

# **DOCUMENTACION**

## Tecnología utilizada Backend

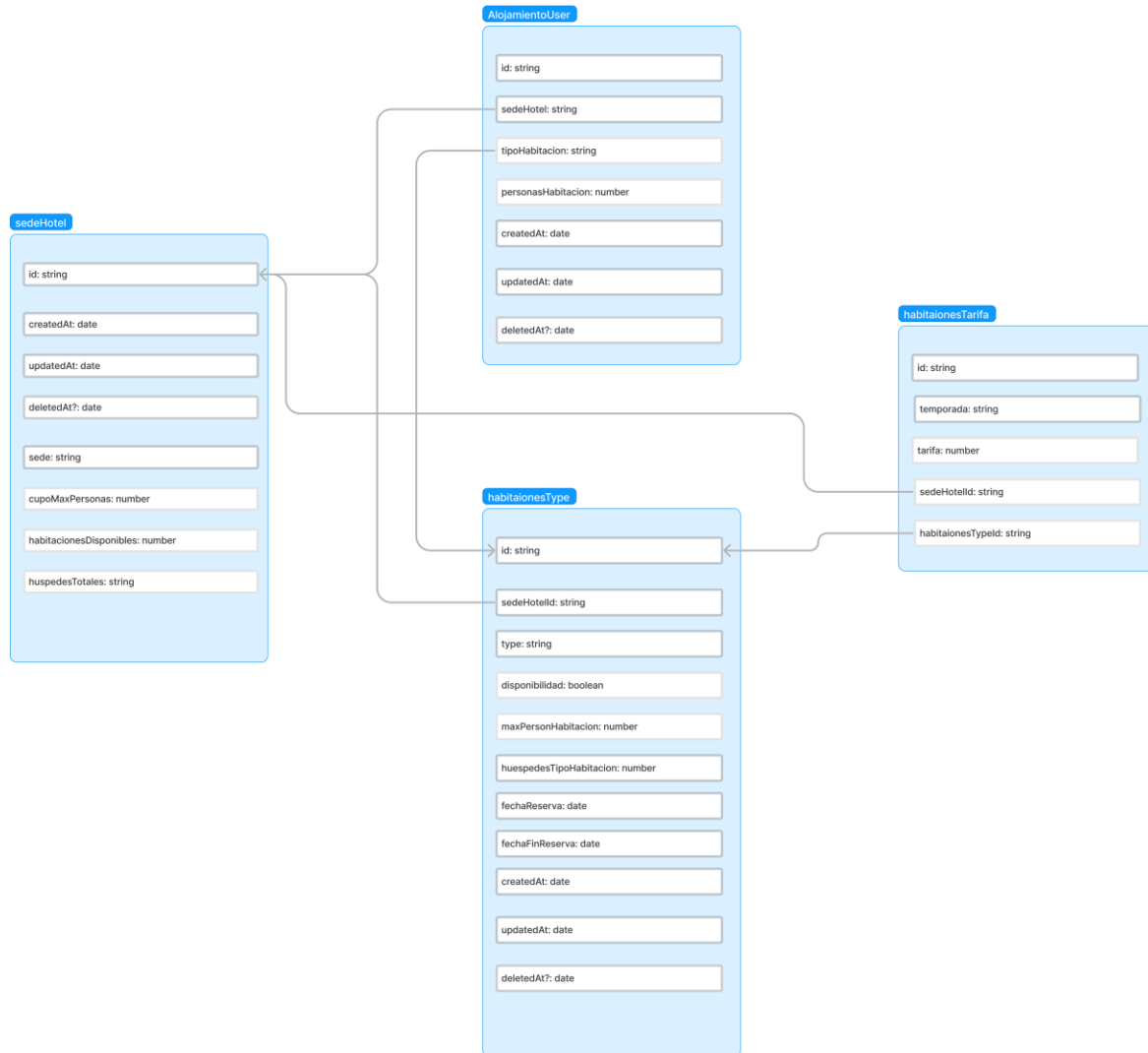
Para el desarrollo del api utilice **LoopBack 4** para desarrollar mi API debido a varias razones que destacan su robustez, flexibilidad y eficiencia en la creación de aplicaciones web y servicios RESTful.

