**Documentación Técnica del Proyecto**

**1. Introducción**

Este documento describe la implementación de un sistema de gestión de hoteles realizado con Laravel 11 y Vue 3, utilizando PostgreSQL como base de datos.

**2. Requisitos del Sistema**

**2.1 Requisitos Funcionales**

* Ingreso de datos básicos del hotel.
* Validación de datos tributarios básicos.
* Asignación de tipos de habitación (Estándar, Junior, Suite).
* Validación de acomodaciones según el tipo de habitación.

**2.2 Requisitos No Funcionales**

* La aplicación debe ser completamente RESTful.
* Separación del frontend y backend.
* Despliegue en la nube.
* Soporte para navegadores Firefox y Chrome.

**3. Arquitectura del Sistema**

La arquitectura del sistema está basada en una estructura de cliente-servidor. El frontend se implementa con Vue 3, mientras que el backend se desarrolla con Laravel 11 y se conecta a una base de datos PostgreSQL.

**4. Modelo Entidad-Relación (MER)**

El modelo entidad-relación describe las entidades del sistema y sus relaciones. A continuación, se presenta el diccionario de datos.

**Diccionario de Datos**

| **Table** | **Column** | **Type** | **Description** |
| --- | --- | --- | --- |
| hotels | id | SERIAL (PK) | Unique identifier for the hotel |
| hotels | name | VARCHAR | Name of the hotel (unique) |
| hotels | address | VARCHAR | Address of the hotel |
| hotels | city | VARCHAR | City where the hotel is located |
| hotels | tax\_id | VARCHAR | Tax identification number of the hotel (unique) |
| hotels | number\_of\_rooms | INTEGER | Maximum number of rooms in the hotel |
| hotels | timestamps | TIMESTAMPS, SOFT DELETES | Timestamps and soft deletes |
| room\_types | id | SERIAL (PK) | Unique identifier for the room type |
| room\_types | description | VARCHAR | Description of the room type (Standard, Junior, Suite) |
| accommodations | id | SERIAL (PK) | Unique identifier for the accommodation |
| accommodations | description | VARCHAR | Description of the accommodation (Single, Double, Triple, Quadruple) |
| rooms | id | SERIAL (PK) | Unique identifier for the room |
| rooms | hotel\_id | INTEGER (FK) | Unique identifier for the hotel |
| rooms | room\_type\_id | INTEGER (FK) | Unique identifier for the room type |
| rooms | accommodation\_id | INTEGER (FK) | Unique identifier for the accommodation |
| rooms | quantity | INTEGER | Number of rooms of this type and accommodation |
| accommodation\_restrictions | room\_type\_id | INTEGER (PK, FK) | Unique identifier for the room type |
| accommodation\_restrictions | accommodation\_id | INTEGER (PK, FK) | Unique identifier for the accommodation |

**5. Manual de Usuario**

**5.1 Guía de Instalación**

Para instalar la aplicación, sigue los siguientes pasos “Back-end”:

1. Clona el repositorio del proyecto:

git clone <https://github.com/gersonbr15/decameron.git>

1. Ingresa a la carpeta “itbf\_laravel”
2. Instala las dependencias de PHP con Composer:

composer install

1. Configura el archivo .env con las credenciales de tu base de datos PostgreSQL

DB\_CONNECTION=pgsql

DB\_HOST=localhost

DB\_PORT=5432

DB\_DATABASE=ITBF

DB\_USERNAME=postgres

DB\_PASSWORD=123456

1. Ejecuta las migraciones para crear las tablas en la base de datos.

php artisan migrate

1. Inicia el servidor de desarrollo con el siguiente comando:

php artisan serve

Para instalar la aplicación, sigue los siguientes pasos “Front-end”:

1. Ingresa a la carpeta “itbf\_vue”
2. Instala las dependencias de NPM:

npm install

1. Inicia el servidor de desarrollo con el siguiente comando:

npm run dev

**5.2 Guía de Uso**

Para utilizar la aplicación, sigue los siguientes pasos:

1. Accede a la aplicación desde tu navegador.

ruta front-end: <http://localhost:5173/>

ruta back-end: <http://127.0.0.1:8000/>

1. Registra un nuevo hotel ingresando los datos requeridos.
2. Asigna tipos de habitación y acomodaciones al hotel.

6. **Manual de Desarrollador**

**6.1 Estructura del Proyecto**

La estructura del proyecto está organizada de la siguiente manera:

app/ - Contiene el código fuente de la aplicación

├── Console/

├── Exceptions/

├── Http/

│ ├── Controllers/

│ │ └── HotelController.php

│ ├── Middleware/

│ └── Kernel.php

├── Models/

│ ├── Hotel.php

│ └── Room.php

├── Providers/

├── Services/

│ └── HotelService.php

bootstrap/

config/

database/

├── factories/

├── migrations/

│ ├── 2024\_01\_01\_000001\_create\_hotels\_table.php

│ ├── 2024\_01\_01\_000002\_create\_room\_types\_table.php

│ ├── 2024\_01\_01\_000003\_create\_accommodations\_table.php

│ ├── 2024\_01\_01\_000004\_create\_rooms\_table.php

│ ├── 2024\_01\_01\_000005\_create\_accommodation\_restrictions.php

├── seeders/

public/

resources/

routes/

└── api.php

storage/

tests/

.env

artisan

composer.json

composer.lock

package.json