**Definicion y especificaciones del requerimiento**

Rivers Emmanuel Morales.

La siguiente Ali se desarrolla con el fin de realizar una reservación en la cual integre

Una lista de requisitos (modelos) para efectuar el pago.

**Challenge**

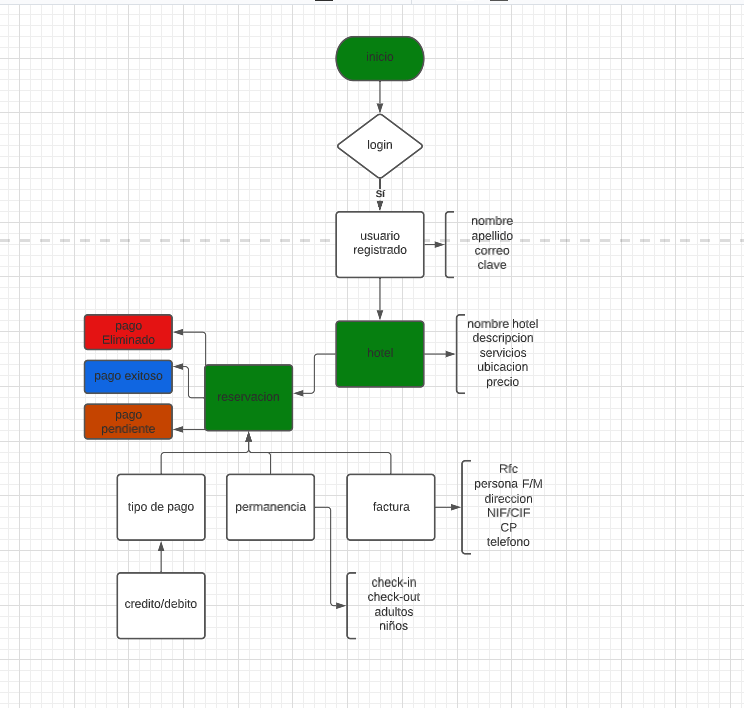
* Utilizando Node.js y Express Framework, desarrollá los endpoints para el sistema de reservas de habitación de un hotel.
* CONDICIONES:
* ✅ Las reservas pueden tener 3 estados: Pendiente, Pagado y Eliminado.
* ✅Los datos a almacenar para la reserva son:
* los detalles del cuarto reservado,
* los días de estadía,
* los datos de facturación e identificación del cliente,
* el monto pagado
* el método de pago.
* ✅ Proponé los endpoints a crearse para tratar de cubrir el flujo normal de operación de reserva y explicar por qué.
* Luego que tengas ya todo el código
* ✅Crear un repositorio para la entrega del código y en un readme del repositorio la justificación de los endpoints creados.
* - El proyecto debe correrse en un contenedor de docker, se puede usar cualquier gestor de base de datos relacionales.
* - Adjuntar también, en el repositorio, un archivo con ejemplos de consumo de la API para Postman
* tenemos 48 hrs para poder resolver el challenge

Se busca que la aplicación pueda ejecutarse tal y como se especifica en la parte superior del texto

Dentro del siguiente proyecto se cubrirán/usaran los siguientes frameworks o librerías

* Relaciones con tablas
* JWT — validación de sesiónelos, protección de claves de acceso
* Bcryptjs - para hashing de claves
* Middleware’s - intercambio de código/informacion entre enpoints o rutas
* Express - entorno de trabajo para programación en nodejs
* Nodejs - entorno de desarrollo
* Sequelize - manipulación de la base de datos postres.
* Base de datos relacional (postres SQL) — almacenamiento de información, consulta, entre otros.
* Heroku - almacenamiento de información en la nube.
* Nodemon - esta librería nos permitirá reiniciar automáticamente el proceso para ver los cambios de forma agilizada.
* Multar - manejo de archivos /form-data
* Dotenv — carga de archivo o información .env sensible
* Helmet - protección de Http
* Morgan - registro de errores (middleware’s)
* Postman - pruebas de endpoints o rutas