LoopBack 4 está diseñado específicamente para la creación de APIs RESTful, proporcionando herramientas y características que simplifican la implementación de operaciones CRUD y la gestión de modelos de datos. Su capacidad para generar automáticamente documentación Swagger también facilita la integración y el uso de la API por parte de otros desarrolladores.

Como gestor de base de datos se manejo MongoDB ya que LoopBack proporciona una fácil integración y manejo de diversas fuentes de datos.

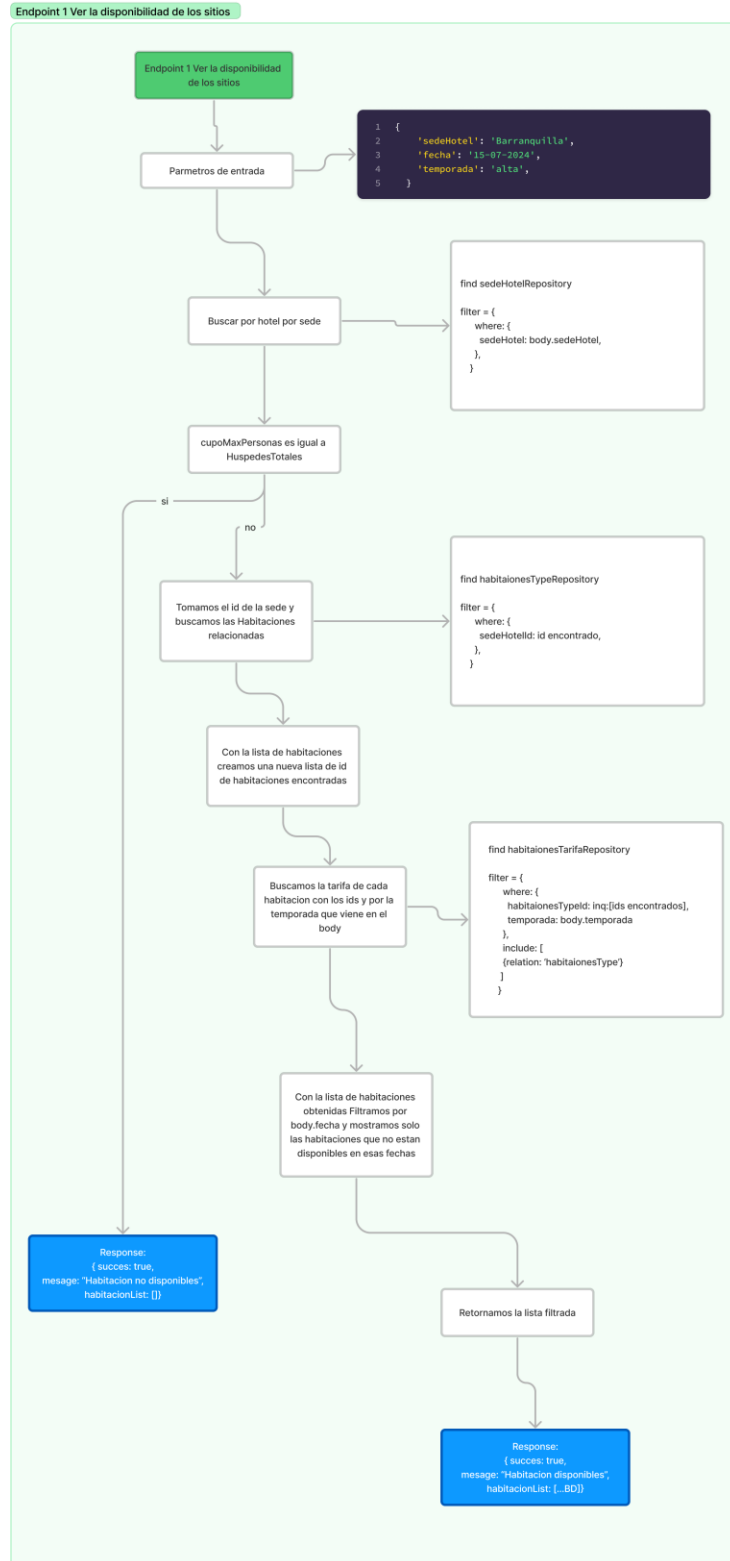
Se manejo el patrón de diseño de controladores y repositorios, lo que proporciona una separación clara entre la lógica de negocio y el acceso a datos.

