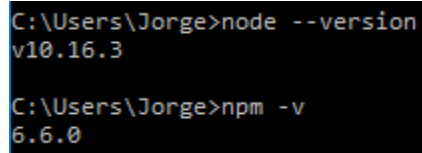


PRACTICA GRUPAL SERVICIOS WEB REST CON NODEJS

1. Instalar/Actualizar NodeJS 10.16.3 (Recommended For Most Users): <https://nodejs.org/en/> Validar la instalación ejecutando el siguiente comando:

```
$ node --version
```

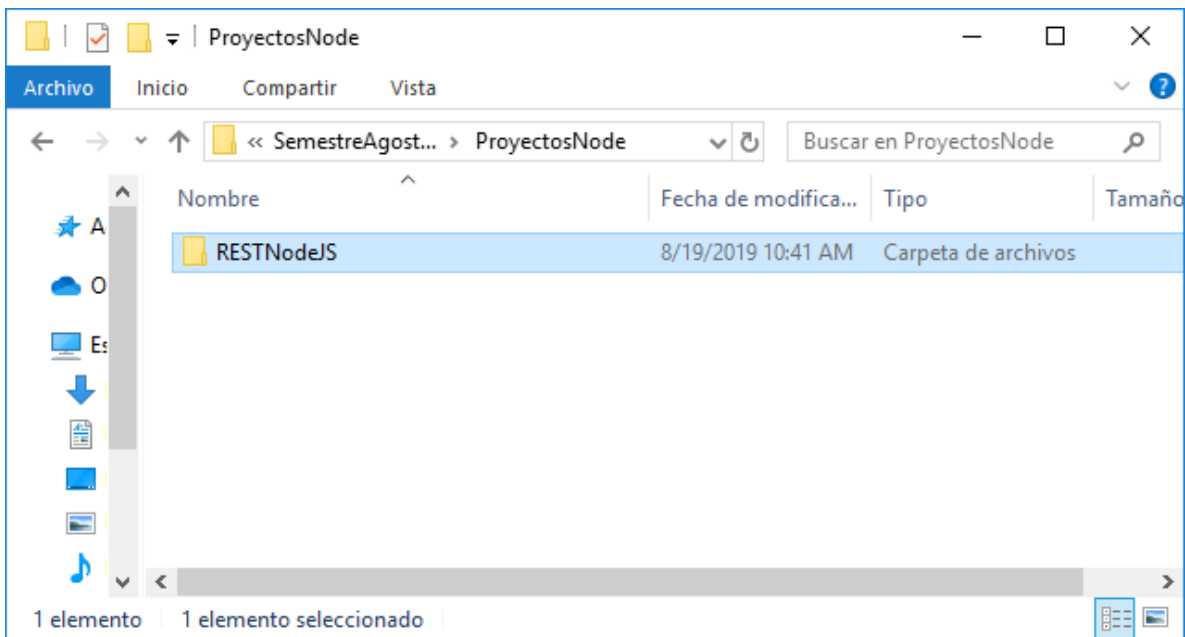
```
$ npm -v
```



```
C:\Users\Jorge>node --version
v10.16.3

C:\Users\Jorge>npm -v
6.6.0
```

