

Wheels

la app de viajes compartidos

¿Qué es Wheels?

Wheels es un aplicativo web que tiene como finalidad facilitar a sus usuarios, la búsqueda y reserva de viajes compartidos a destinos locales o nacionales. Contamos con una plataforma intuitiva y dinámica, ofreciendo una experiencia única en su clase.

Creación y despliegue del microservicio “travels-ms”. Publicación, consulta, edición y eliminación de viajes

- ✓ Se creo el microservicio ‘travels-ms’ el cual maneja dos entidades de datos, ‘travel’ y ‘city’, en donde la primera permite almacenar toda la información de los viajes y la segunda se limita únicamente a gestionar los códigos y nombres de las distintas ciudades.

Schemas	
Travel ▾ {	
idTravel	string
idDriver	string
nameDriver	string
fromPlace	string
toPlace	string
passThrough	string
published	string(\$date-time)
dateTravel	string(\$date-time)
seats	integer(\$int32)
price	number(\$double)
}	
City ▾ {	
codeCity	string
nameCity	string
}	

- ✓ Se uso Java Spring Boot y MongoDB Atlas como gestor y base de datos, para el desarrollo de este microservicio
- ✓ Se implemento la Herramienta Swagger UI durante el testing y desarrollo del microservicio.

travel-controller			^
GET	/travels/{idTravel}		▼
PUT	/travels/{idTravel}		▼
DELETE	/travels/{idTravel}		▼
GET	/travels		▼
POST	/travels		▼
GET	/travels/{toPlace}		▼
GET	/travels/{passthrough}		▼
GET	/travels/{idDriver}		▼
GET	/travels/{fromPlace}		▼
city-controller			^
GET	/city/{codeCity}		▼
PUT	/city/{codeCity}		▼
DELETE	/city/{codeCity}		▼
POST	/city		▼
GET	/cities		▼

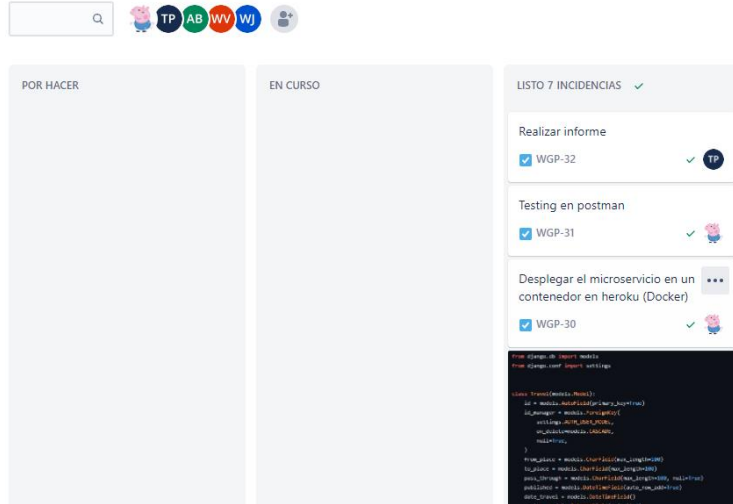
1. Asignación de tareas y flujo de trabajo en Jira <https://proyecto-wheels.atlassian.net/jira/software/projects/WGP/boards/1>

Proyectos / Wheels-Grupo6-P53


Sprint 3

Q

POR HACER	EN CURSO	LISTO 7 INCIDENCIAS ✓
		<div>Se debe crear el modelo de travels</div> <div> <input checked="" type="checkbox"/> WGP-21 ✓ </div>
		<div>Se debe crear el modelo city</div> <div> <input checked="" type="checkbox"/> WGP-27 ✓ </div>
		<div>Crear la base de datos, generar las credenciales y realizar la conexión</div> <div> <input checked="" type="checkbox"/> WGP-20 ✓ </div>
		<div>Crear controllers</div> <div> <input checked="" type="checkbox"/> WGP-24 ✓ </div>



- Despliegue del microservicio en un contenedor de Heroku <https://travels-ms.herokuapp.com/swagger-ui/index.html?configUrl=/v3/api-docs/swagger-config>

 **Swagger**
Supported by SMARTBEAR

/v3/api-docs

Explore

OpenAPI definition

v0 OAS3

/v3/api-docs

Servers

https://travels-ms.herokuapp.com - Generated server url

travel-controller

POST

/travels

GET

/travels/{idTravel}

city-controller

POST

/city

GET

/city/{codeCity}

3. Repositorio en GitHub del microservicio https://github.com/Grupo6-P53-MinTIC/Travel_ms.git

Overview
Yours
Active
Stale
All branches

Search branches...