### DIAGRAMA BASE DE DATOS



Antes de realizar la codificación se realizo un pequeño flujo lógico del comportamiento esperado del endpoint que permite ver la disponibilidad de los sitios

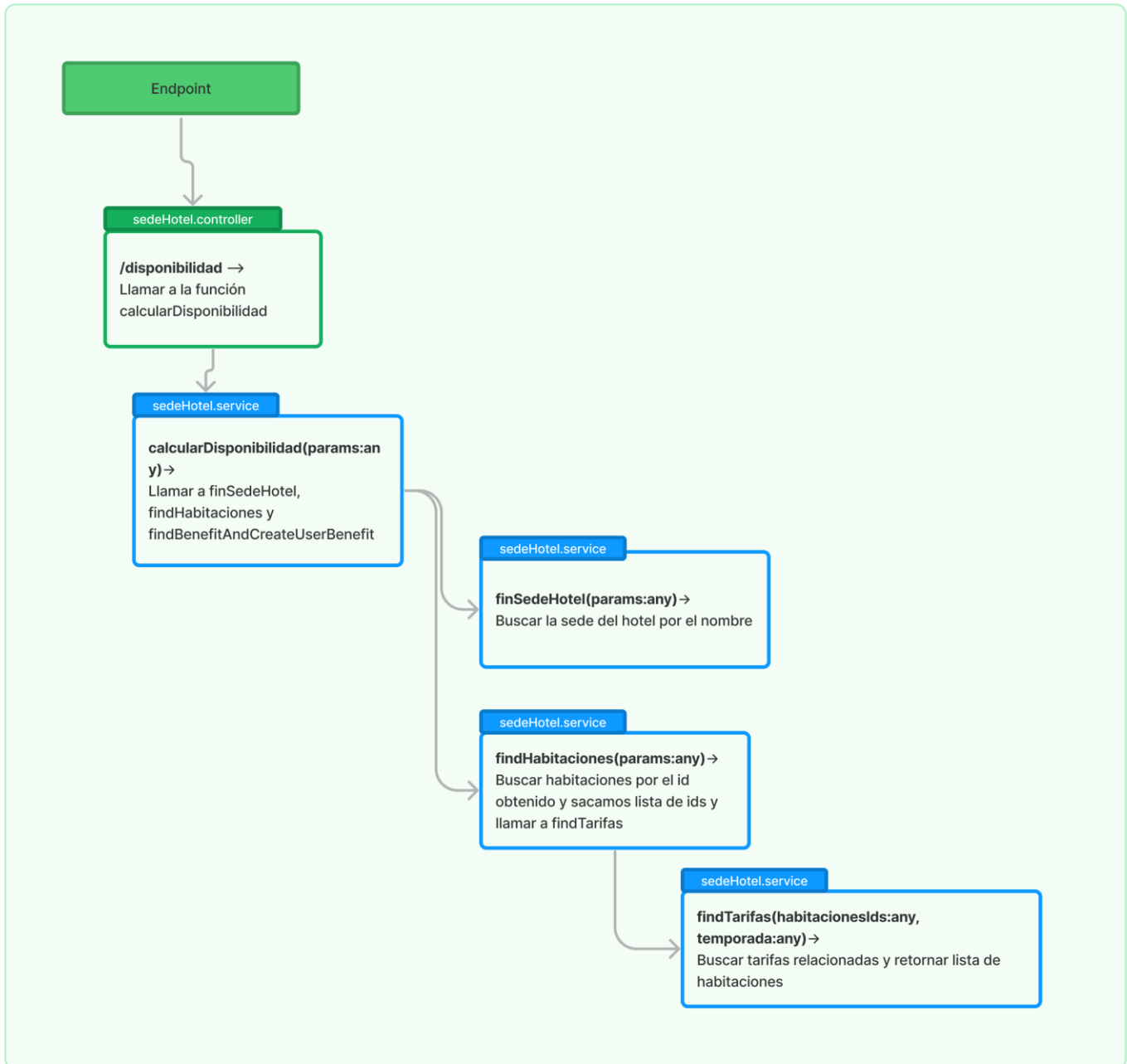
## Flujo Lógico Planteado Endpoint



## Solución Técnica Planteada Endpoint

Una vez realizado el flujo lógico se planteó una solución técnica la cual nos muestra a grandes rasgos como esta compuesto el endpoint desde su path hasta cada función que utiliza. Además, también nos permite ubicar de una manera más clara donde están ubicadas estas funciones para si se da el caso saber dónde están y corregirlas

### Endpoint



## Prueba Postman

Para la prueba con postman primero se ingresaron datos de prueba en este caso la sede del hotel:

[HTTP](#) axede / Guardar Sede

**POST** ▼ http://[::1]:3000/sede-hotels

Params Authorization Headers (8) **Body** ● Pre-request Script Tests

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL

```
1  {
2    "sede": "Barranquilla",
3    "cupoMaxPersonas": 132,
4    "habitacionesDisponibles": 30,
5    "huespedesTotales": 0,
6    "createdAt": "2024-04-17T18:29:38.699+00:00",
7    "updatedAt": "2024-04-17T18:29:38.699+00:00"
8  }
9
```

Luego se ingreso el tipo de habitación que ofrece el hotel:

[HTTP](#) axede / Guardar Habitacion Type

**POST** ▼ http://[::1]:3000/habitaiones-types

Params Authorization Headers (8) **Body** ● Pre-request Script Tests Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL J

```
1  {
2    "sedeHotelId": "6657af9fd2b8255ce87c678b",
3    "type": "estandar",
4    "disponibilidad": true,
5    "maxPersonHabitacion": 4,
6    "huespedesTipoHabitacion": 0,
7    "fechaReserva": "2024-06-17T18:29:38.699+00:00",