2. Crear una carpeta para el proyecto. Es donde se colocarán todos los archivos necesarios para la práctica.



3. Abrir una ventana de comandos y ubicarse en la carpeta generada. Ejecutar el siguiente comando y especificar los datos que solicite:

```
$ npm init
```

```
C:\JMA\Personal\ITESM\SemestreAgostoDiciembre2019\ProyectosNode\RESTNodeJS>npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

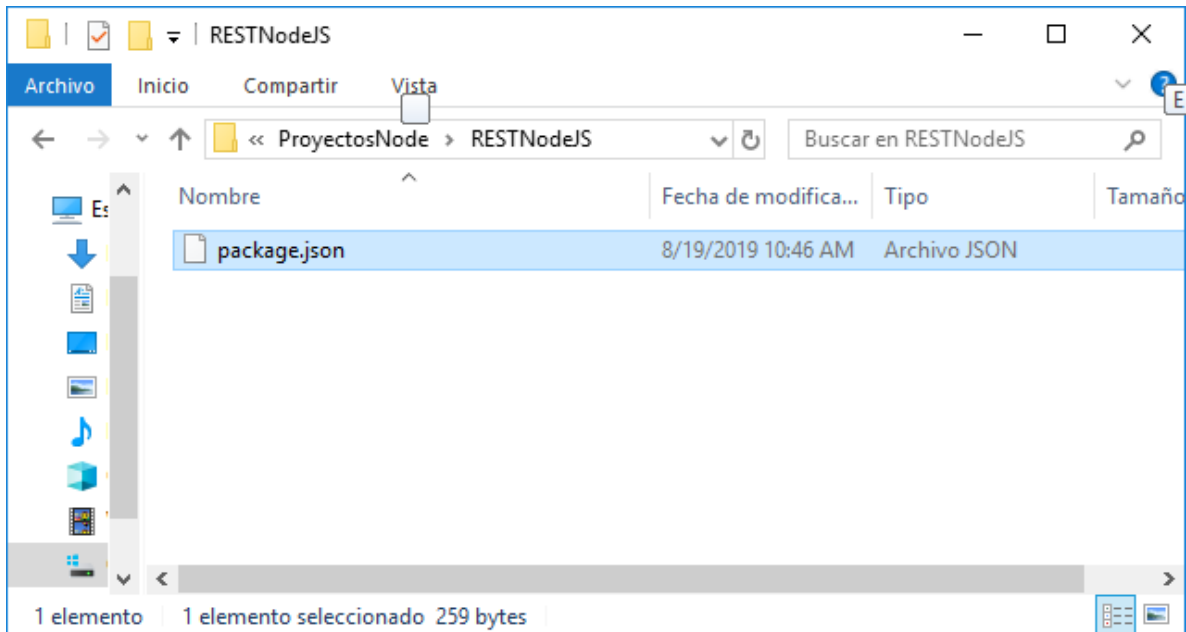
See `npm help json` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (restnodejs)
version: (1.0.0)
description: Servicios web REST con NodeJS
entry point: (index.js)
test command:
git repository:
keywords:
author: Jorge Miramontes Amezcua
license: (ISC)
About to write to C:\JMA\Personal\ITESM\SemestreAgostoDiciembre2019\ProyectosNode\RESTNodeJS\package.json:
{
  "name": "restnodejs",
  "version": "1.0.0",
  "description": "Servicios web REST con NodeJS",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "Jorge Miramontes Amezcua",
  "license": "ISC"
}

Is this OK? (yes) yes
C:\JMA\Personal\ITESM\SemestreAgostoDiciembre2019\ProyectosNode\RESTNodeJS>
```

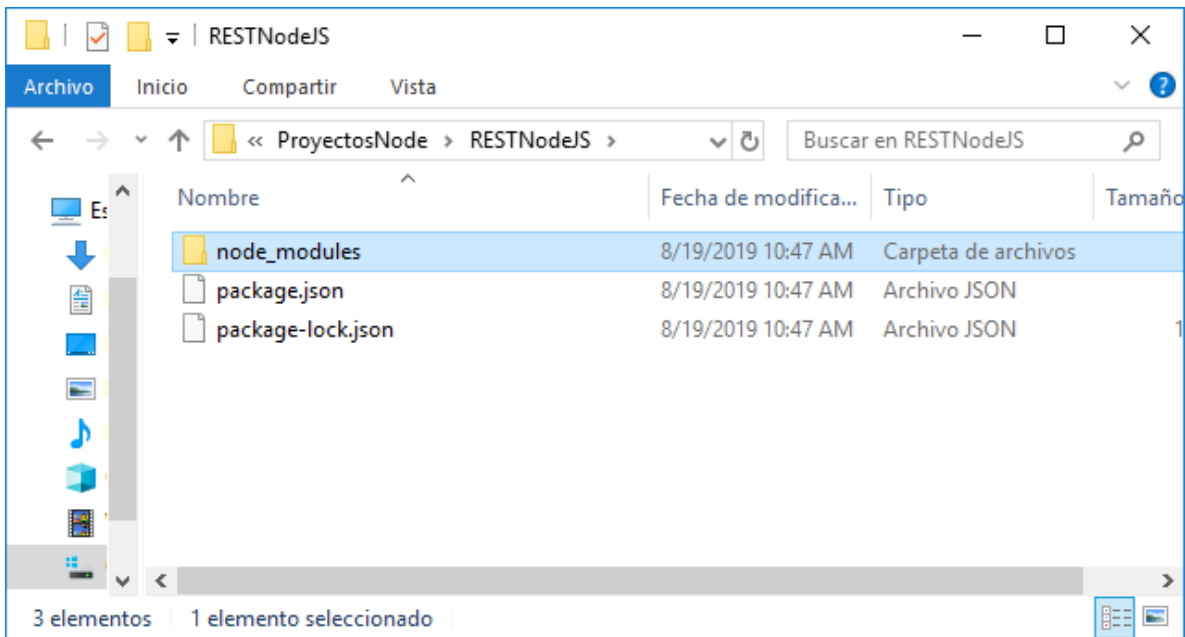
4. Dentro de la carpeta se genera el archivo package.json