Default branch

main Updated 9 hours ago by TaniaPedrazaG

Default

Your branches

dev	Updated 1 minute ago by BayronDavid	8 2	#7	Open		
revert-1-feat/adriana	Updated 11 hours ago by BayronDavid	13 0	#3	Merged		
feat/wilson	Updated 11 hours ago by BayronDavid	14 0	#2	Merged		

Active branches

dev	Updated 1 minute ago by BayronDavid	8 2	#7	Open		
feat/tania	Updated 9 hours ago by TaniaPedrazaG	8 0	#6	Merged		
feat/adriana	Updated 10 hours ago by TaniaPedrazaG	14 0	#4	Merged		
revert-1-feat/adriana	Updated 11 hours ago by BayronDavid	13 0	#3	Merged		
feat/wilson	Updated 11 hours ago by BayronDavid	14 0	#2	Merged		

View more active branches >

dev

8 branches

0 tags

Go to file

Add file

Code

This branch is 2 commits ahead, 8 commits behind main.

#7

BayronDavid

Crud para cada uno de los modelos

5a2e37b 19 seconds ago

11 commits

.mvn/wrapper	first commit	5 days ago
src	Crud para cada uno de los modelos	19 seconds ago
.gitignore	first commit	5 days ago
Dockerfile	Errores en Get	1 hour ago
mvnw	Errores en Get	1 hour ago
mvnw.cmd	Errores en Get	1 hour ago
pom.xml	Errores en Get	1 hour ago

Help people interested in this repository understand your project by adding a README.

Add a README

4. Base de datos MongoDB Atlas

The screenshot displays the MongoDB Atlas web interface. On the left, a sidebar shows the database structure: 'travels_ms_wheels' containing 'city' and 'travel' namespaces. The main panel is split into two sections, each showing a different collection.

travels_ms_wheels.travel

COLLECTION SIZE: 714B TOTAL DOCUMENTS: 3 INDEXES TOTAL SIZE: 36KB

Find Indexes Schema Anti-Patterns 0 Aggregation Search Indexes ●

FILTER { field: 'value' } OPTIONS

QUERY RESULTS 1-3 OF 3

```
{
  "_id": "002",
  "idDriver": "102",
  "nameDriver": "Pedro",
  "fromPlace": "Mocoa",
  "toPlace": "Pasto",
  "passThrough": "Sibundoy",
  "published": "2021-11-29T05:07:15.741+00:00",
  "dateTravel": "2021-11-29T05:07:15.741+00:00",
  "seats": 3,
  "price": 50000,
  "_class": "com.example.travels_ms.models.Travel"
}
```

travels_ms_wheels.city

COLLECTION SIZE: 170B TOTAL DOCUMENTS: 2 INDEXES TOTAL SIZE: 36KB

Find Indexes Schema Anti-Patterns 0 Aggregation Search Indexes ●

FILTER { field: 'value' } OPTIONS

QUERY RESULTS 1-2 OF 2

```
{
  "_id": "11001",
  "nameCity": "Bogota",
  "_class": "com.example.travels_ms.models.City"
}
```

```
{
  "_id": "13001",
  "nameCity": "Cartagena",
  "_class": "com.example.travels_ms.models.City"
}
```

5. Testing con Swager UI

Travel controller

Post de un viaje

The image shows the Swagger UI interface for a REST API. The top section displays a POST request to the endpoint `/travels`. The request body is a JSON object with the following fields: `idTravel`, `idDriver`, `nameDriver`, `fromPlace`, `toPlace`, `passThrough`, `published`, `dateTravel`, `seats`, and `price`. The `published` and `dateTravel` fields are timestamps. The `seats` field is an integer, and the `price` field is a number. The bottom section shows the server response, which is a 200 status code with a JSON body containing the same data as the request. The response headers include `connection: keep-alive`, `content-type: application/json`, `date: Mon, 29 Nov 2021 04:57:56 GMT`, `keep-alive: timeout=60`, and `transfer-encoding: chunked`.

POST /travels

Parameters Cancel Reset

No parameters

Request body required application/json

```
{
  "idTravel": "001",
  "idDriver": "101",
  "nameDriver": "Marcos",
  "fromPlace": "Zipaquira",
  "toPlace": "Bogota",
  "passThrough": "Chia",
  "published": "2021-11-29T04:55:10.239Z",
  "dateTravel": "2021-11-29T04:55:10.239Z",
  "seats": 3,
  "price": 12000
}
```

Execute

Request URL
`http://localhost:8080/travels`

Server response

Code Details

200

Response body

```
{
  "idTravel": "001",
  "idDriver": "101",
  "nameDriver": "Marcos",
  "fromPlace": "Zipaquira",
  "toPlace": "Bogota",
  "passThrough": "Chia",
  "published": "2021-11-29T04:55:10.239+00:00",
  "dateTravel": "2021-11-29T04:55:10.239+00:00",
  "seats": 3,
  "price": 12000
}
```

