



# MEMORIA PROYECTO FINAL DE TRIMESTRE

## Descripción breve

Este es el documento de memoria técnica para el proyecto final del primer trimestre correspondiente al módulo Despliegue de Aplicaciones Web

**Iván Hernández Fuentes**

**Profesor: Jesús Arribi Vilela**

# Resumen

El término REST ha ido extendiéndose a lo largo de los años siendo popularmente conocido en las API REST, REST ha ido adaptándose cada vez más por parte de los desarrolladores de servicios Web, que buscan la escalabilidad y simplicidad.

El protocolo **REST (Representational state transfer) o Transferencia de estado representacional** es un protocolo de comunicación que se caracteriza por establecer las comunicaciones sin estado (**Stateless**), esta característica es la que nos permite que los mensajes **HTTP** sean **más simples** que los utilizados en otros protocolos como **SOAP (Simple Object Access Protocol)** en donde se deben incluir información de la sesión como es la autenticación o documentos XML para definir las operaciones posibles entre el cliente y el Servidor.

# Índice

## Tabla de ilustraciones

<b>RESUMEN .....</b>	<b>2</b>
<b>ÍNDICE .....</b>	<b>3</b>
<b>1 INTRODUCCIÓN .....</b>	<b>4</b>
<b>2 ANÁLISIS E IMPLEMENTACIÓN .....</b>	<b>5</b>
2.1 Estructura de la Aplicación .....	6
2.2 Módulos.....	6
2.3 Diagrama de Gantt .....	6
2.4 End Points y Operaciones.....	7-12
2.4.1 <b>POST</b> .....	7
2.4.2 <b>GET</b> .....	8-9
2.4.3 <b>PUT</b> .....	10-11
2.4.4 <b>DELETE</b> .....	12
<b>3 FORMULARIOS.....</b>	<b>13-18</b>
<b>5 PROBLEMAS ENCONTRADOS .....</b>	<b>19</b>
<b>6 CONCLUSIONES Y FUTUROS TRABAJOS A DESARROLLAR .....</b>	<b>13</b>

# Introducción

La idea de este proyecto es realizar una **API REST CRUD con nodeJS, MongoDB y Mongoose** poniendo en práctica los conocimientos del alumno a lo largo del trimestre.

La idea que vamos a desarrollar es una aplicación para la gestión de un Menú en un restaurante, para ello implementaremos todas las operaciones CRUD, ***crear, leer, actualizar y eliminar.***

# Análisis e implementación

Para elaborar la aplicación partiremos de la siguiente estructura, dónde tenemos los siguientes objetos.

```
▼ object {4}
  ► dishes [4]
  ► promotions [1]
  ► leaders [4]
  ► feedback [0]
```

En el anterior JSON cabe destacar que en el objeto **“Dishes”** trabajaríamos con un segundo nivel que serían los comentarios, donde comentarios es un array.

```
▼ object {4}
  ▼ dishes [4]
    ▼ 0 {8}
      name : Uthappizza
      image : images/uthappizza.png
      category : mains
      label : Hot
      price : 4.99
      featured : true
      description : A unique combination of Indian
                    Uthappam (pancake) and Italian pizza,
                    topped with Cerignola olives, ripe
                    vine cherry tomatoes, Vidalia onion,
                    Guntur chillies and Buffalo Paneer.
      ► comments [5]
```



## Estructura de la aplicación

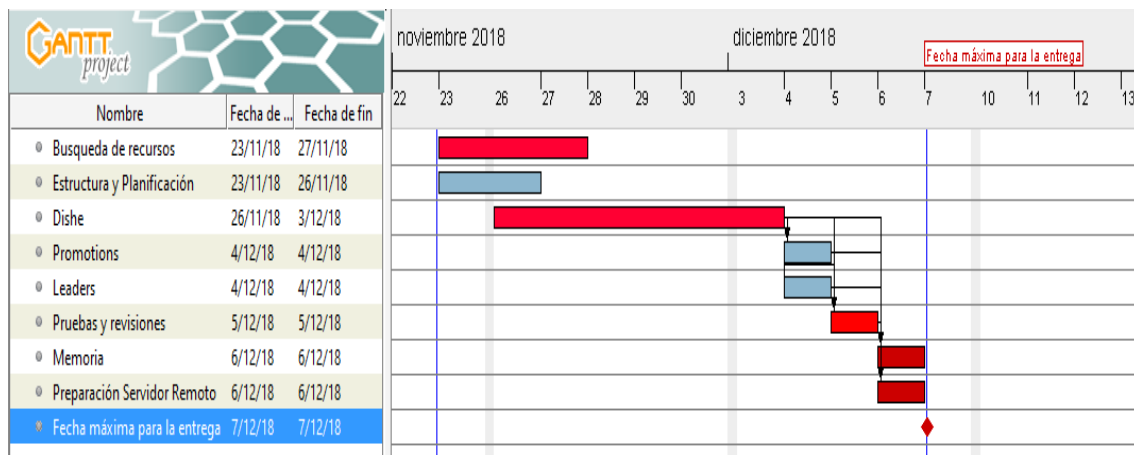
La estructura de nuestra aplicación se ha generado en base al **“Express application Generator”** del framework **express**.

## Módulos

En el desarrollo de la API REST hemos utilizado los siguientes módulos.

