Miriam Guerra Guerra

Tarea NODE.

Conexión y esquema tienda (pop up) y cliente:

```
//conectamos

const mongoose = require('mongoose'); //paquete mongoose
const express = require('express');

const app = express(); //creamos la app
const port = 3000;
//arranca el servidor
app.listen(port, () => {console.log(`puerto ${port}`);});

app.use(express.urlencoded({ extended: true}));
// app.use(express.static('htdocs'));

main();

async function main(){
    await mongoose.connect ('mongodb+srv://usermm:zLj0Ux2lCmJYeKS1@majadabae.egiyzan.mongodb.
    net/popup?retryWrites=true&w=majority&appName=MajadaBAE')
console.log("conectado");

// Datos de las tablas

const popupSchema = new mongoose.Schema({
    nombre: String,
    precio: Number,
    tipoPago: String,
    fecha: Date,
    alquilado: Boolean,
    lugar: String,
    tipo: Array
  });

  const clienteSchema = new mongoose.Schema({
    nombre: String,
    apellidos: String,
    tarifa: Array, //tipo de suscripción, para saber la confianza en el cliente
    contacto: String
  });
```

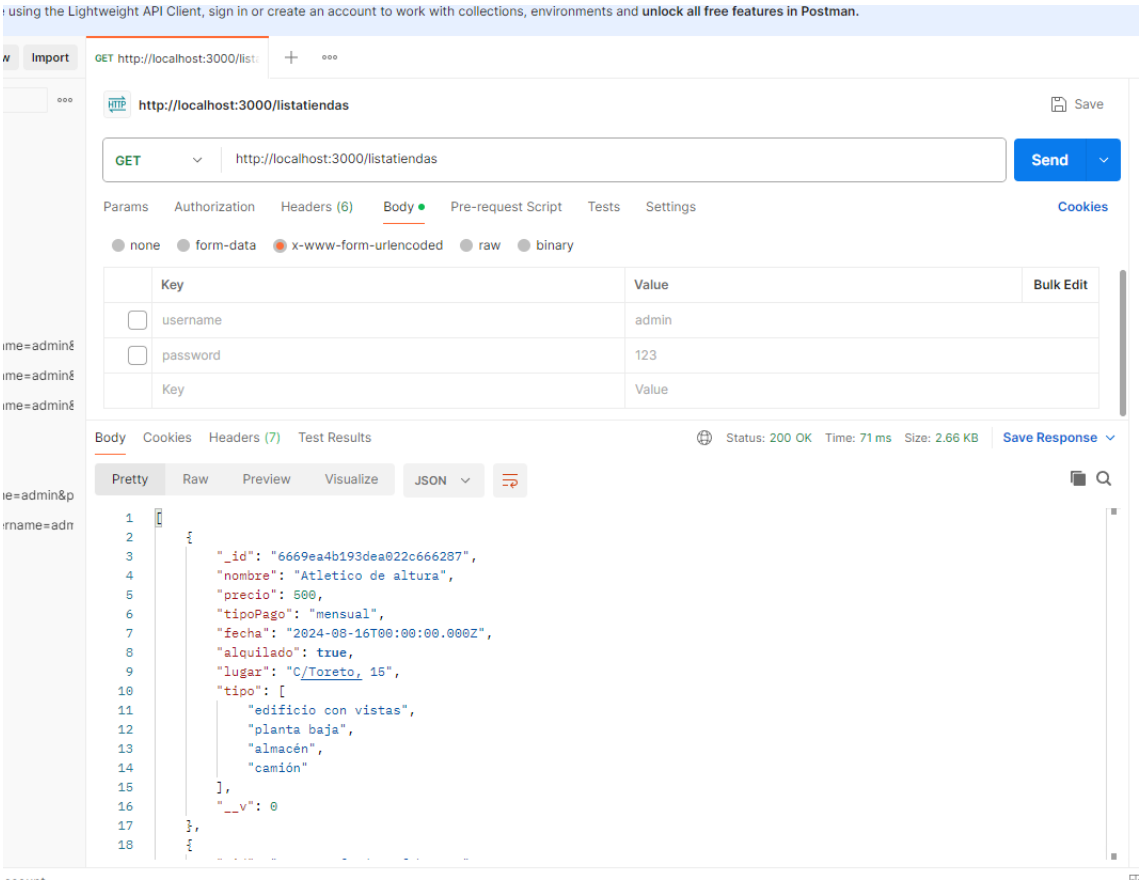
Tablas y guardado de datos:

```
// creamos la colección de las tiendas
const Tienda = mongoose.model('popups', popupSchema);

// creamos la colección de los clientes

const Cliente = mongoose.model('clientes', clienteSchema);

// metemos los datos en las tablas

const latienda =
  new Tienda ({
  nombre: 'Atletico de altura',
  precio: 500,
  tipoPago: 'mensual',
  fecha: new Date ('2024-08-16'),
  alquilado: true,
  lugar: 'C/Toreto, 15',
  tipo: ["edificio con vistas", "planta baja", "almacén", "camión"]
});

/* ...

const losclientes =
  new Cliente({
      nombre: 'Lorenzo',
      apellidos: 'Bermudez Moreno',
      tarifa: [
          { tipo: 'premium', constancia: 'ininterrumpida' }
      ],
      contacto: 'lorenzo@gmail.com'
  });


// guardamos los datos

await latienda.save();
await losclientes.save();
console.log(latienda.nombre);
console.log(losclientes.nombre);
```