Download

Response headers

```
connection: keep-alive
content-type: application/json
date: Mon, 29 Nov 2021 04:57:56 GMT
keep-alive: timeout=60
transfer-encoding: chunked
```

Consulta de un viaje

GET

/tarvels/{idTravel}

⌵

Parameters

Cancel

Name	Description
idTravel * required	
string	001
(path)	

Execute

Clear

Request URL

http://localhost:8080/tarvels/001

Server response

Code

Details

200

Response body

```
{
  "idTravel": "001",
  "idDriver": "101",
  "nameDriver": "Marcos",
  "fromPlace": "Zipaquirá",
  "toPlace": "Bogotá",
  "passThrough": "Chía",
  "published": "2021-11-29T04:55:10.239+00:00",
  "dateTravel": "2021-11-29T04:55:10.239+00:00",
  "seats": 3,
  "price": 12000
}
```

Download

Response headers

```
connection: keep-alive
content-type: application/json
date: Mon, 29 Nov 2021 05:00:35 GMT
keep-alive: timeout=60
transfer-encoding: chunked
```

Edición de un viaje

PUT

/tarvels/{idTravel}

⌵

Parameters

Cancel

Reset

Name	Description
idTravel * required	
string	001
(path)	

Request body

required

application/json

⌵

```
{
  "idTravel": "001",
  "idDriver": "101",
  "nameDriver": "Marcos",
  "fromPlace": "Zipaquirá",
  "toPlace": "Sopchá",
  "passThrough": "Gadisa",
  "published": "2021-11-29T04:55:10.239+00:00",
  "dateTravel": "2021-11-29T04:55:10.239+00:00",
  "seats": 7,
  "price": 15000
}
```

Execute

Request URL	
<code>http://localhost:8080/travels/001</code>	
Server response	
Code	Details
200	<p>Response body</p> <pre>{ "idTravel": "001", "idDriver": "101", "nameDriver": "Marcos", "fromPlace": "Zipaquira", "toPlace": "Zipaquira", "passThrough": "Cajica", "published": "2021-11-29T04:55:10.239+00:00", "dateTravel": "2021-11-29T04:55:10.239+00:00", "seats": 7, "price": 15000 }</pre> <p>Response headers</p> <pre>connection: keep-alive content-type: application/json date: Mon, 29 Nov 2021 05:04:53 GMT keep-alive: timeout=60 transfer-encoding: chunked</pre>

Eliminación de un viaje

<div> DELETE /travels/{idTravel} </div> <div> Parameters Cancel </div> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>idTravel * required</td> <td></td> </tr> <tr> <td>string</td> <td>001</td> </tr> <tr> <td>(path)</td> <td></td> </tr> </tbody> </table> <div>Execute</div>		Name	Description	idTravel * required		string	001	(path)	
Name	Description								
idTravel * required									
string	001								
(path)									
<p>Curl</p> <pre>curl -X 'DELETE' \ 'http://localhost:8080/travels/001' \ -H 'accept: */*'</pre> <p>Request URL</p> <pre>http://localhost:8080/travels/001</pre> <p>Server response</p> <table border="1"> <thead> <tr> <th>Code</th> <th>Details</th> </tr> </thead> <tbody> <tr> <td>200</td> <td> <p>Response headers</p> <pre>connection: keep-alive content-length: 0 date: Mon, 29 Nov 2021 05:05:53 GMT keep-alive: timeout=60</pre> </td> </tr> </tbody> </table>		Code	Details	200	<p>Response headers</p> <pre>connection: keep-alive content-length: 0 date: Mon, 29 Nov 2021 05:05:53 GMT keep-alive: timeout=60</pre>				
Code	Details								
200	<p>Response headers</p> <pre>connection: keep-alive content-length: 0 date: Mon, 29 Nov 2021 05:05:53 GMT keep-alive: timeout=60</pre>								

Consulta de todos los viajes

<div> GET /travels </div> <div> Parameters Cancel </div> <p>No parameters</p> <div>Execute</div>	
--	--

Curl

```
curl -X 'GET' \
  'http://localhost:8080/travels' \
  -H 'accept: */*'
```