Se crea el siguiente diagrama para determinar las relaciones, el proceso o los requisitos que necesita la API para realizar una reservación

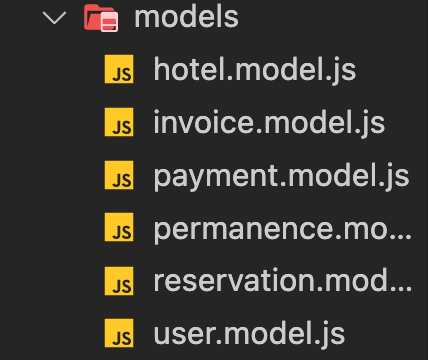


A continuación se consideran las siguientes rutas.

|  |  |  |  |
| --- | --- | --- | --- |
| **modelos** |  | **peticion** | **ruta** |
| *hotel* | *obtener* | *GET* | https://hotel-reservation12.herokuapp.com/api/v1/hotel/ |
|  | *crear* | *Post* | https://hotel-reservation12.herokuapp.com/api/v1/hotel/create/ |
| *Invoice* | *obtener* | *Get* | https://hotel-reservation12.herokuapp.com/api/v1/invoice/ |
|  | *crear* | *Post* | https://hotel-reservation12.herokuapp.com/api/v1/invoice/create/ |
| *Payment* | *obtener* | *Get* | https://hotel-reservation12.herokuapp.com/api/v1/payment/ |
|  | *crear* | *Post* | https://hotel-reservation12.herokuapp.com/api/v1/payment/create/ |
| *Permanence* | *obtenr* | *Get* | https://hotel-reservation12.herokuapp.com/api/v1/permanence/ |
|  | *crear* | *Post* | https://hotel-reservation12.herokuapp.com/api/v1/permanence/create/ |
| *Reservation* | *pago exit* | *Post* | https://hotel-reservation12.herokuapp.com/api/v1/reservation/create/ |
|  | *pendiente* | *Post* | https://hotel-reservation12.herokuapp.com/api/v1/reservation/pending/ |
|  | *obtener* | *Get* | https://hotel-reservation12.herokuapp.com/api/v1/reservation/ |
|  | *eliminar* | *Delete* | https://hotel-reservation12.herokuapp.com/api/v1/reservation/id |
| *User* | *obtener* | *Get* | https://hotel-reservation12.herokuapp.com/api/v1/user/all/ |
|  | *crear usuario* | *Post* | https://hotel-reservation12.herokuapp.com/api/v1/user/ |
|  | *login* | *Post* | https://hotel-reservation12.herokuapp.com/api/v1/user/login |

