

Asignatura	Datos del alumno	Fecha
Ingeniería y Desarrollo en la Web	Apellidos: Quintero Gutiérrez	08/03/2019
	Nombre: Claudia Lorena	

Actividades

Implementación de una solución basada en una aplicación web II

Manual del Usuario

Esta aplicación permite realizar una lista de blog y de usuarios que se encuentran en el sistema

Primero se realiza un fichero serviWeb.js que contiene las peticiones al servidor web. Contiene la referencia al framework de express, body-parser, la base de datos MongoDB la cual está en mat lab, y todo lo relacionado con el intercambio de datos el cual se va a realizar en formato JSON. Por otra parte el servicio funcionará por el puerto 8080.

```
var express = require('express');
var app = express();
var bodyParser = require('body-parser');
var MongoClient = require('mongodb').MongoClient;
//Enviar datos a través de formularios
//Meter en la aplicación el parseador de url
app.use(bodyParser.urlencoded({extended:true}));

//Enviar datos a través de servicios web en json
app.use(bodyParser.json());

app.listen(8080,function(){
  console.log('Funcionando en el puerto 8080');
})
```

Asignatura	Datos del alumno	Fecha
Ingeniería y Desarrollo en la Web	Apellidos: Quintero Gutiérrez	08/03/2019
	Nombre: Claudia Lorena	

GET

Obtener todas las entradas del blog a través de la siguiente url:

GET
http://localhost:8080
Params
Send

```
//Devolver todos los entradas
```

```
Complexity is 3 Everything is cool!
```

```
app.get("/",function(req,res){
```

```

    // //Para que devuelva un Json le damos stringify(entradas);
    // res.send(JSON.stringify(anuncios));
    MongoClient.connect('mongodb://masterdb:GabSant2@ds225375.mlab.com:25375/blogtec',
    function(err, db) {
        console.log("Conectado al servidor")

        var collection = db.collection('entradas');
        collection.find({}).toArray(function (err, result) {
            res.status(200);
            res.json(result);

            // Cerrar el cliente
            db.close();
        });
    });
});

```

En postman podemos visualizar lo siguiente

http://localhost:8080/
No Environment
GET
http://localhost:8080/
Params
Send
Save

Body
Cookies (1)
Headers (6)
Test Results
Status: 200 OK Time: 1378 ms Size: 1.54 KB

Pretty
Raw
Preview
JSON

```

1 [
2   {
3     "_id": "5c5e314be0e40aae0534a6d8",
4     "autor": "franck",
5     "fecha": "2019-02-13",
6     "tema": "node js"
7   },
8   {
9     "_id": "5c5e3165e0e40aae0534a6d9",
10    "autor": "javier",
11    "fecha": "2019-02-14",
12    "tema": "angular",
13    "comentarios": [
14      "este es el comentario",
15      "este es el segundo comentario",
16      "Este es el tercer comentario",
17      "Este es el cuarto comentario",
18      "Este es el quinto comentario",
19      "Este es el sexto comentario"
20    ]
21  },
22  {
23    "_id": "5c5e4c4258b47808088b24de",
24    "autor": "Gabriel",

```

Asignatura	Datos del alumno	Fecha
Ingeniería y Desarrollo en la Web	Apellidos: Quintero Gutiérrez	08/03/2019
	Nombre: Claudia Lorena	

Ahora obtenemos una entrada por id de la siguiente forma proporcionando en la url el id de la entrada a revisar:

GET
▼
http://localhost:8080/entrada/5c5e314be0e40aae0534a6d8
Params
Send
▼

```
//Devolver una entrada predeterminada
Complexity is 3 Everything is cool!
app.get("/entrada/:id", function(req, res){
  //recibimos el id
  //Transformar en un objeto id de mongo
  var idEntrada = require('mongodb').ObjectID(req.params.id);
  MongoClient.connect('mongodb://masterdb:GabSant2@ds225375.mlab.com:25375/blogtec',
  function(err, db) {
    console.log("Conectado al servidor")
    var collection = db.collection('entradas');
    collection.find({_id:idEntrada}).toArray(function (err, result) {
      res.status(200);
      res.json(result);
      // Cerrar el cliente
      db.close();
    });
  });
});
```

Y desde PostMan

http://localhost:8080/
+
...
No Environment
▼
👁
⚙

GET
▼
http://localhost:8080/entrada/5c5e314be0e40aae0534a6d8
Params
Send
▼
Save
▼

TYPE
Inherit auth from parent
▼
The authorization header will be automatically generated when you send the request. [Learn more about authorization](#)
This request is not inheriting any authorization helper at the moment. Save it in a collection to use the parent's authorization helper.

