el proyecto completo desde un repositorio remoto (por ejemplo, GitHub) a tu máquina local. Esto te permite tener todos los archivos del proyecto y comenzar a trabajar en ellos.

=> git clone https://github.com/tu-usuario/cadena_hotelera.git

Luego ingresamos a la carpeta recién clonada con la ayuda de cd

=> cd cadena_hotelera

2. Backend (Laravel)

Realizaremos la configuración del Backend para ello seleccionamos la carpeta Backend.

=> cd backend

Realizamos la instalación de todas las dependencias necesarias que Laravel necesita para funcionar. “composer” es el gestor de paquetes para PHP.

=> composer install

Utilizamos el siguiente comando el cual copia el archivo .env.example como .env, y este es utilizado para configurar variables de entorno (como credenciales de la base de datos).

=> cp .env.example .env

Genera una nueva clave de aplicación única en el archivo .env. Esta clave es utilizada por Laravel para cifrar información sensible.

= > php artisan key:generate

Ejecuta las migraciones, que crean las tablas en la base de datos. El --seed además llena esas tablas con datos iniciales definidos en los seeders.

= > php artisan migrate --seed

Se inicia el servidor de desarrollo de Laravel, lo que te permite acceder a la API en tu navegador o desde el frontend en <http://127.0.0.1:8000>.

= > php artisan serve

3. Frontend (React + Vite)

Cambia el directorio actual a la carpeta del frontend, donde está el proyecto React.

= > cd ../frontend

Se instala todas las dependencias del proyecto React. npm es el gestor de paquetes para JavaScript.

= > npm install

Se inicia el servidor de desarrollo de Vite, permitiéndote ver el frontend en tiempo real en <http://localhost:5173>. Este servidor se actualiza automáticamente al guardar los cambios.

= > npm run dev

Aplicamos los siguientes comandos para el Backend (Laravel optimización):

- `php artisan config:cache`
- `php artisan route:cache`
- `php artisan optimize`

Tener presente que siempre estos serán los puertos por Defecto

- Laravel: <http://127.0.0.1:8000>

- Vite (React): <http://localhost:5173>