Request URL

http://localhost:8080/travels

Server response

Code Details

200

Response body

```
{
  "idTravel": "002",
  "idDriver": "102",
  "nameDriver": "Pedro",
  "fromPlace": "Mocoa",
  "toPlace": "Pasto",
  "passThrough": "Sibundoy",
  "published": "2021-11-29T05:07:15.741+00:00",
  "dateTravel": "2021-11-29T05:07:15.741+00:00",
  "seats": 3,
  "price": 50000
},
{
  "idTravel": "003",
  "idDriver": "103",
  "nameDriver": "Maria",
  "fromPlace": "Neiva",
  "toPlace": "Pitalito",
  "passThrough": "Garzon",
  "published": "2021-11-29T05:07:15.741+00:00",
  "dateTravel": "2021-11-29T05:07:15.741+00:00",
  "seats": 4,
  "price": 15000
},
{
  "idTravel": "004",
  "idDriver": "104",
  "nameDriver": "Enith"
}
```

Response headers

```
connection: keep-alive
content-type: application/json
date: Mon, 29 Nov 2021 05:10:45 GMT
keep-alive: timeout=60
transfer-encoding: chunked
```

Endpoints adicionales

- Obtener viajes por el lugar de salida
- Obtener viajes por el lugar de destino
- Obtener viajes por los lugares que pasa
- Obtener los viajes por conductor

travel-controller

GET /tarvels/{idTravel}

PUT /tarvels/{idTravel}

DELETE /tarvels/{idTravel}

GET /travels

POST /travels

GET /travels/{toPlace}

GET /travels/{passThrough}

GET /travels/{idDriver}

GET /travels/{fromPlace}

City controller

Post de una ciudad

POST /city

Parameters

Cancel

Reset

No parameters

Request body required

application/json

```
{  "codeCity": "05001",  "nameCity": "Medellin"}  
```

Execute

Clear

Curl

```
curl -X 'POST' \  'http://localhost:8080/city' \  -H 'accept: */*' \  -H 'Content-Type: application/json' \  -d '{  "codeCity": "05001",  "nameCity": "Medellin"}'
```

Request URL

http://localhost:8080/city

Server response

Code

Details

200

Response body

```
{  "codeCity": "05001",  "nameCity": "Medellin"}  
```

Download

Response headers

```
connection: keep-alive  content-type: application/json  date: Mon, 29 Nov 2021 05:16:24 GMT  keep-alive: timeout=60  transfer-encoding: chunked
```

Obtener una ciudad

GET /city/{codeCity}

Parameters

Cancel

Name	Description
codeCity <small>* required</small>	
string	05001
(path)	

Execute

Curl

```
curl -X 'GET' \
  'http://localhost:8080/city/05001' \
  -H 'accept: */*'
```

Request URL

```
http://localhost:8080/city/05001
```

Server response

Code	Details
200	<p>Response body</p> <pre>{ "codeCity": "05001", "nameCity": "Medellin" }</pre> <p>Response headers</p> <pre>connection: keep-alive content-type: application/json date: Mon, 29 Nov 2021 05:21:08 GMT keep-alive: timeout=60 transfer-encoding: chunked</pre>

Editar una ciudad

PUT

/city/{codeCity}

Parameters

Cancel

Reset

Name	Description
codeCity * required	
string	05001
(path)	

Request body required

application/json

```
{
  "codeCity": "05001",
  "nameCity": "Bogota"
}
```

Execute

Clear

Curl

```
curl -X 'PUT' \
  'http://localhost:8080/city/05001' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
    "codeCity": "05001",
    "nameCity": "Bogota"
  }'
```

Request URL

http://localhost:8080/city/05001

Server response

Code	Details
200	<p>Response body</p> <pre>{ "codeCity": "05001", "nameCity": "Bogota" }</pre> <p>Response headers</p> <pre>connection: keep-alive content-type: application/json date: Mon, 29 Nov 2021 05:18:14 GMT keep-alive: timeout=60 transfer-encoding: chunked</pre>

Eliminar una ciudad

DELETE /city/{codeCity}

Parameters Cancel

Name	Description
codeCity * required string (path)	05001

Execute

Curl

```
curl -X 'DELETE' \
  'http://localhost:8080/city/05001' \
  -H 'accept: */*'
```

Request URL

http://localhost:8080/city/05001

Server response

Code	Details
200	<p>Response headers</p> <pre>connection: keep-alive content-length: 0 date: Mon, 29 Nov 2021 05:22:46 GMT keep-alive: timeout=60</pre>

Obtener Todas las ciudades

GET

/cities

Parameters

No parameters

Execute

Curl

```
curl -X 'GET' \
'http://localhost:8080/cities' \
-H 'accept: */*'
```

Request URL

```
http://localhost:8080/cities
```

Server response

Code

Details

200

Response body

```
[
  {
    "codeCity": "11001",
    "nameCity": "Bogota"
  },
  {
    "codeCity": "13001",
    "nameCity": "Cartagena"
  }
]
```

Download

Response headers

```
connection: keep-alive
content-type: application/json
date: Mon, 29 Nov 2021 05:25:12 GMT
keep-alive: timeout=60
transfer-encoding: chunked
```