

## API REST

Para el desarrollo de esta practica eh optado por el uso de formato JSON para la respuesta de los mensajes.

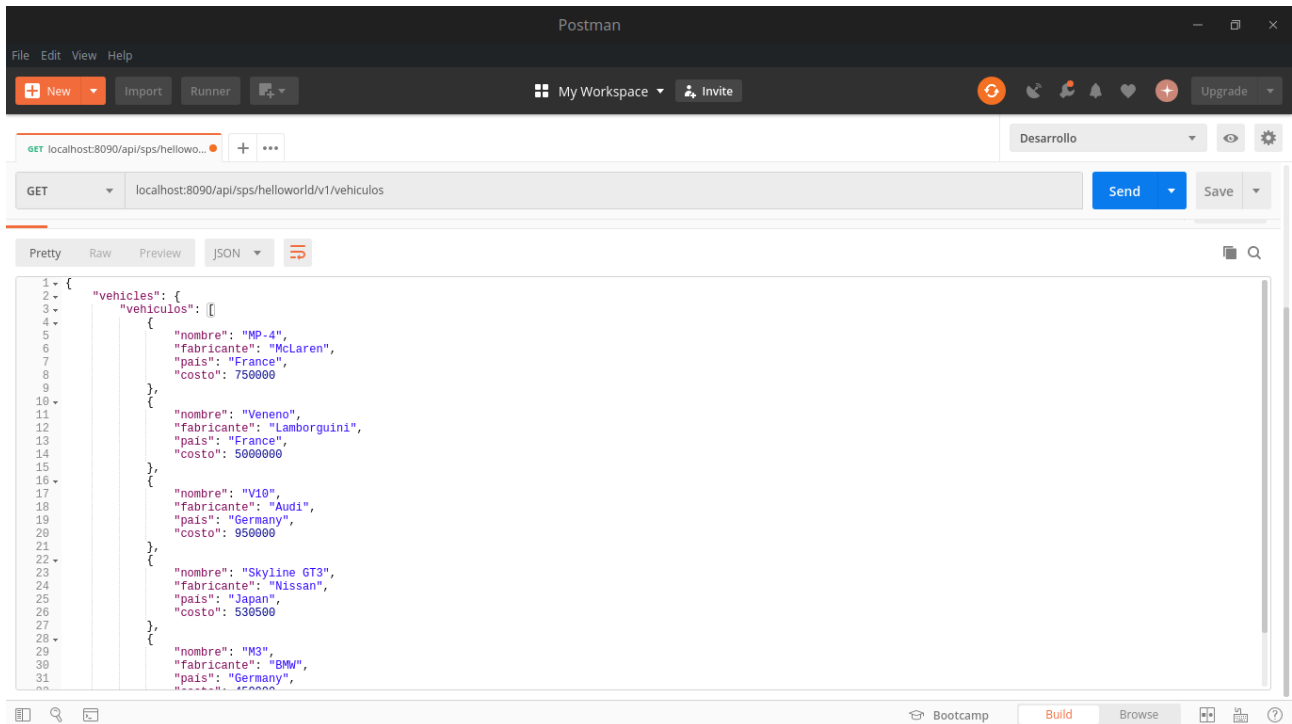
### 1. Mensajes de Entrada y Salida

Entrada	Salida
GET <i>/Vehiculo</i>	JSON con información sobre vehículos. Cada JSON esta compuesto por: <ul style="list-style-type: none"><li>-nombre</li><li>-fabricante</li><li>-país</li><li>-costo</li></ul>
GET <i>/libros</i>	JSON con información sobre libros. Cada JSON esta compuesto por: <ul style="list-style-type: none"><li>-título</li><li>-autor</li><li>-editorial</li></ul>
GET <i>/usuarios</i>	JSON con información sobre usuarios. Cada JSON esta compuesto por: <ul style="list-style-type: none"><li>-id</li><li>-name</li><li>-username</li><li>-email</li><li>-addres</li><li>-phone</li><li>-website</li><li>-company</li></ul>