- Body-parser
- Debug
- Express
- http-errors
- mongoose
- morgan
- nodemon
- pug

## Diagrama de Gantt

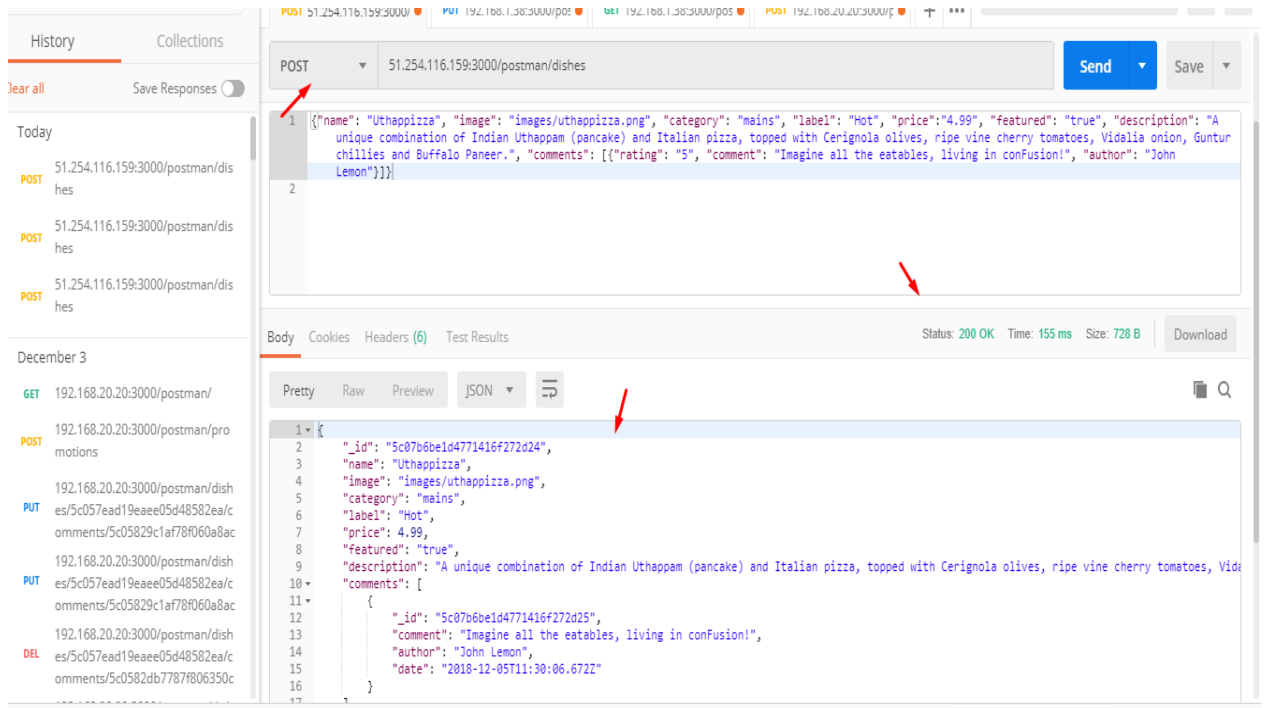


## End Points y Operaciones

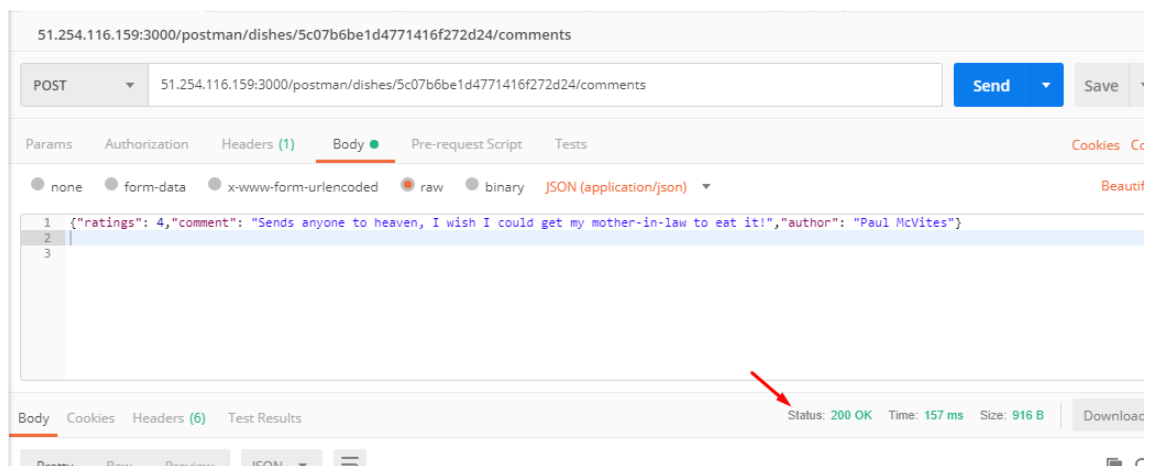
La aplicación que en mi caso he desarrollado, e **implementado formularios** para que sea mas visual al usuario, y se le de aún una mayor utilidad a la práctica. Aunque también se puede trabajar desde **POSTMAN** con el objeto **“Dishe”**

### POST

Para Insertar un dishe accederíamos a `/postman/dishes`



Ahora añadiremos a ese plato un comentario más, para ello deberemos de acceder a `/postman/dishes/id_dishe/comments`