5. Abrir una ventana de comandos, ubicarse dentro de la carpeta generada para instalar *express* con el siguiente comando:

```
$ npm install express --save
```

6. Se genera la carpeta “node_modules” y el archivo “package-lock.json”, el cual define las referencias a todas las dependencias que se encuentran en la carpeta node_modules:



7. Base de datos JSON con la información de los juegos: Dentro de la carpeta de nuestro proyecto (RESTNodeJS en éste caso), crear un archivo “juegos.json” y colocar la siguiente información:

```
{
  "tlou2" : {
    "nombre" : "The Last Of Us 2",
    "plataforma" : "Playstation 4/5",
    "developer" : "Naughty Dog",
    "lanzamiento": "TBD"
  },
  "smm2" : {
    "nombre" : "Super Mario Maker 2",
    "plataforma" : "Switch",
    "developer" : "Nintendo",
    "lanzamiento": "28-06-2019"
  },
  "avengers" : {
    "nombre" : "Marvel's Avengers",
```

```
"plataforma" : "Multiplataforma",  
"developer" : "Square Enix",  
"lanzamiento": "15-05-2020"  
}  
}
```

8. Dentro de la carpeta de nuestro proyecto, crear un archivo llamado “gameCollectionRoutes.js” y colocar el siguiente contenido:

```
//https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Strict_mode  
'use strict';  
  
/*  
 * module.exports existe por default en cada archivo JS de Node.  
 * module es una variable que representa un módulo, y exports es un objeto que  
 * expone dicho módulo.  
 */  
module.exports = function(app) {  
  var gameCollection = require('./gameCollectionController');  
  
  app.route('/games')  
    .get(gameCollection.obtener_juegos)  
    .post(gameCollection.agregar_juego);  
  
  app.route('/games/:gameId')  
    .get(gameCollection.obtener_juego);  
};
```

9. Abrir una ventana de comandos y ubicarse en la carpeta del proyecto. Instalar fs con el siguiente comando:

```
$ npm install fs --save
```

10. Dentro de la carpeta de nuestro proyecto, crear un archivo llamado “gameCollectionController.js” y colocar el siguiente contenido:

```
'use strict';  
  
var fs = require("fs");  
  
module.exports.obtener_juegos = function(req, res) {  
  fs.readFile( __dirname + "/" + "juegos.json", 'utf8', function (err, data) {  
    console.log( err );  
    console.log( data );  
    res.end( data );  
  });  
};
```

```
module.exports.agregar_juego = function(req, res) {
  fs.readFile( __dirname + "/" + "juegos.json", 'utf8', function (err, data) {
    data = JSON.parse( data );
    for(var key in req.body) {
      if(req.body.hasOwnProperty(key)){
        data[key] = req.body[key];
      }
    }
    console.log( err );
    console.log( data );
    var nuevo = JSON.stringify(data);

    fs.writeFile(__dirname + "/" + "juegos.json", nuevo, 'utf8', function(err, data){
      console.log( err );
      res.end( err );
    });

    res.end( JSON.stringify(data));
  });
};

module.exports.obtener_juego = function(req, res) {
  fs.readFile( __dirname + "/" + "juegos.json", 'utf8', function (err, data) {
    var juegos = JSON.parse( data );
    var juego = juegos[req.params.gameld]
    console.log( juego );
    res.end( JSON.stringify(juego));
  });
};
```

11. Abrir una ventana de comandos y ubicarse en la carpeta del proyecto. Instalar body-parser con el siguiente comando:

```
$npm install body-parser --save
```