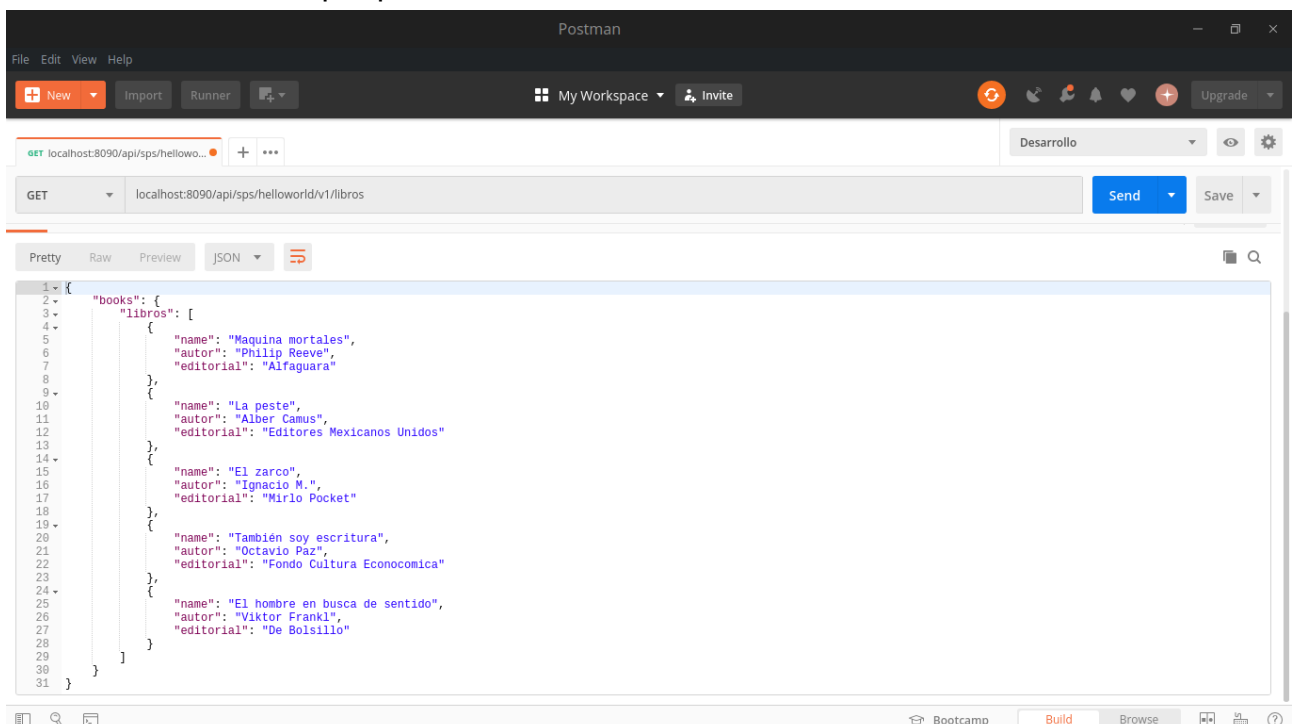
## 2. Pruebas con Postman

Para hacer las pruebas eh decidio hacer uso de Postman para hacer las peticiones al API REST.