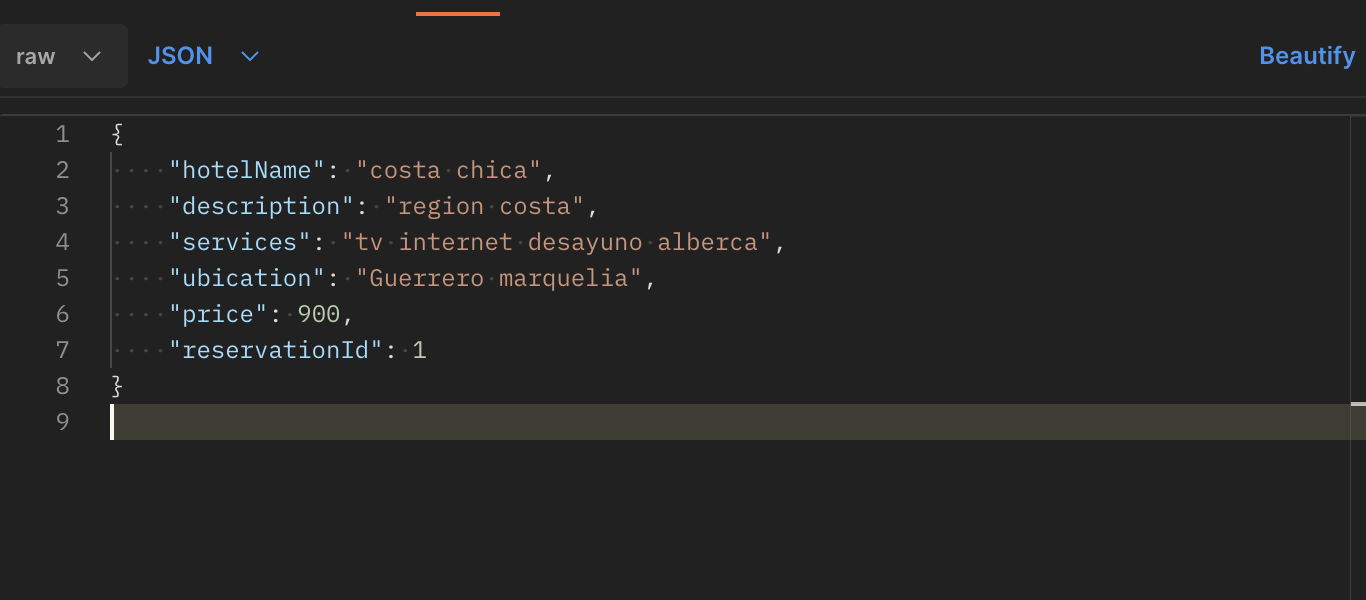
Las rutas hotel, invoice, payment, permanece, user, se crean con el fin de obtener la información que se incluirá en la ruta reservación la cual será mostrado una vez ejecutado el post.

Para cada Post de solicitud se toma la información de los modelos de cada ruta.

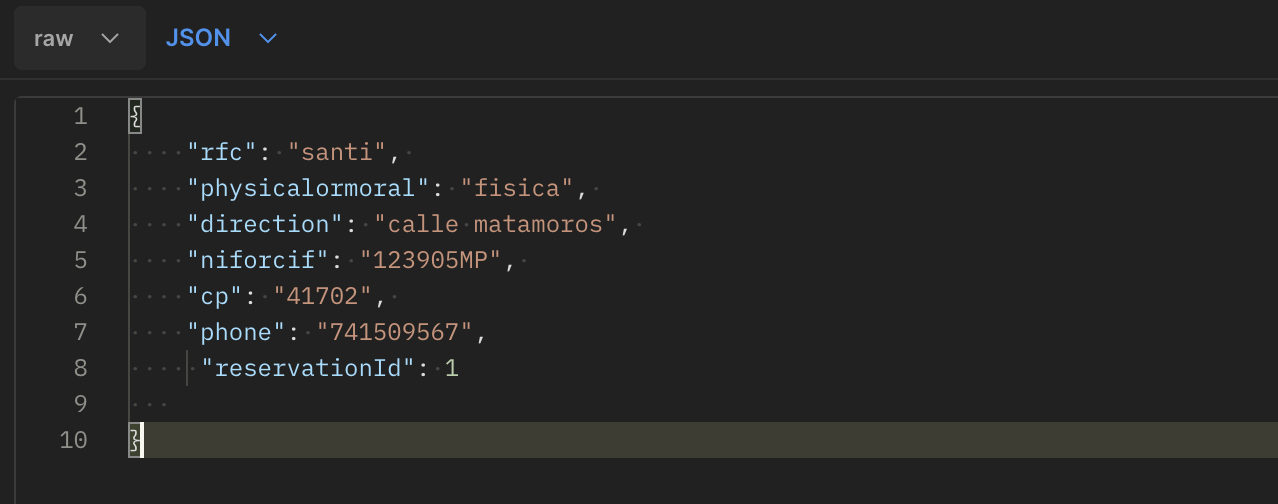


Las siguientes son imágenes de solicitud Post de cada ruta, mismas que realizaron en Postman