Observamos como se añadió el comentario

```
{
  "_id": "5c07b6bed4771416f272d24",
  "name": "Uthappizza",
  "image": "images/uthappizza.png",
  "category": "mains",
  "label": "Hot",
  "price": 4.99,
  "featured": "true",
  "description": "A unique combination of Indian Uthappam (pancake) and Italian pizza, topped with Cerignola olives, ripe vine cherry tomatoes, Vid",
  "comments": [
    {
      "date": "2018-12-05T11:30:06.672Z",
      "_id": "5c07b6bed4771416f272d25",
      "comment": "Imagine all the eatables, living in confusion!",
      "author": "John Lemon"
    },
    {
      "_id": "5c07bc01718462423ea25493",
      "ratings": 4,
      "comment": "Sends anyone to heaven, I wish I could get my mother-in-law to eat it!",
      "author": "Paul McVites",
      "date": "2018-12-05T11:52:33.588Z"
    }
  ],
  "_v": 2
}
```

## GET

En cuanto al método GET tenemos varios ejemplos, como buscar un “dishe” según una categoría. Accediendo a ***/postman/dishes/category/:nombre\_categoria***

GET 51.254.116.159:3000/postman/dishes/category/mains Send Save

Body Cookies Headers (6) Test Results Status: 200 OK Time: 69 ms Size: 918 B Download

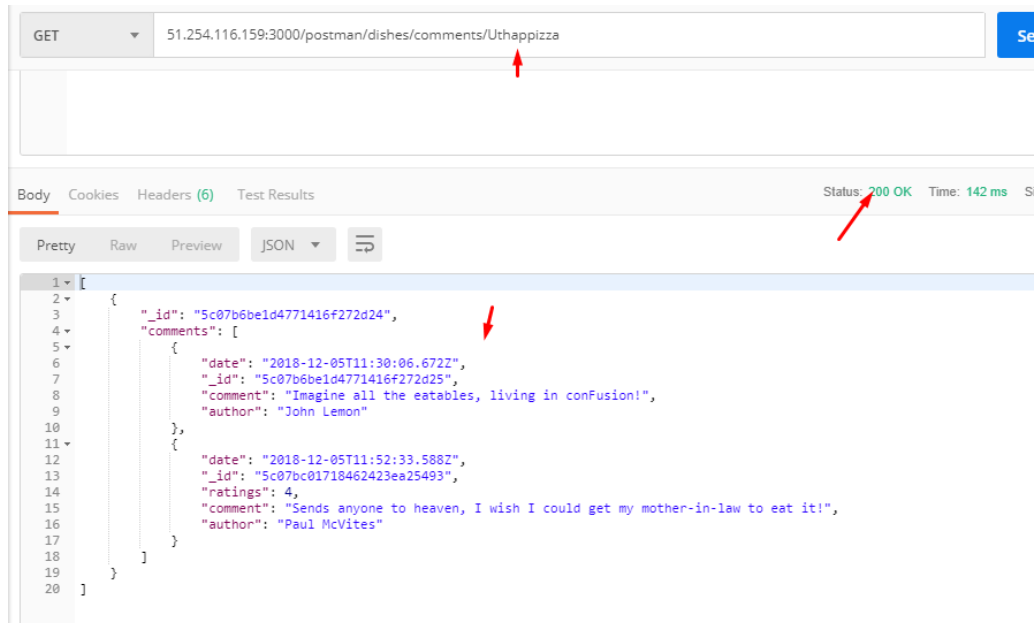
Pretty Raw Preview JSON

```
1 [
2   {
3     "_id": "5c07b6bed4771416f272d24",
4     "name": "Uthappizza",
5     "image": "images/uthappizza.png",
6     "category": "mains",
7     "label": "Hot",
8     "price": 4.99,
9     "featured": "true",
10    "description": "A unique combination of Indian Uthappam (pancake) and Italian pizza, topped with Cerignola olives, ripe vine cherry tomato",
11    "comments": [
12      {
13        "date": "2018-12-05T11:30:06.672Z",
14        "_id": "5c07b6bed4771416f272d25",
15        "comment": "Imagine all the eatables, living in confusion!",
16        "author": "John Lemon"
17      },
18      {
19        "date": "2018-12-05T11:52:33.588Z",
20        "_id": "5c07bc01718462423ea25493",
21        "ratings": 4,
22        "comment": "Sends anyone to heaven, I wish I could get my mother-in-law to eat it!",
23        "author": "Paul McVites"
24      }
25    ],
26    "_v": 2
27  }
28 ]
```



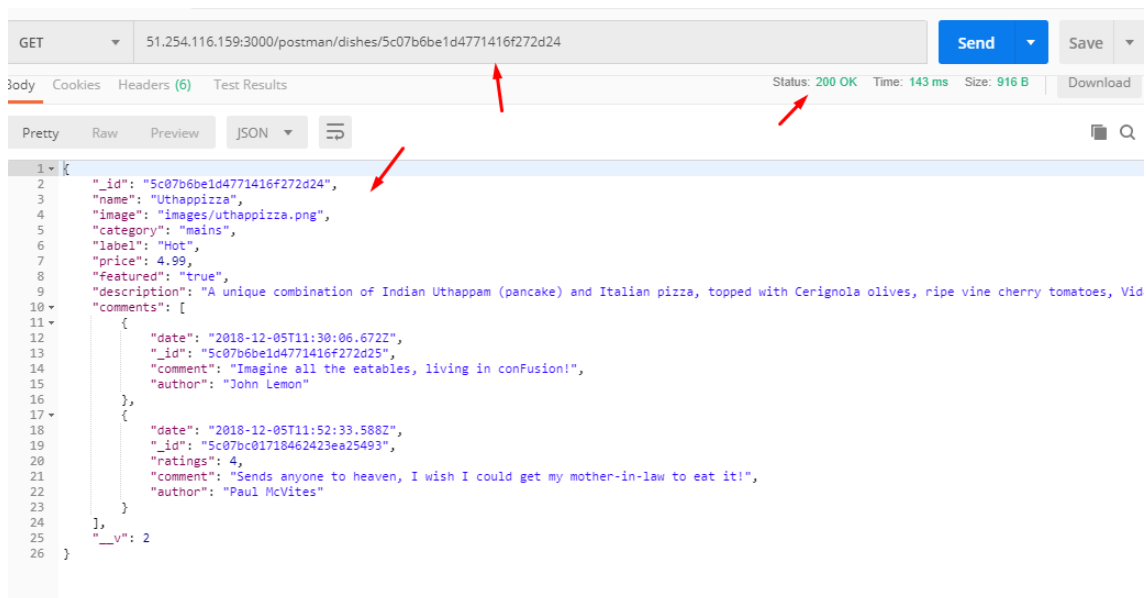
También mediante GET podemos acceder a los comentarios de un plato en concreto.