Find TIENDA:

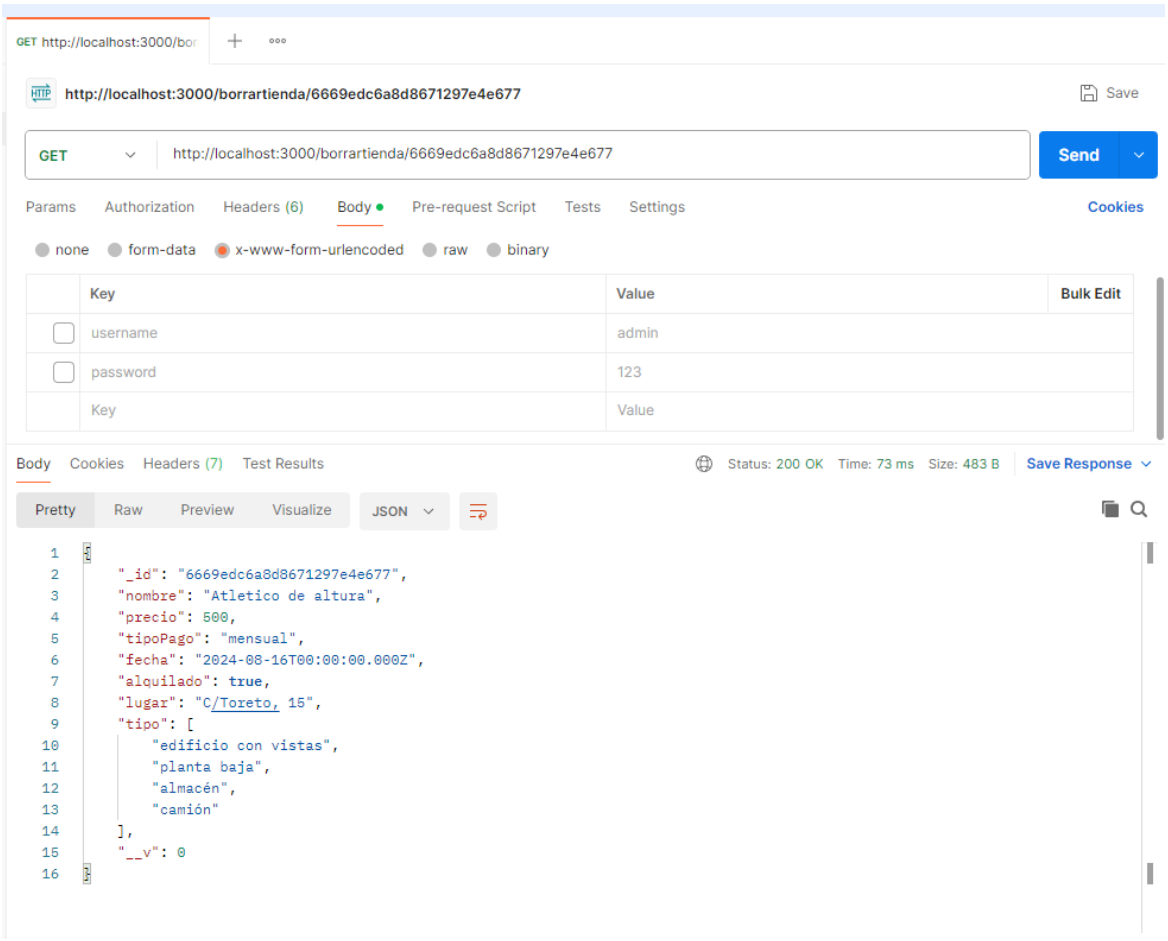
```
//FIND/SELECT

app.get('/listatiendas', async (req,res)=>{
  let shop = await Tienda.find();
  res.send(shop);
})

const tiendita = await Tienda.find();
console.log(tiendita);
```