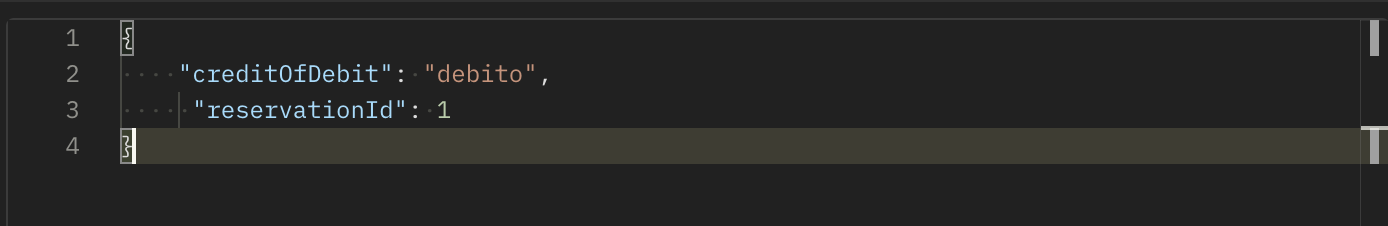
|  |  |  |
| --- | --- | --- |
| *hotel* | *crear* | *Post* |



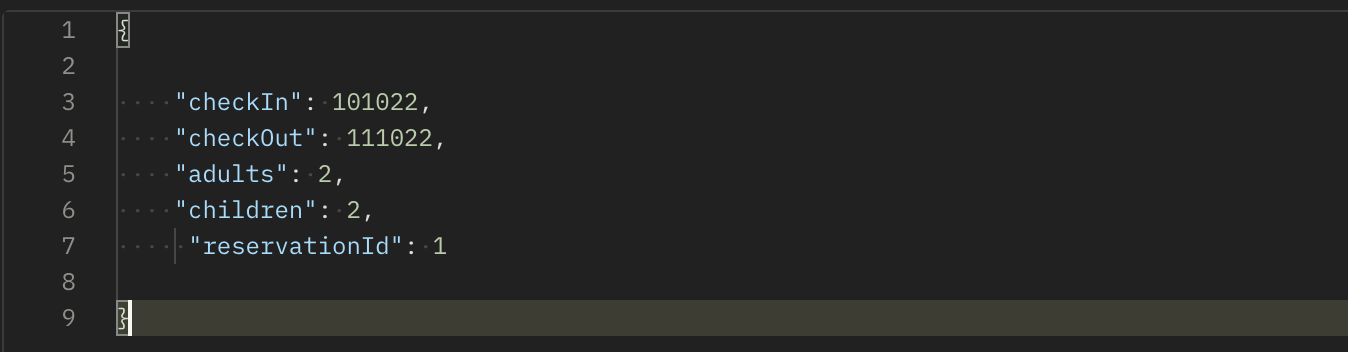
|  |  |  |
| --- | --- | --- |
| *Invoice* | *crear* | *Post* |



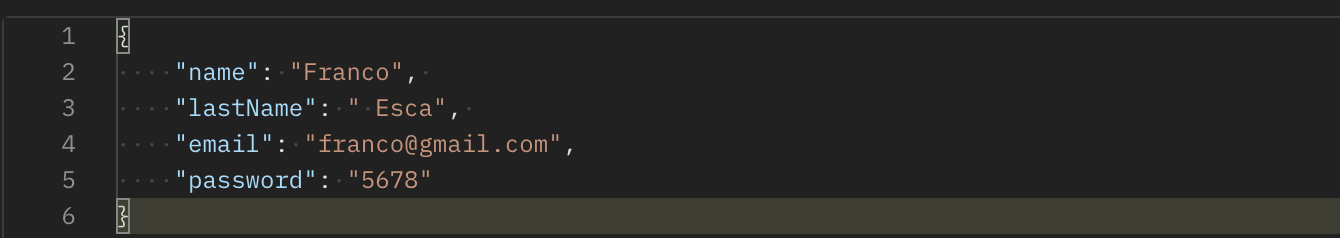
|  |  |  |
| --- | --- | --- |
| *Payment* | *crear* | *Post* |