Accederíamos mediante `/postman/dishes/comments/:nombre_dishe`



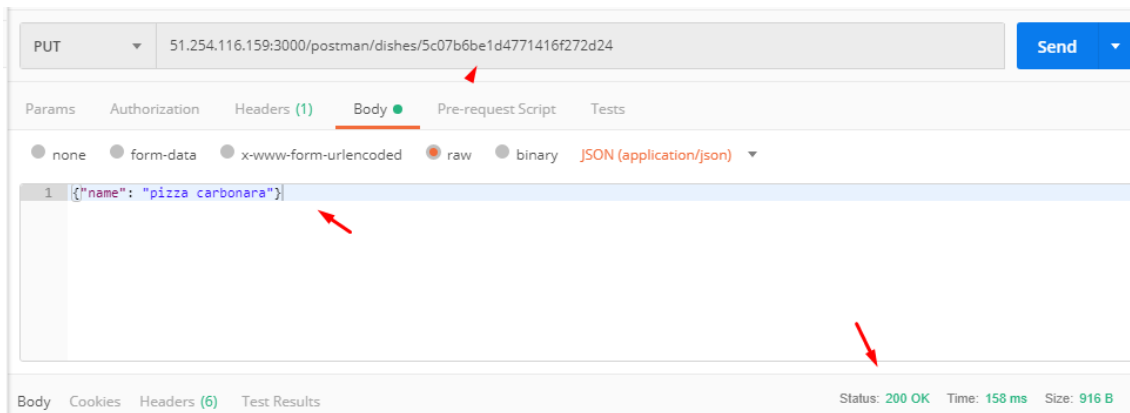
Por último, tenemos un GET que permite **buscar toda la información de un “Dishe” mediante su “id”**

Accederíamos con `/postman/dishes/:id_dishe`

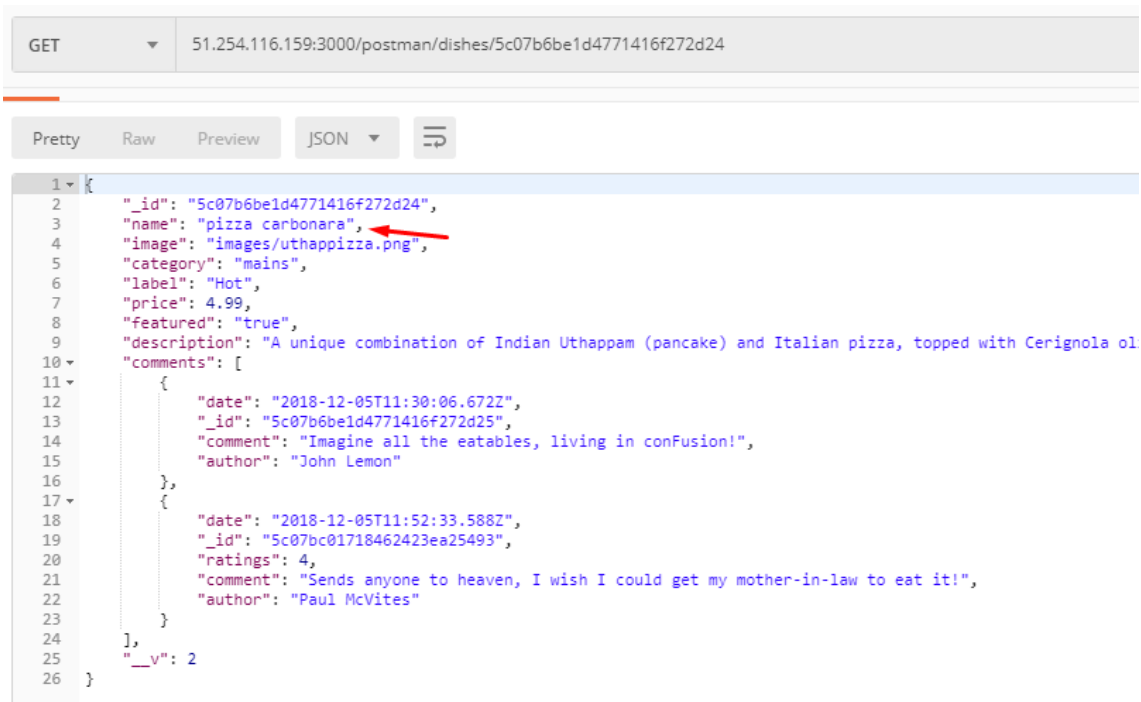


## PUT

Con PUT podemos actualizar un plato. (En este caso vamos actualizar el nombre del plato)