Body
Cookies (1)
Headers (6)
Test Results
Status: 200 OK
Time: 8720 ms
Size: 303 B

Pretty
Raw
Preview
JSON
▼
🔍

1 [
2 {
3 "_id": "5c5e314be0e40aae0534a6d8",
4 "autor": "franck",
5 "fecha": "2019-02-13",
6 "tema": "node js"
7 }
8]

Asignatura	Datos del alumno	Fecha
Ingeniería y Desarrollo en la Web	Apellidos: Quintero Gutiérrez	08/03/2019
	Nombre: Claudia Lorena	

PUT

La función de actualizar esta presente cuando deseamos agregarle comentarios a una entrada determinada, es un arreglo y puede tener varios arreglos. Para poder agregar comentarios es necesario que la entrada este guardada previamente. Podemos acceder con el verbo PUT suministrando el id del blog.

The screenshot shows a REST client interface. At the top, a PUT request is configured for the URL `http://localhost:8080/entrada/5c5e314be0e40aae0534a6d8`. The request body is set to raw JSON: `{ "comentarios": "Este es un comentario desde postman" }`. The response status is 200 OK. The response body, shown in JSON format, is: `{ "mensaje": "Modificado con éxito" }`.

Cuando volvemos a consultar

The screenshot shows a REST client interface. A GET request is configured for the same URL: `http://localhost:8080/entrada/5c5e314be0e40aae0534a6d8`. The response status is 200 OK. The response body, shown in JSON format, is: `{ "_id": "5c5e314be0e40aae0534a6d8", "autor": "franck", "fecha": "2019-02-13", "tema": "node js", "comentarios": ["Este es un comentario desde postman"] }`.

Asignatura	Datos del alumno	Fecha
Ingeniería y Desarrollo en la Web	Apellidos: Quintero Gutiérrez	08/03/2019
	Nombre: Claudia Lorena	

```
//No se puede ingresar un comentario si la entrada
//No existe
Complexity is 5 Everything is cool!
app.put("/entrada/:id",function(req,res){
  //recibimos el id
  //Transformar en un objeto id de mongo
  console.log(req.body.comentarios);
  var idEntrada = require('mongoose').ObjectId(req.params.id);
  MongoClient.connect('mongodb://masterdb:GabSant2@ds225375.mlab.com:25375/blogtec',
  Complexity is 4 Everything is cool!
  function(err, db) {
    if (err) {
      res.status(500);
      //Esto es para crear un json de error
      res.json({
        mensaje:"Error en la base de datos",
        insertado:false
      });
    } else {
      console.log("Conectado al servidor")
      var collection = db.collection('entradas');
      collection.update({_id:idEntrada}, {$addToSet:{"comentarios":req.body.comentarios}},function (err, result) {
        res.status(200);
        //res.json(result);
        res.json(
          {
            mensaje:"Modificado con exito"
          });
        // Cerrar el cliente
        db.close();
      });
    }
  });
});
});
```

DELETE

Para acceder se utiliza el verbo DELETE y el id del elemento al que se desea borrar