Resultado find:



delete TIENDA:

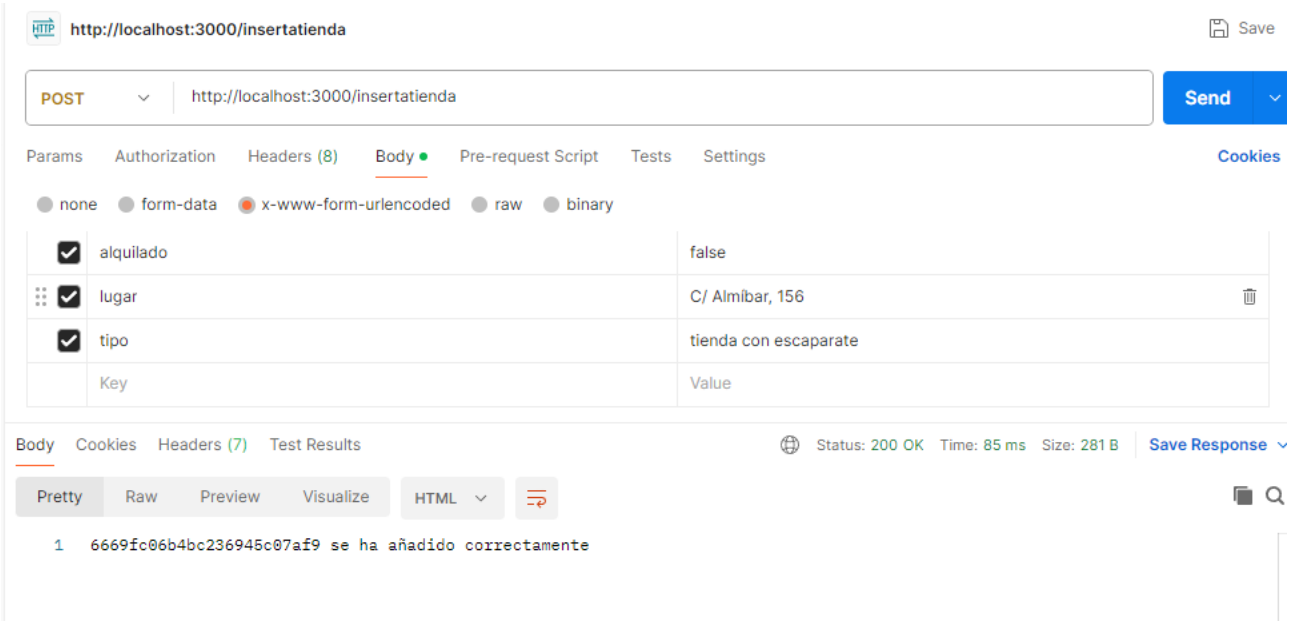
Resuldado delete:



Insert TIENDA:

```
// insertar

// datos del formulario
app.post('/insertatienda', async (req,res)=>{
  let nombre = req.body.nombre;
  let precio = req.body.precio;
  let tipoPago = req.body.tipoPago;
  let fecha = req.body.fecha;
  let alquilado = req.body.alquilado;
  let lugar = req.body.lugar;
  let tipo = req.body.tipo;
// crea la nueva tienda
  const nuevaTienda =
  new Tienda ({
  nombre: nombre,
  precio: precio,
  tipoPago: tipoPago,
  fecha: fecha,
  alquilado: alquilado,
  lugar: lugar,
  tipo: tipo
});
// guarda los datos
await nuevaTienda.save();
// muestra que todo salió bien
res.send(nuevaTienda.id+' se ha añadido correctamente');


});
```

Resultado insert:



```
197
198   const tiendita = await Tienda.find();
199   console.log(tiendita);
200
201
202
203
204
205
```