12. Crear un archivo llamado “server.js” y colocar el siguiente contenido:

```
// https://expressjs.com/en/guide/using-middleware.html
var express = require('express'),
    app = express(),
    port = process.env.PORT || 8585,
    bodyParser = require('body-parser');

app.use(bodyParser.urlencoded({ extended: true }));
```

```
app.use(bodyParser.json());

var routes = require('./gameCollectionRoutes');
routes(app);

app.listen(port);

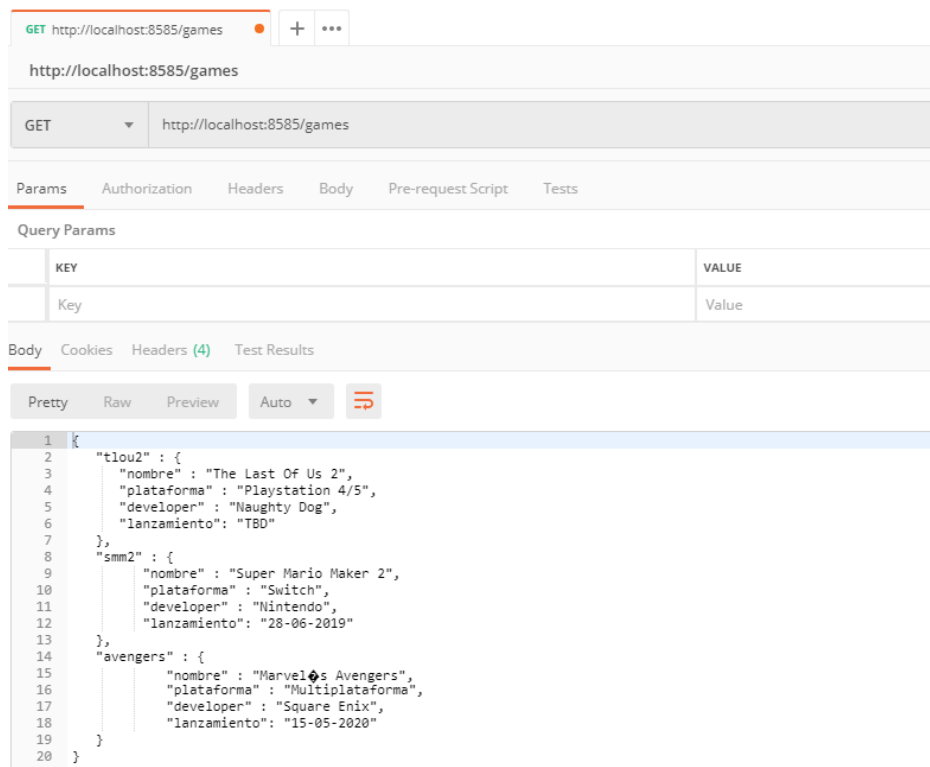
console.log('Servidor escuchando en puerto: ' + port);
```

13. Abrir una ventana de comandos y ubicarse en la carpeta del proyecto. Ejecutar server.js con el siguiente comando:

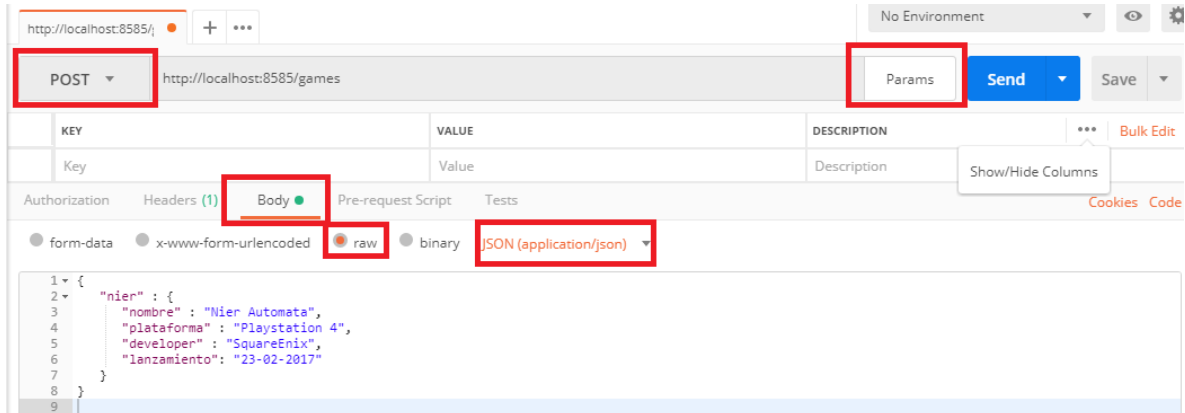
```
$ node server.js
```

```
C:\JMA\Personal\ITESM\SemestreEneroMayo2019\ProyectosNode\RESTNodeJS>node server.js
Servidor escuchando en puerto: 8585
```

14. Descarga e instala la herramienta “Postman”: <https://www.getpostman.com/>
15. Prueba GET para listar todos los juegos: En Postman, elija la operación GET y utilice la URL <http://localhost:8585/games> y haga clic en el botón Send:



16. Prueba POST para insertar un juego: En Postman, elija la operación POST y utilice la URL <http://localhost:8585/games>. Seleccione el tab “Body”, después la opción “raw”, y finalmente indique que el contenido es de tipo JSON. Agregue como key la clave del juego y como value un objeto JSON con los datos del juego. Haga clic en Send:



```
{
  "nier": {
    "nombre": "Nier Automata",
    "plataforma": "Playstation 4",
    "developer": "SquareEnix",
    "lanzamiento": "23-02-2017"
  }
}
```

El resultado se mostrará en la respuesta de la invocación.