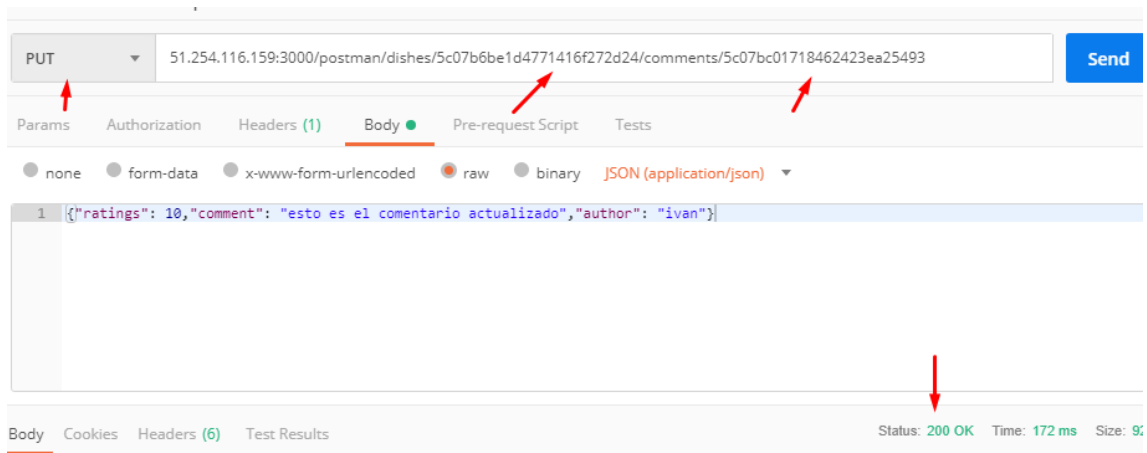


Volvemos a realizar un GET al plato, y observamos que se actualizó correctamente.



El siguiente **PUT** será **para actualizar un comentario** del plato.

Accederíamos mediante **/postman/dishes/:id/comments/:id\_comment**



Comprobamos como el comentario se actualizo correctamente.

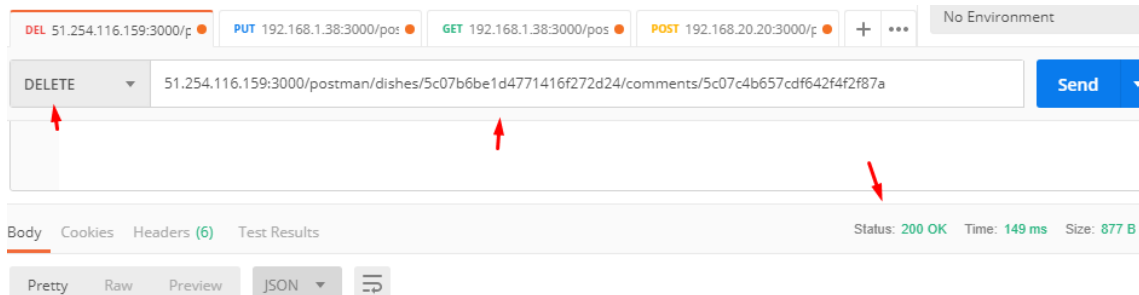


## DELETE

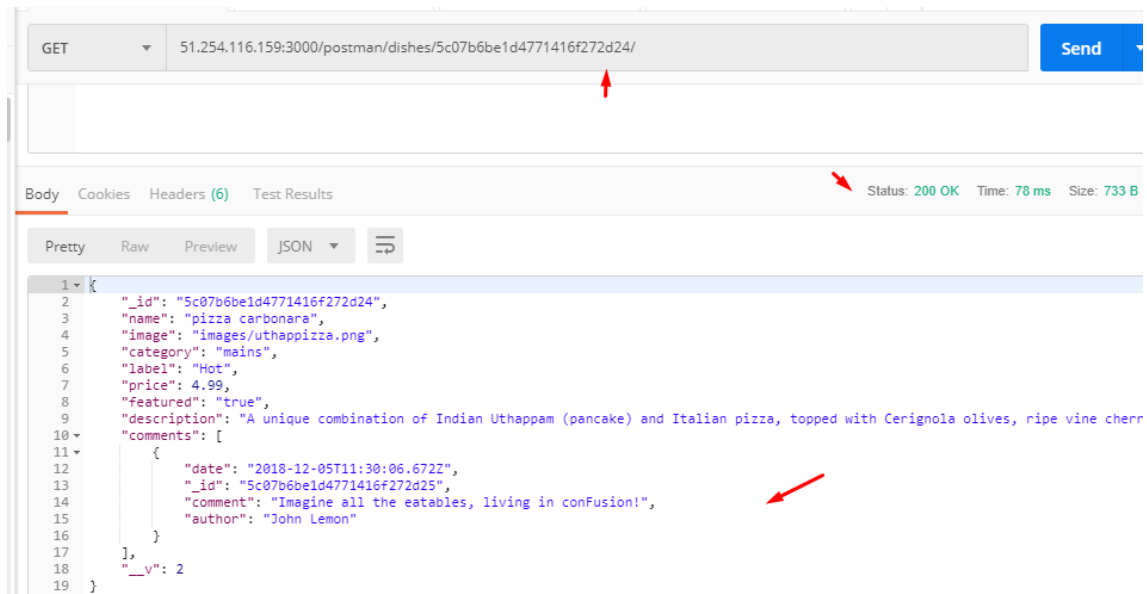
Por último, queda mostrar el método **“Delete”**, lo utilizaremos para borrar un comentario.

Accederíamos mediante ***/postman/dishes/:id\_dishe/comments/:id\_comment***

Eliminamos el segundo Comentario



Realizamos el GET y el comentario ya no existe



# Formularios

Como explicaba anteriormente el principal desarrollo de la aplicación ha sido mediante la implementación de formularios. *(Solo se describirá el objeto `dishes`)*

Tenemos la vista de los platos donde mostramos todos los platos almacenados en la BD y un formulario para añadir uno nuevo, a continuación añadimos un nuevo plato.