DELETE ▾

http://localhost:8080/entrada/5c81e7294dfc8585077c9dc9

Params

Send ▾

http://localhost:8080/ + ... No Environment ▾ 👁 ⚙

DELETE ▾

http://localhost:8080/entrada/5c81e7294dfc8585077c9dc9

Params

Send ▾

Save ▾

Pretty Raw Preview JSON ▾ ≡

```

1 {
2   "mensaje": "borrado con exito"
3 }
```

Asignatura	Datos del alumno	Fecha
Ingeniería y Desarrollo en la Web	Apellidos: Quintero Gutiérrez	08/03/2019
	Nombre: Claudia Lorena	

```
//Eliminar una entrada determinada
Complexity is 5 Everything is cool!
app.delete("/entrada/:id",function(req,res){
  //recibimos el id
  //Transformar en un objeto id de mongo
  var idEntrada = require('mongodb').ObjectId(req.params.id);
  MongoClient.connect('mongodb://masterdb:GabSant2@ds225375.mlab.com:25375/blogtec',
    function(err, db) {
      console.log("Conectado al servidor")
      var collection = db.collection('entradas');
      Complexity is 3 Everything is cool!
      collection.remove({_id:idEntrada},function (err, result) {
        if (err) {
          res.status(500);
          //Esto es para crear un json de error
          res.json({
            mensaje:"Error borrando en la base de datos",
            insertado:false
          });
        } else {
          res.status(200);
          res.json({
            mensaje:"borrado con exito"
          });
          // Cerrar el cliente
        }
      });
      db.close();
    });
  });
  Complexity is 7 We time to do something
```

POST

Guardar una entrada, a partir del verbo POST y la siguiente ruta

POST ▾	http://localhost:8080/entrada	Params	Send ▾
--------	-------------------------------	--------	--------

Asignatura	Datos del alumno	Fecha
Ingeniería y Desarrollo en la Web	Apellidos: Quintero Gutiérrez	08/03/2019
	Nombre: Claudia Lorena	

http://localhost:8080/ + ...

POST http://localhost:8080/entrada Params Send Save

Authorization Headers (1) Body Pre-request Script Tests Cookies Code

form-data x-www-form-urlencoded raw binary JSON (application/json)

```
1 {"autor": "Jordan", "fecha": "2019-02-19", "tema": "Rest"}
```

Body Cookies (1) Headers (6) Test Results Status: 201 Created Time: 913 ms Size: 235 B

Pretty Raw Preview JSON

```
1 {
2   "insertado": true
3 }
```

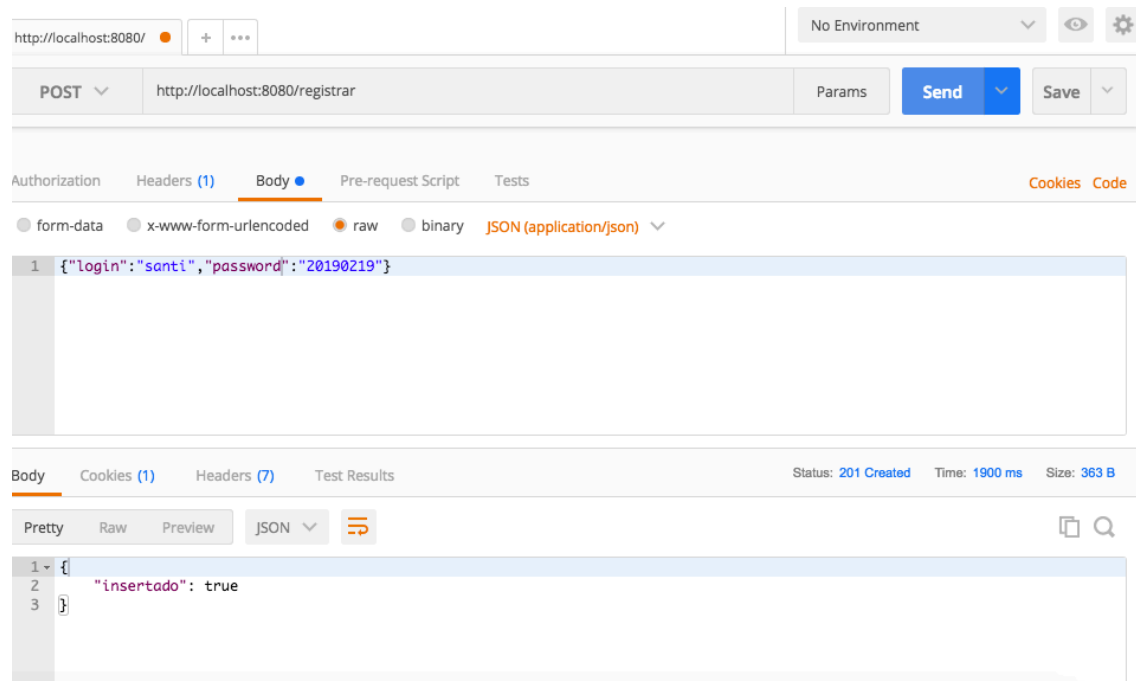
Complexity is 7 It's time to do something...

```
app.post("/entrada", function(req, res){
  MongoClient.connect('mongodb://masterdb:GabSant2@ds225375.mlab.com:25375/blogtec',
    function(err, db){
      if (err) {
        res.status(500);
        //Esto es para crear un json de error
        res.json({
          mensaje: "Error en la base de datos",
          insertado: false
        });
      } else {
        console.log('Conectado al servidor');
        var entrada = {
          autor: req.body.autor,
          fecha: req.body.fecha,
          tema: req.body.tema
        };
        var collection = db.collection('entradas');
        Complexity is 3 Everything is cool!
        collection.insert(entrada, function(err, result){
          if (err) {
            res.status(500);
            //Esto es para crear un json de error
            res.json({
              mensaje: "Error en la base de datos",
              insertado: false
            });
          }
        });
      }
    }
  );
});
```

