

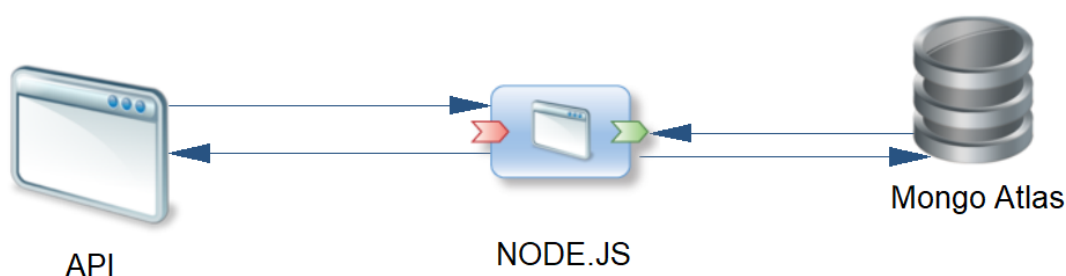


Nombres y Apellidos: Andrés Cosios Pineda

Asignatura: Desarrollo de Aplicaciones WEB

Examen

1. Architecture design:5 points.



2. URI design:5 points.

Metodo	GET
URI	http://localhost:4000/api/sports/all
Parametros	all
Formato	JSON
Devuelve	<pre>1 [2 { 3 "_id": "62b676648f6ccad9c3de7a83", 4 "impact": "medio", 5 "kind": "pelota", 6 "name": "footbal", 7 "place": "America", 8 "__v": "0" 9 }, 10]</pre>



3. REST Service:10 points.

Postman interface showing a GET request to `http://localhost:4000/api/sports/all`. The response is a JSON object:

```
{
  "_id": "62b676648f6ccad9c3de7a83",
  "impact": "medio",
  "kind": "pelota",
  "name": "fútbol",
  "place": "America",
  "___v": 0
}
```

4. Database data:5 points.

Postman interface showing a GET request to `http://localhost:4000/api/sports/all`. The response is a JSON object:

```
{
  "_id": "62b676648f6ccad9c3de7a83",
  "impact": "medio",
  "kind": "pelota",
  "name": "fútbol",
  "place": "America",
  "___v": 0
}
```



5. Clean Code:5 points.

```
1 const mongoose = require('mongoose');
2 const Sport = require('../models/Sport');
3
4 const findAllSports = (req, res) => {
5   Sport.find((err, sports)=>{
6     err && res.status(500).send(err.message);
7
8     res.status(200).json(sports);
9   })
10 }
11
12 const findById = (req, res) => {
13   Sport.findById(req.params.id, (err, sports)=> {
14     err && res.status(500).send(err.message);
15
16     res.status(200).json(sports);
17   })
18 }
19
20 const addSport = (req, res) => {
21
22   let sport = new Sport({
23     name: req.body.name,
24     kind: req.body.kind,
25     place: req.body.place,
26     impact: req.body.impact,
27   });
28 }
```