The screenshot shows a web browser at the URL `51.254.116.159:3000/dishes/view`. The page title is "Gestión de los Platos". Below the title, there is a "Filtrar por Categoría" section with three buttons: "Appetizer", "Mains", and "Dessert". The "Appetizer" button is highlighted with a red arrow. Below the filters is a table of dishes:

Nombre	Categoría	Precio	Reseñas	Editar	Eliminar
pizza carbonara	mains	4.99	<a href="#">Ver</a>	<a href="#">Editar</a>	<a href="#">Eliminar</a>
Zucchipakoda	appetizer	1.99	<a href="#">Ver</a>	<a href="#">Editar</a>	<a href="#">Eliminar</a>

A red arrow points to the "Zucchipakoda" row. To the right of the table is a form titled "Añadir un n" (partially visible) with a "Nombre" label and a text input field containing the word "titulo".

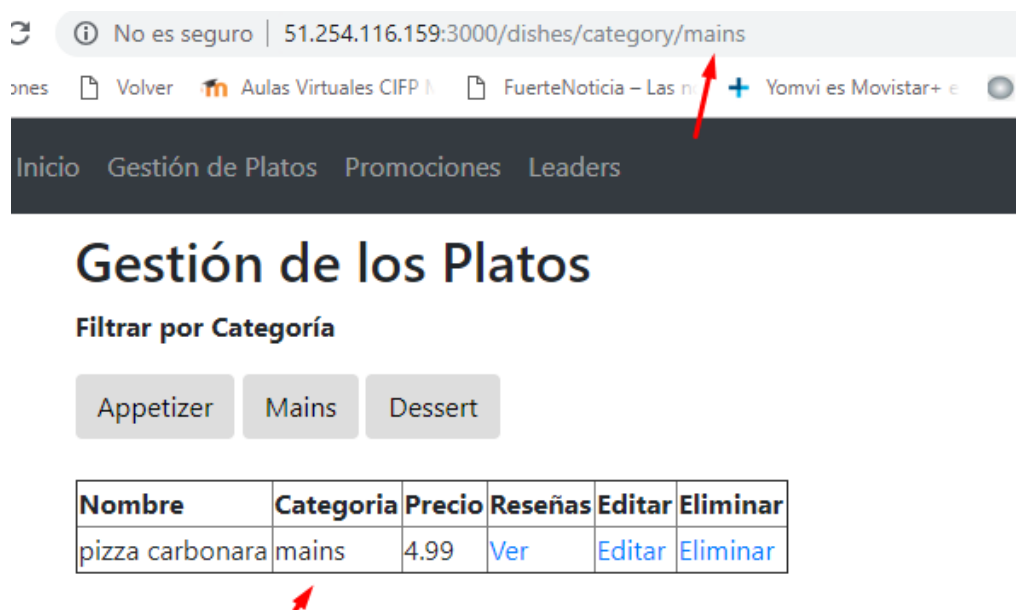
Podemos también filtrar según su categoría en este caso probamos con *(Appetizer)*

The screenshot shows the same web application but with the URL `51.254.116.159:3000/dishes/category/appetizer`. The "Appetizer" button in the "Filtrar por Categoría" section is now highlighted with a red arrow. The table of dishes is filtered to show only the "Zucchipakoda" dish:

Nombre	Categoría	Precio	Reseñas	Editar	Eliminar
Zucchipakoda	appetizer	1.99	<a href="#">Ver</a>	<a href="#">Editar</a>	<a href="#">Eliminar</a>

A red arrow points to the "Zucchipakoda" row. The "Añadir un n" form is still visible on the right.

## Filtramos por Categoría “Mains”



Inicio Gestión de Platos Promociones Leaders

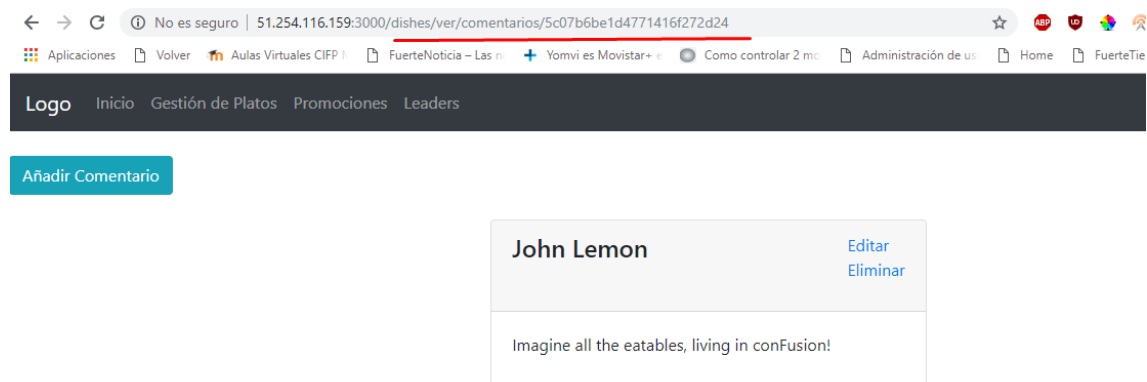
### Gestión de los Platos

Filtrar por Categoría

Appetizer Mains Dessert

Nombre	Categoría	Precio	Reseñas	Editar	Eliminar
pizza carbonara	mains	4.99	Ver	Editar	Eliminar

En la tabla nos salen varias opciones, entre ellas “Ver” para visualizar los comentarios, cuando accedemos vemos en la url que carga “/ver/comentarios/:id\_dishe”.