Asignatura	Datos del alumno	Fecha
Ingeniería y Desarrollo en la Web	Apellidos: Quintero Gutiérrez	08/03/2019
	Nombre: Claudia Lorena	

Registro de usuarios

Para registrar desde postman



Asignatura	Datos del alumno	Fecha
Ingeniería y Desarrollo en la Web	Apellidos: Quintero Gutiérrez	08/03/2019
	Nombre: Claudia Lorena	

```
//Registrar usuario
Complexity is 7 It's time to do something...
app.post("/registrar",function(req,res){
  MongoClient.connect('mongodb://masterdb:GabSant2@ds225375.mlab.com:25375/blogtec',
    function(err,db){
      if (err) {
        res.status(500);
        //Esto es para crear un json de error
        res.json({
          mensaje:"Error en la base de datos",
          insertado:false
        });
      } else {
        console.log('Conectado al servidor');
        var usuario = {
          login:req.body.login,
          password:req.body.password
        };
        var collection = db.collection('usuarios');
        Complexity is 3 Everything is cool!
        collection.insert(usuario,function(err,result){
          if (err) {
            res.status(500);
            //Esto es para crear un json de error
            res.json({
              mensaje:"Error en la base de datos",
              insertado:false
            });
          } else {
            //Hay éxito y se ha creado un nuevo recurso
            res.status(201);
            //Esto es para crear un json de error
            res.json({
              insertado:true
            });
            db.close();
          }
        });
      }
    });
});
```

Asignatura	Datos del alumno	Fecha
Ingeniería y Desarrollo en la Web	Apellidos: Quintero Gutiérrez	08/03/2019
	Nombre: Claudia Lorena	

Autenticación

Verificamos si las credenciales ingresadas corresponden a las de la base de datos

```

app.get("/usuario/:login/:pass", function(req, res){
  //recibimos el id
  //Transformar en un objeto id de mongo
  var loginIng = req.params.login;
  var passIng = req.params.pass;
  MongoClient.connect('mongodb://masterdb:GabSant2@ds225375.mlab.com:25375/blogtec',
    function(err, db) {
      console.log("Conectado al servidor")
      var collection = db.collection('usuarios');
      collection.find({login:loginIng}).toArray(function (err, result) {
        res.status(200);
        //res.json(result);
        console.log(passIng);
        console.log(result[0].password);
        if (result[0].password == passIng) {
          console.log("usuario autenticado");
          res.json({
            mensaje:"El usuario se encuentra autenticado en el sistema"
          });
          req.session.nombre = loginIng
          console.log(req.session.nombre);
        } else {
          res.json({
            mensaje:"Password incorrecto"
          });
          console.log("credenciales inválidas")
        }
      });
    }
  );
  // Cerrar el cliente
  db.close();
}