8    "fechaFinReserva": "2024-06-20T18:29:38.699+00:00",
9    "createdAt": "2024-05-29T18:29:38.699+00:00",
10   "updatedAt": "2024-05-29T18:29:38.699+00:00"
11 }
12
```

Y por ultimo se le asigno una tarifa a esta habitación ingresada:

axede / Guardar Tarifas Habitacion

POST http://[::1]:3000/habitaiones-tarifas

Params Authorization Headers (8) Body ● Pre-request Script Tests

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL

```
1 {
2   "temporada": "alta",
3   "tarifa": 35000,
4   "sedeHotelId": "6657af9fd2b8255ce87c678b",
5   "habitacionesTypeId": "6657b53ce0e09c2a34709f6d",
6   "createdAt": "2024-05-29T18:29:38.699+00:00",
7   "updatedAt": "2024-05-29T18:29:38.699+00:00"
8 }
9
```

Una vez ingresados datos de prueba, se procede a probar el endpoint realizado:


axede / Disponibilidad


POST http://[::1]:3000/disponibilidad

Params Authorization Headers (8) Body ● Pre-request Script Tests Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL ☒ JSON

```
1 {
2   "sedeHotel": "Barranquilla",
3   "fecha": "2024-06-13T18:29:38.699+00:00",
4   "temporada": "alta"
5 }
```

Body Cookies Headers (8) Test Results  Status: 200

Pretty Raw Preview Visualize JSON 

```
1 {
2   "suces": true,
3   "mesage": "Habitaciones disponibles",
4   "habitacionList": [
5     {
6       "id": "6657b6ae6d8c901910a40776",
7       "temporada": "alta",
8       "tarifa": 35000,
9       "habitacionesTypeId": "6657b53ce0e09c2a34709f6d",
10      "sedeHotelId": "6657af9fd2b8255ce87c678b",
11    }
12  ]
13 }
```

El endpoint me retorna un mensaje de disponibilidad y un arreglo de las habitaciones que están disponibles para ese momento

**Nota:** En el repositorio adjunto la colección de postman