← → ↻ No es seguro | 51.254.116.159:3000/dishes/ver/comentarios/5c07b6be1d4771416f272d24

Aplicaciones Volver Aulas Virtuales CIPP FuerteNoticia – Las n Yomvi es Movistar+ e Como controlar 2 mo Administración de us Home FuerteTie

Logo Inicio Gestión de Platos Promociones Leaders

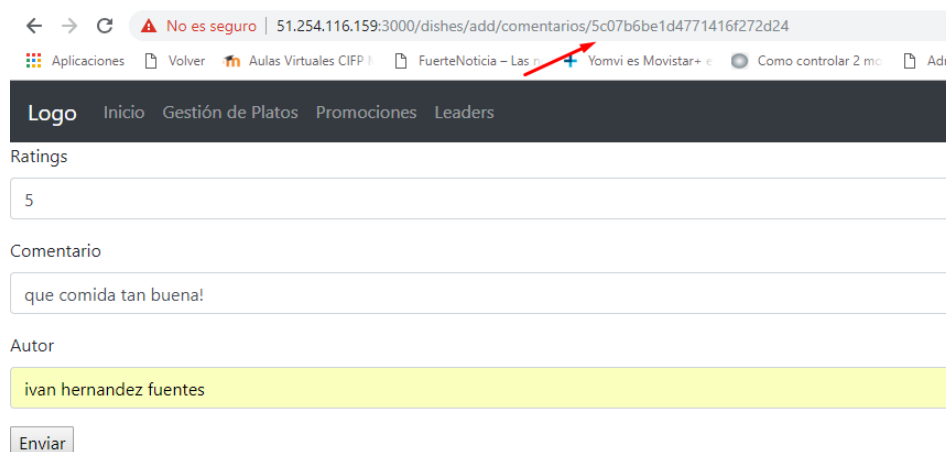
Añadir Comentario

John Lemon

Editar  
Eliminar

Imagine all the eatables, living in conFusion!

## Añadimos un nuevo comentario



← → ↻ No es seguro | 51.254.116.159:3000/dishes/add/comentarios/5c07b6be1d4771416f272d24

Aplicaciones Volver Aulas Virtuales CIPP FuerteNoticia – Las n Yomvi es Movistar+ e Como controlar 2 mo Administración de us

Logo Inicio Gestión de Platos Promociones Leaders

Ratings

5

Comentario

que comida tan buena!

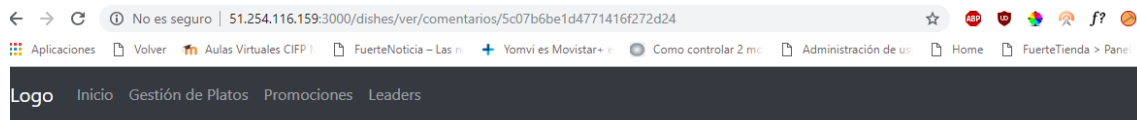
Autor

ivan hernandez fuentes

Enviar

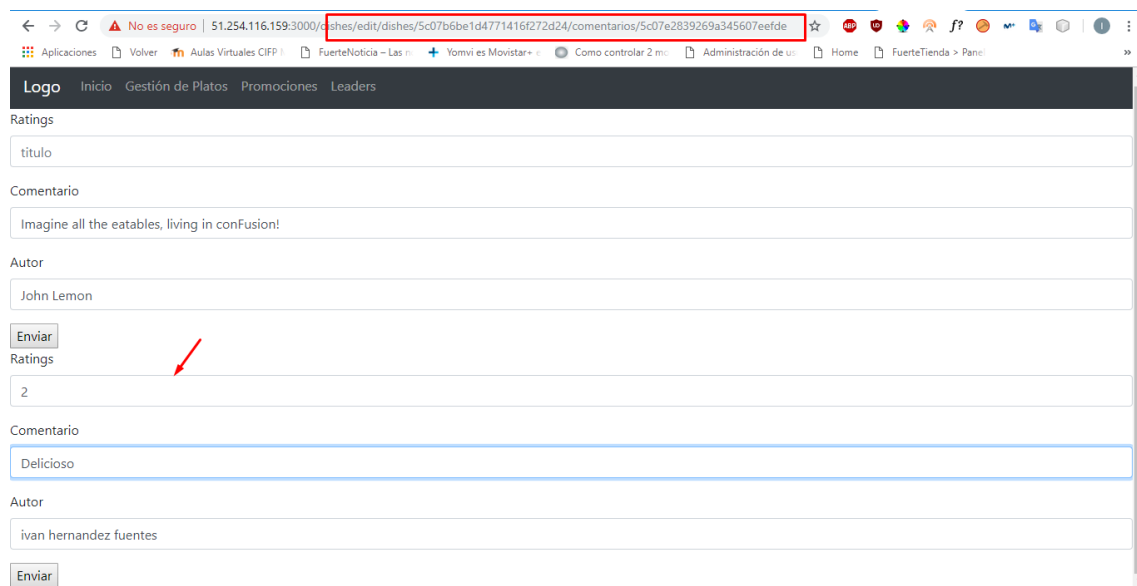
Al añadir un nuevo comentario nos sale en la url el **id del plato** y abajo visualizamos el formulario

## Volvemos a ver los comentarios y ya nos aparece el introducido anteriormente



En la misma página también podemos realizar la operación PUT que anteriormente vimos con POSTMAN.