```

Password incorrecto

Asignatura	Datos del alumno	Fecha
Ingeniería y Desarrollo en la Web	Apellidos: Quintero Gutiérrez	08/03/2019
	Nombre: Claudia Lorena	

The screenshot shows a REST client interface with the following details:

- URL:** `http://localhost:8080/`
- Method:** GET
- Path:** `http://localhost:8080/usuario/lorena27/123`
- Authorization:** Inherit auth from parent
- Body:** The response body is displayed in JSON format: `{ "mensaje": "Password incorrecto" }`
- Status:** 200 OK
- Time:** 1685 ms
- Size:** 245 B

Password Correcto

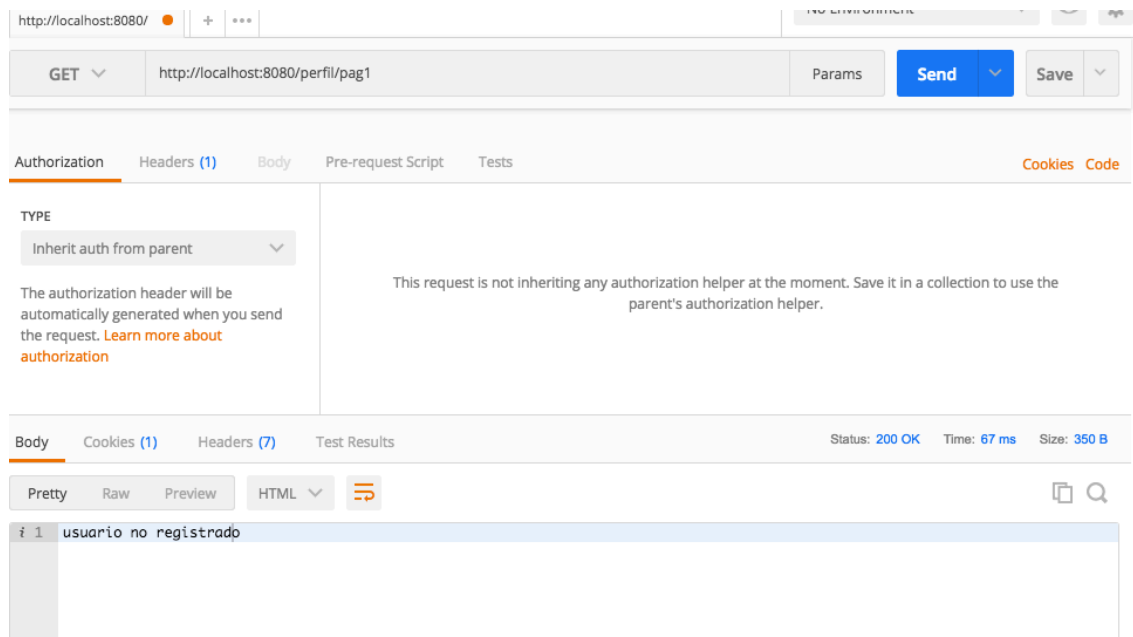
Usuario autenticado en el sistema

The screenshot shows a REST client interface with the following details:

- URL:** `http://localhost:8080/`
- Method:** GET
- Path:** `http://localhost:8080/usuario/lorena27/123Ga`
- Authorization:** Inherit auth from parent
- Body:** The response body is displayed in JSON format: `{ "mensaje": "El usuario se encuentra autenticado en el sistema" }`
- Status:** 200 OK
- Time:** 1389 ms
- Size:** 275 B

El usuario no tiene acceso a esta página

Asignatura	Datos del alumno	Fecha
Ingeniería y Desarrollo en la Web	Apellidos: Quintero Gutiérrez	08/03/2019
	Nombre: Claudia Lorena	



Para la sesión hacemos lo siguiente, instalamos express-session y definimos una ruta para la sesión

```
var session = require('express-session');
var zonaPrivada = express.Router();

app.use(session({
  secret: 'uiomdf',
  resave: true,
  saveUninitialized: true
}));
```