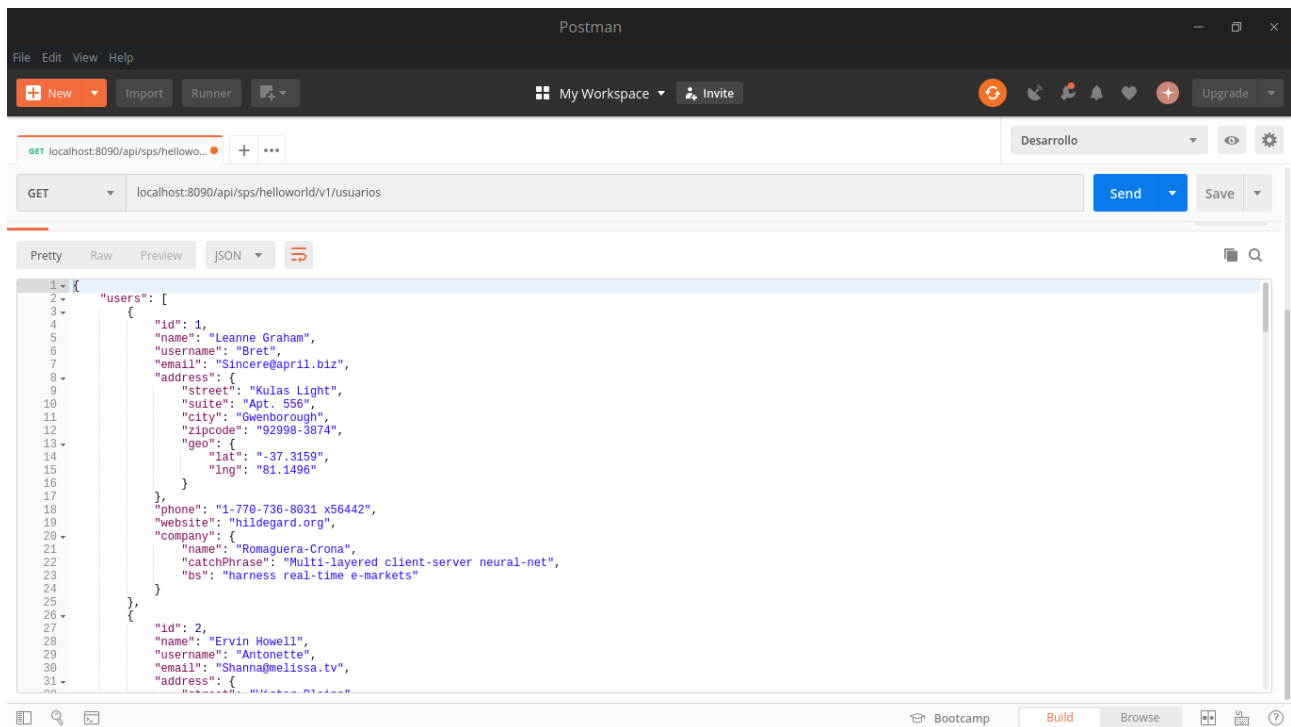
**GET** localhost:8090/api/sps/helloworld/v1/vehiculos



**GET** localhost:8090/api/sps/helloworld/v1/libros



## GET localhost:8090/api/sps/helloworld/v1/usuarios



### 3. Contenerizar la aplicación.

Para la creación del contenedor donde esta la aplicación, opté por el uso de Docker.

Primeramente, para crear el contenedor se deben agregar un par de archivos. El primero *Dockerfile*, para indicar la imagen que tendrá e indicar el directorio de trabajo, el puerto por dónde se expondrá el contenedor. El segundo archivo es *.dockerignore*, para indicar que archivos y/o directorios no deberán incluirse en el contenedor. El contenedor tendrá por nombre *practica-rest*.

```
Dockerfile ▶ ...
1 FROM node:10
2 ENV NODE_ENV production
3 WORKDIR /home/node/app
4
5 COPY package*.json ./
6 RUN npm install
7 COPY . .
8 EXPOSE 8090
9 CMD ["node", "server/server"]
```

Una vez creado e iniciado el contenedor, se comprueba que el puerto expuesto esta enlazado al mismo puerto del equipo anfitrión.

miguel@miguel-d:~/Documents/practicaAPI\$ docker ps						
CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
a9636bd60c1f	migue0edu/practica-rest	"docker-entrypoint.s..."	9 seconds ago	Up 6 seconds	0.0.0.0:8090->8090/tcp	sad_hellman

Desde la consola, con la ayuda del comando CURL se puede probar que el contenedor response a las solicitudes.

```
miguel@miguel-d:~/Documents/practicaAPI$ curl -i localhost:8090/api/sps/helloworld/v1/libros
HTTP/1.1 200 OK
X-Powered-By: Express
Content-Type: application/json; charset=utf-8
Content-Length: 434
ETag: W/"1b2-Adv0c+sEqFGwrSIF0U8F561FaFo"
Date: Wed, 10 Jul 2019 01:18:20 GMT
Connection: keep-alive

{"books":{"libros":[{"name":"Maquina mortales","autor":"Philip Reeve","editorial":"Alfaguara"}, {"name":"La peste","autor":"Alber Camus","editorial":"Editores Mexicanos Unidos"}, {"name":"El zarco","autor":"Ignacio M.","editorial":"Mirlo Pocket"}, {"name":"Tambi\u00e9n soy escritura","autor":"Octavio Paz","editorial":"Fondo Cultura Economica"}, {"name":"El hombre en busca de sentido","autor":"Viktor Frankl","editorial":"De Bolsillo"}]}}
```

O igualmente se puede hacer la prueba desde Postman.

