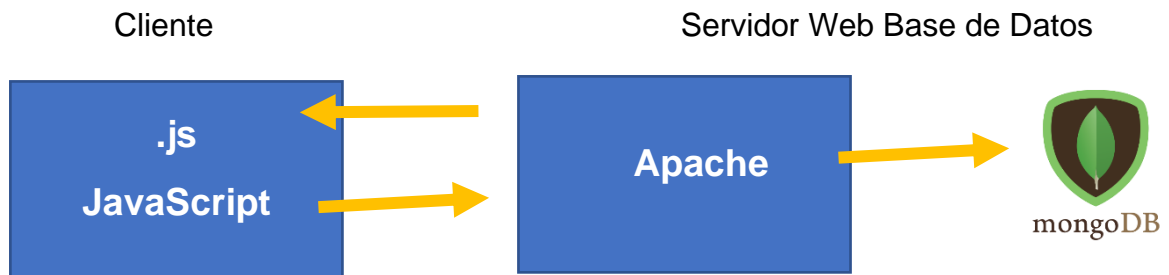


Examen primer parcial

Nombre: Diego Alejandro Guzmán Cajas

NRC: 5450

Architecture Design



URI design

Método	Post
URI	http://localhost:3000/api/create
Body	{ nombre <i>String</i> , pisos : <i>Number</i> , color : <i>String</i> , direccion : <i>String</i> , }
Formato	JSON
Devuelve	Build registrado en formato JSON

REST Service

```
JS index.js X JS services.js JS model.js
JS index.js > ...
1
2 const cors = require('cors');
3 const express = require('express');
4 const mongoose = require('mongoose');
5 const mongoString = "mongodb+srv://admin:<admin>@cluster0.z6oszhk.mongodb.net/test";
6
7 mongoose.connect(mongoString);
8 const database = mongoose.connection;
9
10 database.on('error', (error) => {
11   console.log(error)
12 })
13
14 database.once('connected', () => {
15   console.log('Database Connected');
16 })
17 const app = express();
18 app.use(cors());
19 app.use(express.json());
20
21 const routes = require('./services');
22
23 app.use('/api', routes)
24
25 app.listen(3000, () => {
26   console.log(`Server Started at ${3000}`)
27 })
```

```
JS index.js JS services.js JS model.js X
JS model.js > ...
1 const mongoose = require('mongoose');
2
3 const dataSchema = new mongoose.Schema({
4   nombre: {
5     required: true,
6     type: String
7   },
8   color: {
9     required: true,
10    type: String
11  },
12  direccion: {
13    required: true,
14    type: String
15  },
16  pisos: {
17    required: true,
18    type: Number
19  }
20 })
21
22 module.exports = mongoose.model('Data', dataSchema)
```

```
JS index.js JS services.js JS model.js
JS services.js > ...
1  const express = require('express');
2  const Building = require('./model');
3  const router = express.Router();
4
5  router.post('/create', async (req, res) => {
6      const data = new Building({
7          nombre: req.body.nombre,
8          pisos: req.body.pisos,
9          color: req.body.color,
10         direccion: req.body.direccion,
11     })
12
13     try {
14         const dataToSave = await data.save();
15         res.status(200).json(dataToSave)
16     }
17     catch (error) {
18         res.status(400).json({message: error.message})
19     }
20 })
21
22 module.exports = router;
```

Database data

BUILDINGSOZMAN > BUILDINGS > DATABASES

ClusterO

VERSION 5.0.9 REGION AWS N. Virginia (us-east-1)

Overview Real Time Metrics Collections Search Profiler Performance Advisor Online Archive Cmd Line Tools

DATABASES: 1 COLLECTIONS: 1

+ Create Database

Search Namespaces

Buildings

Buildings

Buildings.Buildings

STORAGE SIZE: 20KB TOTAL DOCUMENTS: 1 INDEXES TOTAL SIZE: 20KB

Find Indexes Schema Anti-Patterns Aggregation Search Indexes

INSERT DOCUMENT

FILTER { field: "value" } OPTIONS Apply Reset

QUERY RESULTS: 1-2 OF 2

```
{
  "_id": "62b6589c3ec48f710b0a9012",
  "nombre": "Dulce",
  "pisos": "3",
  "color": "Azul",
  "direccion": "Pomasqui"
}
```

```
{
  "_id": "62b6589f9a26c40414586f22",
  "nombre": "Diamante",
  "pisos": "7",
  "color": "Blanco",
  "direccion": "Quito"
}
```

Clean code