El usuario que tenga abierta la sesión es el único que puede acceder a /perfil/pag1

```
app.use('/perfil/', zonaPrivada);

app.get('/perfil/pag1', function(req, res){
  res.send('El ususario'+req.session.nombre+" tiene una sesión abierta");
});
```

Asignatura	Datos del alumno	Fecha
Ingeniería y Desarrollo en la Web	Apellidos: Quintero Gutiérrez	08/03/2019
	Nombre: Claudia Lorena	

Cliente Web

Obtener todas las entradas desde el cliente, para eso debemos instalar request

```

var request = require('request');

// Obtener todas las entradas de blog
var configuracion = {
  url: 'http://localhost:8080/',
  method: 'GET'
};

request(configuracion, function(error, response, body){
  console.log('Estos son los blog');
  var entradas = JSON.parse(body);
  for (let index = 0; index < entradas.length; index++) {
    console.log(`-----Este es el Blog:${index}-----`);
    console.log(`Autor: ${entradas[index].autor}`);
    console.log(`Fecha: ${entradas[index].fecha}`);
    console.log(`Tema: ${entradas[index].tema}`);
    console.log(`-----Comentarios-----`);

    for (const key in entradas[index].comentarios) {
      console.log(`Estos son los comentarios del Blog de: ${entradas[index].autor} `);
      console.log(`**${entradas[index].comentarios[key]}`);
    }

    console.log(`-----Fin-----`);
  }
});

```

Recorremos si tiene comentarios y también lo visualizamos así:

Asignatura	Datos del alumno	Fecha
Ingeniería y Desarrollo en la Web	Apellidos: Quintero Gutiérrez	08/03/2019
	Nombre: Claudia Lorena	

Estos son los blog

-----Este es el Blog:0-----

Autor: javier

Fecha: 2019-02-14

Tema: angular

-----Comentarios-----

Estos son los comentarios del Blog de: javier

**este es el comentario

Estos son los comentarios del Blog de: javier

**este es el segundo comentario

Estos son los comentarios del Blog de: javier

**Este es el tercer comentario

Estos son los comentarios del Blog de: javier

**Este es el cuarto comentario

Estos son los comentarios del Blog de: javier

**Este es el quinto comentario

Estos son los comentarios del Blog de: javier

**Este es el sexto comentario

-----Fin-----

-----Este es el Blog:1-----

Autor: Gabriel

Fecha: 2019-02-12

Tema: javascript

-----Comentarios-----

Estos son los comentarios del Blog de: Gabriel

**Primer comentario de gabriel

Estos son los comentarios del Blog de: Gabriel

**Segundo comentario de gabriel

Estos son los comentarios del Blog de: Gabriel

**Tercer comentario de gabriel

-----Fin-----

Asignatura	Datos del alumno	Fecha
Ingeniería y Desarrollo en la Web	Apellidos: Quintero Gutiérrez	08/03/2019
	Nombre: Claudia Lorena	

Crear entrada desde el cliente

```
//Crear entrada
var entrada = {
  autor: 'Miranda desde el cliente',
  fecha: '2019-03-07',
  tema: 'Mongo db'
}
var configuracionCrear = {
  url: 'http://localhost:8080/entrada',
  method: 'POST',
  json: true,
  headers: {
    "content-type": "application/json"
  },
  body: entrada
};
request(configuracionCrear, function (error, response, body) {
  console.log("*****");
  console.log("Blog Creado desde el cliente");
  console.log("*****");
});
```

```
*****
Blog Creado desde el cliente
*****
```

Borrar entrada

Para esto suministramos el ID

Asignatura	Datos del alumno	Fecha
Ingeniería y Desarrollo en la Web	Apellidos: Quintero Gutiérrez	08/03/2019
	Nombre: Claudia Lorena	

```

var idBlogBorrar = "5c820cef378afe43092c48b5";

var configuracionBorrar = {
  url: "http://localhost:8080/entrada/"+idBlogBorrar,
  method: "DELETE",
}

request(configuracionBorrar, function (error, response, body) {
  console.log("-----");
  console.log("Borrando el blog "+idBlogBorrar);
  console.log("-----");
})