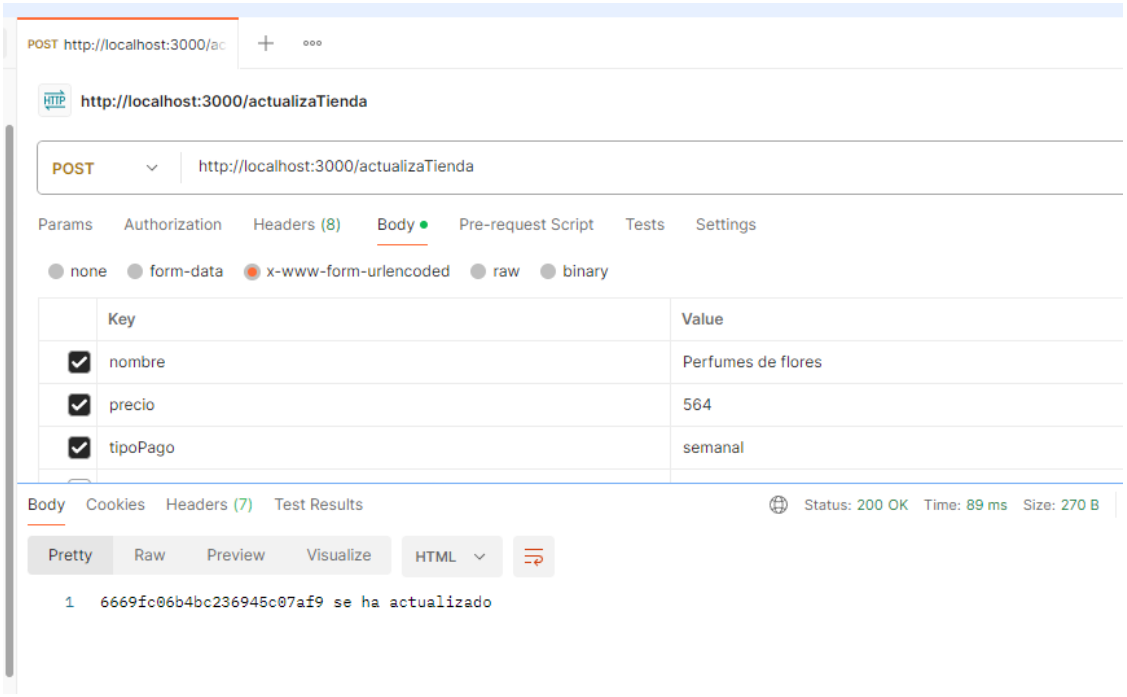
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

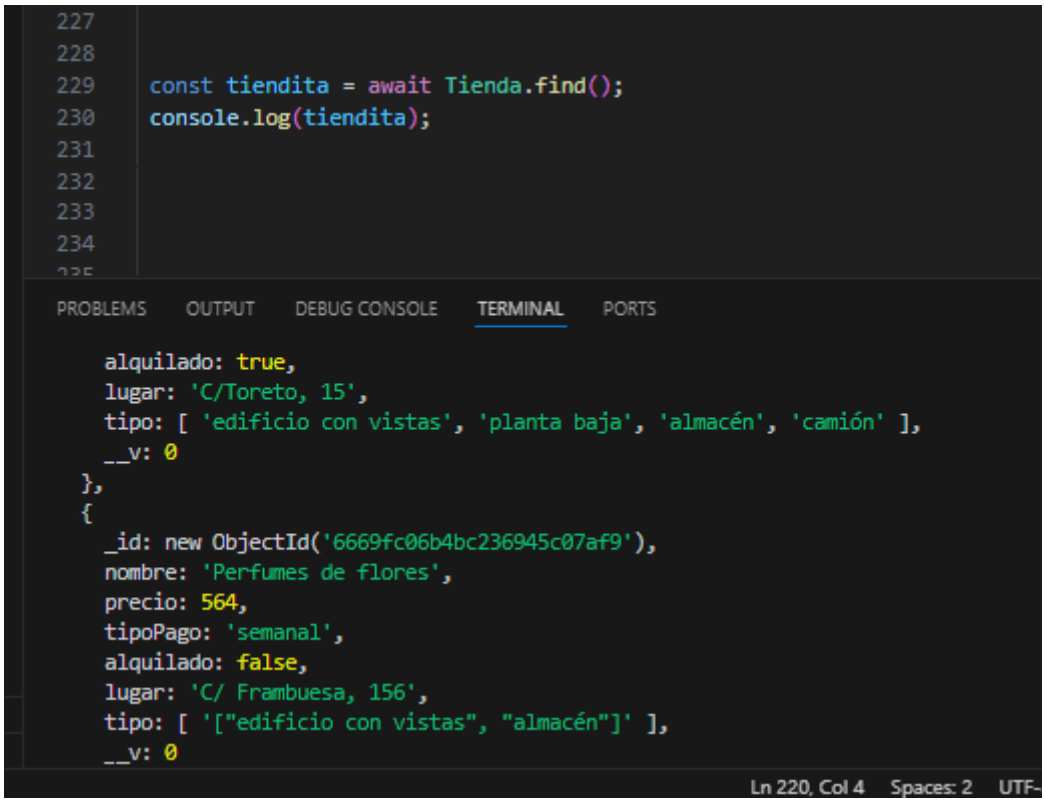
```
    lugar: 'C/Toreto, 15',
    tipo: [ 'edificio con vistas', 'planta baja', 'almacén', 'camión' ],
    __v: 0
  },
  {
    _id: new ObjectId('6669fc06b4bc236945c07af9'),
    nombre: 'Perfumes naturales',
    precio: 564,
    tipoPago: 'semanal',
    alquilado: false,
    lugar: 'C/ Almíbar, 156',
    tipo: [ 'tienda con escaparate' ],
    __v: 0
  },
```