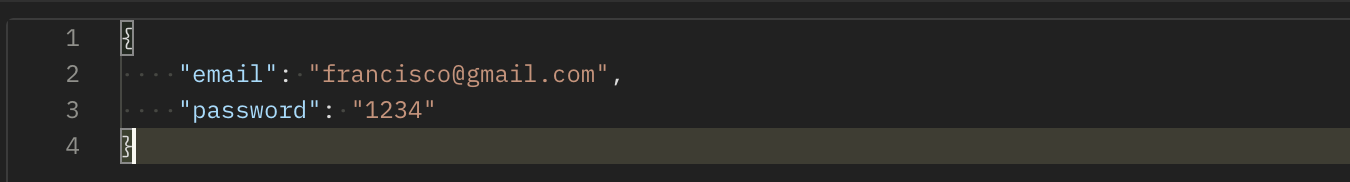
|  |  |  |
| --- | --- | --- |
| *Permanence* | *crear* | *Post* |



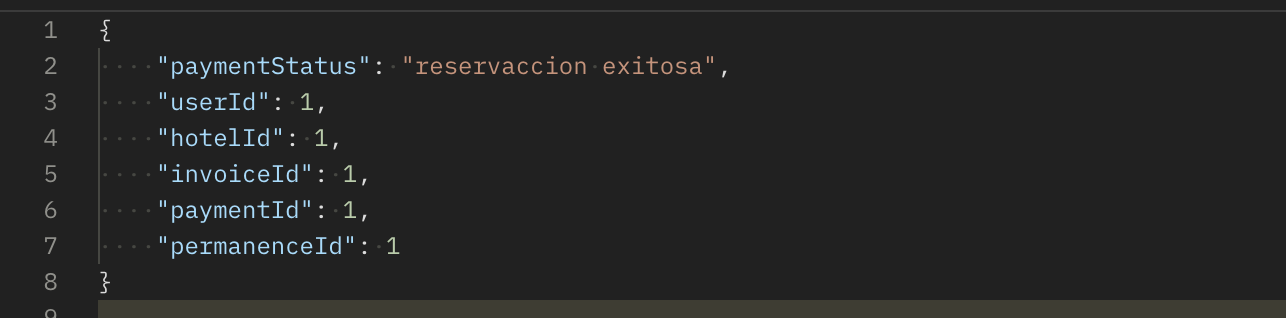
|  |  |  |
| --- | --- | --- |
| *User* | *Crear Usuario* | *Post* |



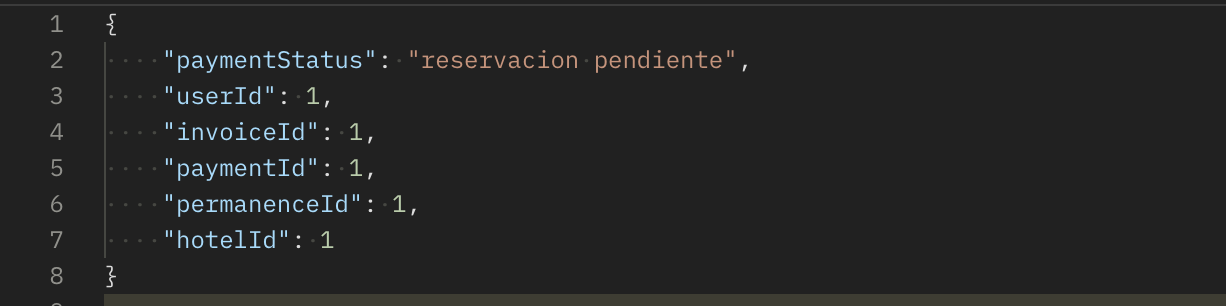
|  |  |  |
| --- | --- | --- |
| *User* | *login* | *Post* |



|  |  |  |
| --- | --- | --- |
| *Reservation* | *pago exit* | *Post* |



|  |  |  |
| --- | --- | --- |
| *Reservation* | *pago pendiente* | *Post* |



Ejemplo de Solicitud Post de reservación exitosa

<https://hotel-reservation12.herokuapp.com/api/v1/reservation/>