```

Y obtenemos

```

-----
Borrando el blog 5c820cef378afe43092c48b5
-----

```

Obtener Entrada por id

```

// //Obtener una entrada por id
var id = '5c820c4f378afe43092c48b4'
var configuracionPorId = {
  url: 'http://localhost:8080/entrada/'+id,
  method: 'GET'
};

request(configuracionPorId, function(error, response, body){
  console.log('Estos es la información del blog'+id);
  var entrada = JSON.parse(body);
  console.log(entrada[0].autor);
  console.log(`-----Este es el Blog:${entrada[0].autor}-----`);
  console.log(`Fecha: ${entrada[0].autor}`);
  console.log(`Tema: ${entrada[0].tema}`);
  console.log(`-----Comentarios-----`);

  for (const key in entrada[0].comentarios) {
    console.log(`Estos son los comentarios del Blog de: ${entrada[0].autor} `);
    console.log(`**${entrada[0].comentarios[key]}`);
  }

  console.log(`-----Fin-----`);
})

```


Asignatura	Datos del alumno	Fecha
Ingeniería y Desarrollo en la Web	Apellidos: Quintero Gutiérrez	08/03/2019
	Nombre: Claudia Lorena	

```

Estos es la información del blog5c820c4f378afe43092c48b4
Miranda desde el cliente
-----Este es el Blog:Miranda desde el cliente-----
Fecha: Miranda desde el cliente
Tema: Mongo db
-----Comentarios-----
-----Fin-----

```

Modificar entrada:

Para esto la entrada debe estar almacenada en la base de datos y lo que hacemos es modificarla ingresando los comentarios respectivos

Asignatura	Datos del alumno	Fecha
Ingeniería y Desarrollo en la Web	Apellidos: Quintero Gutiérrez	08/03/2019
	Nombre: Claudia Lorena	

```
// Modificar una entrada por id, adicionándole comentarios
// Para agregar comentarios se necesita que la entrada
// este almacenada
var id = '5c5e314be0e40aae0534a6d8';

var entrada = {
  comentarios:"Esto es un comentario desde el cliente web "
}
var configuracionPUT = {
  url: 'http://localhost:8080/entrada/'+id,
  method: 'PUT',
  json: true,
  headers: {
    "content-type": "application/json"
  },
  body: entrada
};

Complexity is 3 Everything is cool!
request(configuracionPUT, function (error, response, body) {
  if (error) {
    console.log('Error ');
  } else {
    console.log("*****");
    console.log("Blog Modificado ");
    console.log("*****");
  }
});
```

```
[Running] node "/Users/lorenaquintero/Google Drive/MAESTRIA UNIR 2019/Ingenieria y Desarrollo en la Web/Copia de
Actividad2RestLorenaQuintero/NuevoClienteServicioWeb/main.js"
*****
Blog Modificado
*****

[Done] exited with code=0 in 1.827 seconds
```

Y desde el postman

Asignatura	Datos del alumno	Fecha
Ingeniería y Desarrollo en la Web	Apellidos: Quintero Gutiérrez	08/03/2019
	Nombre: Claudia Lorena	

GET
http://localhost:8080/entrada/5c5e314be0e40aae0534a6d8
Params
Send
Save

TYPE
Inherit auth from parent
The authorization header will be automatically generated when you send the request. [Learn more about authorization](#)
This request is not inheriting any authorization helper at the moment. Save it in a collection to use the parent's authorization helper.

Body
Cookies (1)
Headers (6)
Test Results
Status: 200 OK
Time: 968 ms
Size:

Pretty
Raw
Preview
JSON

```

1  [
2  {
3    "_id": "5c5e314be0e40aae0534a6d8",
4    "autor": "franck",
5    "fecha": "2019-02-13",
6    "tema": "node js",
7    "comentarios": [
8      "Este es un comentario desde postman",
9      "Esto es un comentario desde el cliente web "
10   ]
11  }

```