Update TIENDA:

```
196  });
197
198  // update
199
200  app.post('/actualizaTienda', async function (req,res){
201    // datos del formulario
202    let id = req.body.id;
203    let nombre = req.body.nombre;
204    let precio = req.body.precio;
205    let tipoPago = req.body.tipoPago;
206    let fecha = req.body.fecha;
207    let alquilado = req.body.alquilado;
208    let lugar = req.body.lugar;
209    let tipo = req.body.tipo;
210
211  // buscar por id y cambiar datos
212  let shop = await Tienda.findByIdAndUpdate(id, {
213    nombre: nombre,
214    precio: precio,
215    tipoPago: tipoPago,
216    fecha: fecha,
217    alquilado: alquilado,
218    lugar: lugar,
219    tipo: tipo
220  });
221  res.send (shop.id+' se ha actualizado');
222  });
```

Resultado update:

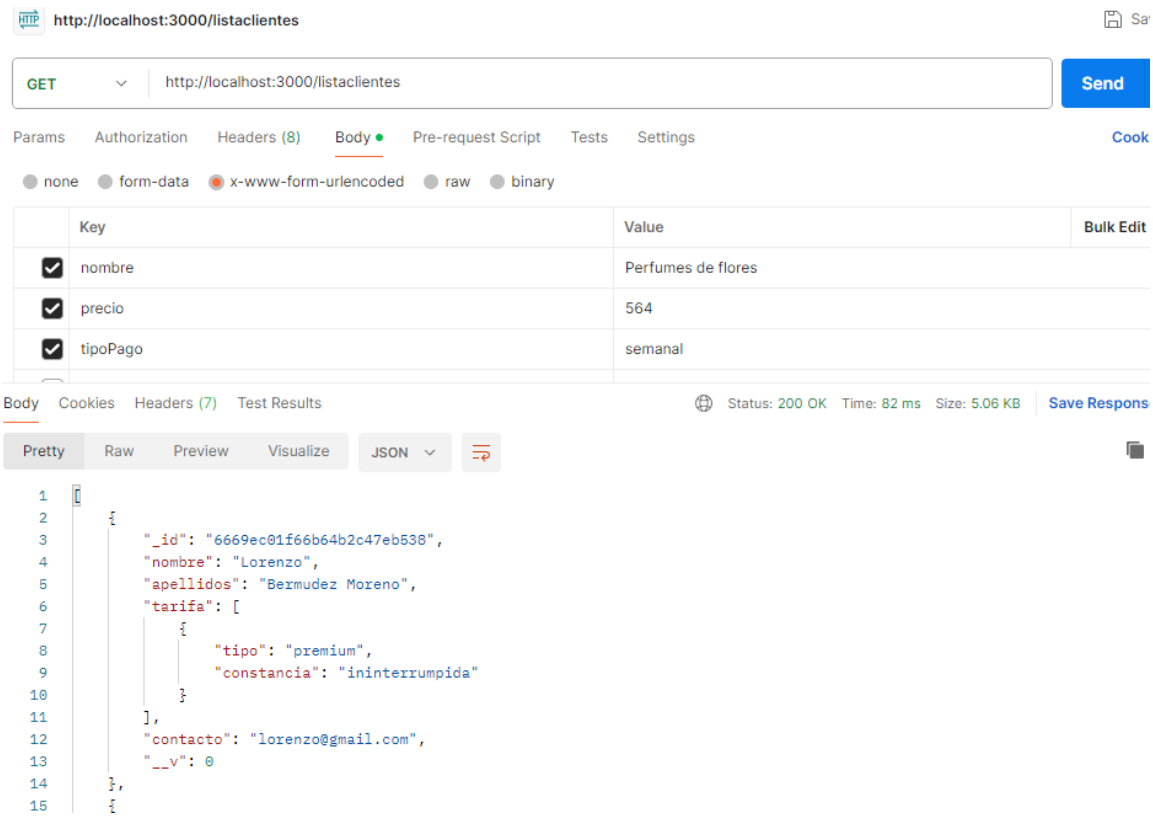




Find CLIENTE:

```
18
19  app.get('/listaclientes', async (req,res)=>{
20    let cliente = await Cliente.find();
21    res.send(cliente);
22  })
```

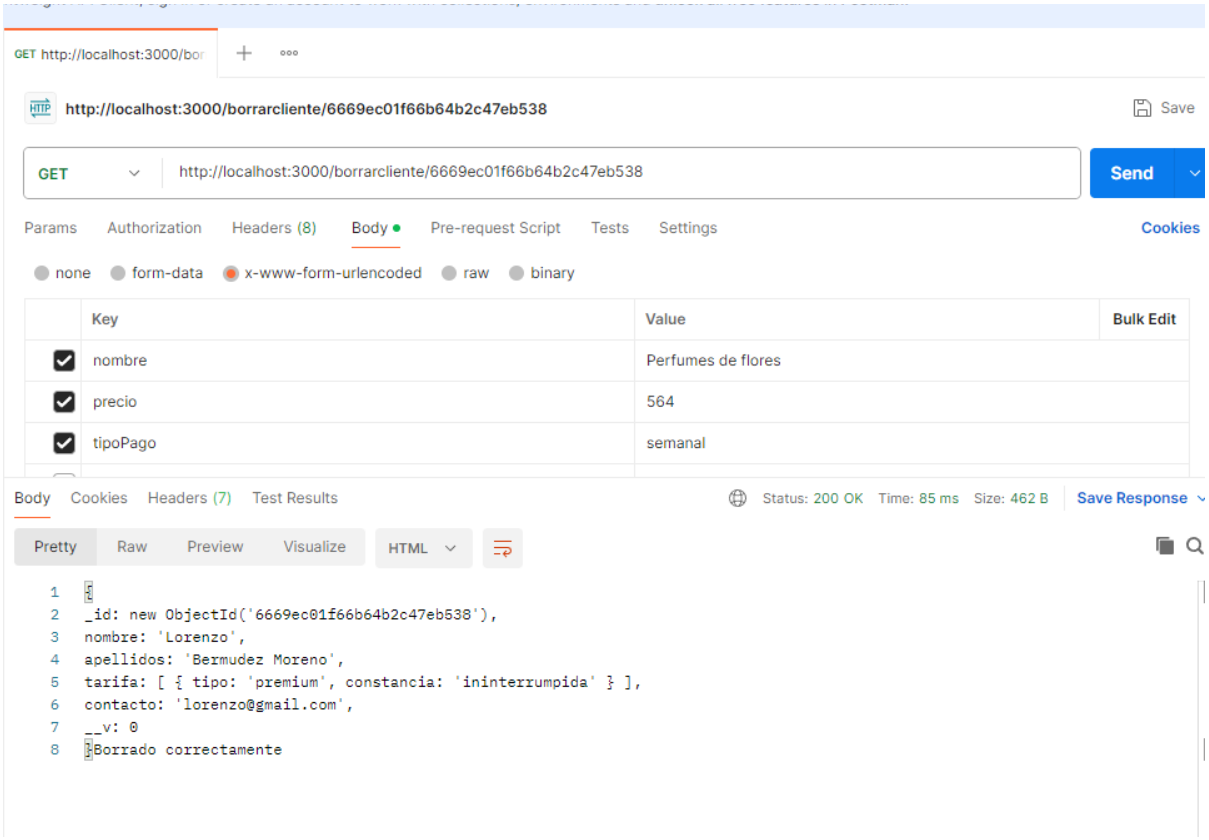
Resultado:



delete CLIENTE:

```
1
2  app.get('/borrarcliente/:id', async (req,res)=>{
3    let borraCliente = await Cliente.findByIdAndDelete(req.params.id);
4    res.send(borraCliente + 'Borrado correctamente');
5  })
6
```