La url se hace mas larga en este caso ya que necesitamos tanto el id del plato y el id del comentario.



Enviamos el formulario y observamos que se actualizo correctamente.

John Lemon

[Editar](#)  
[Eliminar](#)

Imagine all the eatables, living in conFusion!

ivan hernandez fuentes

☆☆

[Editar](#)  
[Eliminar](#)

Delicioso

Por último, podemos eliminar un comentario para ello en la url tiene que cargar los siguientes datos.

← → ↻ ⓘ No es seguro | 51.254.116.159:3000/dishes/ver/comentarios/5c07b6be1d4771416f272d24 ☆

Aplicaciones Volver Aulas Virtuales CIFP FuerteNoticia – Las n + Yomvi es Movistar+ e Como controlar 2 m Administración de us

Logo Inicio Gestión de Platos Promociones Leaders

Añadir Comentario

John Lemon

[Editar](#)  
[Eliminar](#)

Imagine all the eatables, living in conFusion!

ivan hernandez fuentes

☆☆

[Editar](#)  
[Eliminar](#)

Delicioso

1.254.116.159:3000/dishes/delete/dishes/5c07b6be1d4771416f272d24/comentarios/5c07e3d59269a345607eefdf

“/delete/dishes/:id\_dishe/comentarios/:id\_comentario”



El comentario se elimina correctamente

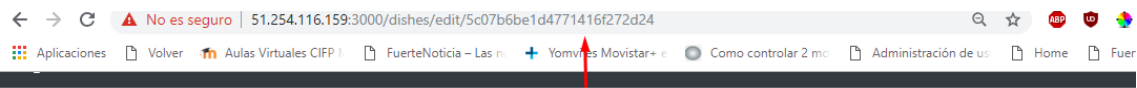
John Lemon

[Editar](#)  
[Eliminar](#)

Imagine all the eatables, living in conFusion!

Para la edición de un plato se basa en las mismas operaciones que los comentarios.

Editaremos el nombre del plato.



### Editar Plato

Nombre

Imagen

Categoría

Label

Precio

Featured

A unique combination of

Enviar

Cargamos la vista de los platos

### Gestión de los Platos

Filtrar por Categoría

[Appetizer](#) [Mains](#) [Dessert](#)

Nombre	Categoría	Precio	Reseñas	Editar	Eliminar
pizza mozzarella	mains	4.99	<a href="#">Ver</a>	<a href="#">Editar</a>	<a href="#">Eliminar</a>
Zucchipakoda	appetizer	1.99	<a href="#">Ver</a>	<a href="#">Editar</a>	<a href="#">Eliminar</a>

Por último, eliminamos el plato.

Nombre	Categoría	Precio	Reseñas	Editar	Eliminar
pizza mozzarella	mains	4.99	<a href="#">Ver</a>	<a href="#">Editar</a>	<a href="#">Eliminar</a>
Zucchipakoda	appetizer	1.99	<a href="#">Ver</a>	<a href="#">Editar</a>	<a href="#">Eliminar</a>

51.254.116.159:3000/dishes/delete/5c07b6be1d4771416f272d24

La operación se ha realizado correctamente

No es seguro | 51.254.116.159:3000/dishes

AplicacionesVolverAulas Virtuales CIPPFuerteNoticia – Las n+ Yomvi es Movis

LogoInicioGestión de PlatosPromocionesLeaders

# Gestión de los Platos

Filtrar por Categoría

Appetizer

Mains

Dessert

Nombre	Categoría	Precio	Reseñas	Editar	Eliminar
Zucchipakoda	appetizer	1.99	<a href="#">Ver</a>	<a href="#">Editar</a>	<a href="#">Eliminar</a>

# Problemas encontrados

Desarrollando la aplicación me encontré con algún que otro problema, uno de los problemas **“más importantes”** era al editar un comentario, puesto que si nos fijamos en las anteriores imágenes al editar un comentario salen todos los demás comentarios de ese plato. La verdad no entiendo el problema ya que la consulta que se realiza es un **“findOneAndUpdate()”** con argumentos como **id\_dishe** y **el id del comentario**. Aunque en la url se muestran el **“id”** correcto muestra todos los comentarios del plato.

Otro problema encontrado fue en los **“textarea”** a la hora de editar un plato, o una promoción como poder mostrar la información y asignarle el atributo **“name”** al mismo tiempo **en pug**.

Ejemplo del textarea definido.

**textarea=dishe.description**

# Conclusiones y futuros trabajos a desarrollar

La realización de este proyecto me ha dado la oportunidad de adquirir y aprender muchos conocimientos en cuanto a las **API REST** y también aprender el funcionamiento de **NodeJS (JavaScript en el entorno Servidor)**, así como el manejo y la utilización de **bases de datos NoSQL**, y **persistencia** en los datos con **Mongoose**.

Los futuros trabajos que puede optar la aplicación son muchos, entre ellos **arreglar el problema al editar un comentario** *(que no salgan todos los de ese plato)*.

**Desarrollar la aplicación basándose en el MVC (Modelo Vista Controlador).**

# Acceso al Proyecto

Para poder acceder al proyecto desarrollado se proporcionan los siguientes enlaces:

- Código en GitHub
  - [https://github.com/iherfue/DPL-2018/tree/master/proyecto\\_final\\_primer\\_trimestre\\_api\\_rest\\_crud\\_iva\\_n/proyecto\\_dpl/myapp](https://github.com/iherfue/DPL-2018/tree/master/proyecto_final_primer_trimestre_api_rest_crud_iva_n/proyecto_dpl/myapp)
- Servidor Web
  - <http://51.254.116.159:3000/>