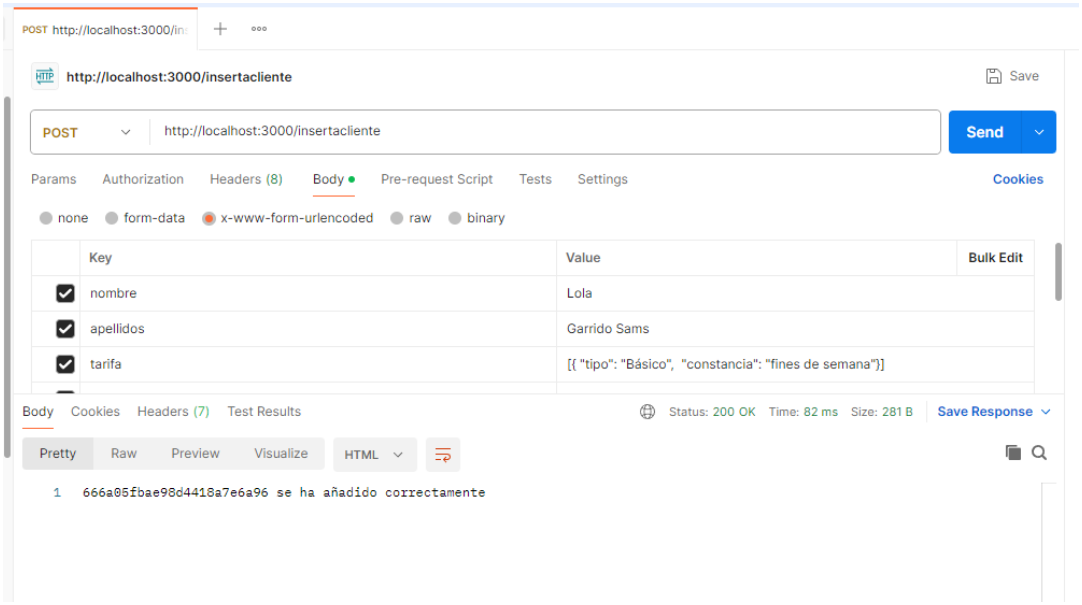
Resuldado delete:

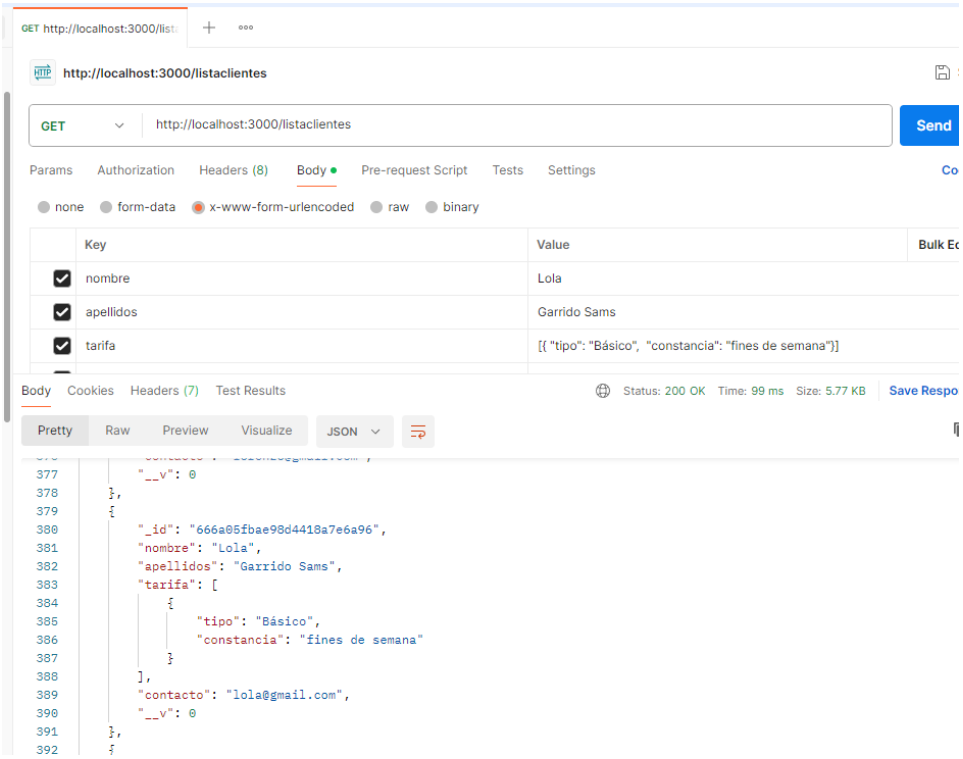


Insert CLIENTE:

```
69
70  app.post('/insertacliente', async (req,res)=>{
71    let nombre = req.body.nombre;
72    let apellidos = req.body.apellidos;
73    let tarifa = JSON.parse(req.body.tarifa);
74    let contacto = req.body.contacto;
75
76  // crea nuevo cliente
77    const nuevoCliente =
78    new Cliente ({
79    nombre: nombre,
80    apellidos: apellidos,
81    tarifa: tarifa,
82    contacto: contacto,
83  });
84  // guarda los datos
85  await nuevoCliente.save();
86  // muestra que todo salió bien
87  res.send(nuevoCliente.id+' se ha añadido correctamente');
88  });
89
90
91
```

Resultado insert:

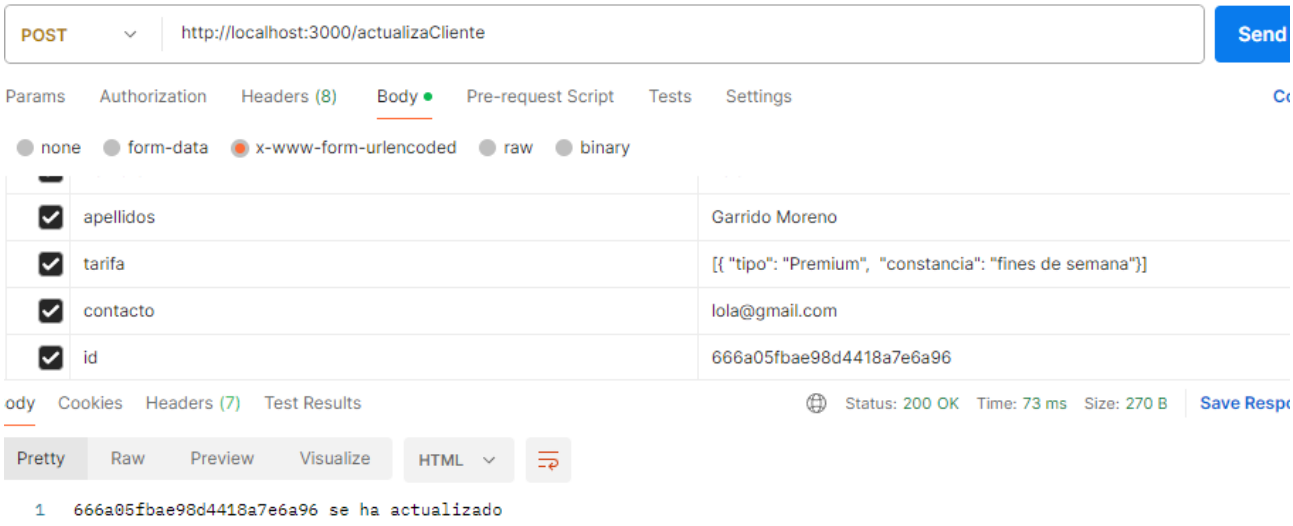




Update CLIENTE:

```
app.post('/actualizaCliente', async function (req,res){
  // datos del formulario
  let id = req.body.id;
  let nombre = req.body.nombre;
  let apellidos = req.body.apellidos;
  let tarifa = JSON.parse(req.body.tarifa);
  let contacto = req.body.contacto;

// buscar por id y cambiar datos
let cliente = await Cliente.findByIdAndUpdate(id, {
  nombre: nombre,
  apellidos: apellidos,
  tarifa: tarifa,
  contacto: contacto,
});
res.send (cliente.id+' se ha actualizado');
});
```

Resultado update:



http://localhost:3000/listaclientes

GET http://localhost:3000/listaclientes

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary

apellidos	Garrido Moreno
tarifa	[{ "tipo": "Premium", "constancia": "fines de semana"}]
contacto	lola@gmail.com
id	666a05fbae98d4418a7e6a96

Body Cookies Headers (7) Test Results Status: 200 OK Time: 65 ms Size: 6.13 KB Save R

Pretty Raw Preview Visualize JSON


```
},
{
    "_id": "666a05fbae98d4418a7e6a96",
    "nombre": "Lola",
    "apellidos": "Garrido Moreno",
    "tarifa": [
        {
            "tipo": "Premium",
            "constancia": "fines de semana"
        }
    ],
    "contacto": "lola@gmail.com",
    "__v": 0
},
```


Todo correcto en ATLAS: