Daniela Aragón Risso

PROGRAMACIÓN 1º DAW 2022-23

Objetivo:

Realizar una **API Rest para el CRUD** de una base de datos Mongo usando Mongoose y Express.js

Requisitos:

Mínimos (5p):

Operaciones CRUD de una colección.
Creación de un Schema con variedad de tipos campos: String, Number, Date, Boolean, Arrays.

Ampliaciones (hasta 10p):

- Otras operaciones: Por ejemplo: Filtro de busqueda (1p)
  Filtro por marca en las fundas
- Otra colección: (2p)
  Cristales
- Relaciones entre colecciones: (1p)
- Frontend web (2p)
- Otras: Control y Gestión de errores (1p) , confirmación de borrado (1p).

Entrega:

Archivos de la API

Lista de endpoints creados con capturas de pantalla demostrando que funcionan.

Usar Postman o extensiones de Visual Studio.

Daniela Aragón Risso

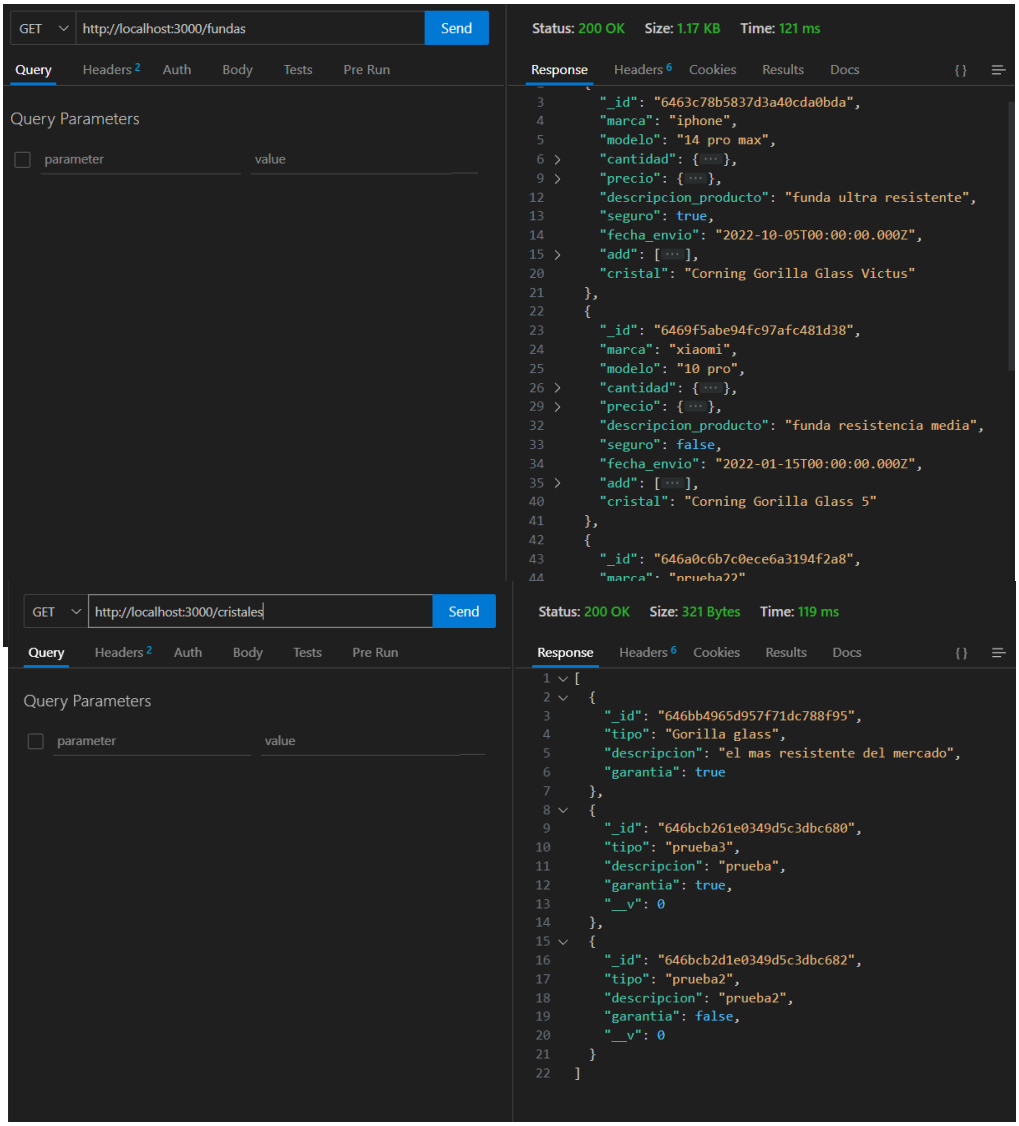
Capturas Funcionamiento Schema

```
1.- Schema

*/


const fundaSchema = new mongoose.Schema({
  marca: String,
  modelo: String,
  cantidad: Decimal128,
  precio: Decimal128,
  fecha_envio: Date,
  descripcion_producto: String,
  cristal: String,
  seguro: Boolean,
  add: Array
});

const cristalesSchema = new mongoose.Schema({
  tipo: String,
  descripcion: String,
  garantia: Boolean,
});
```

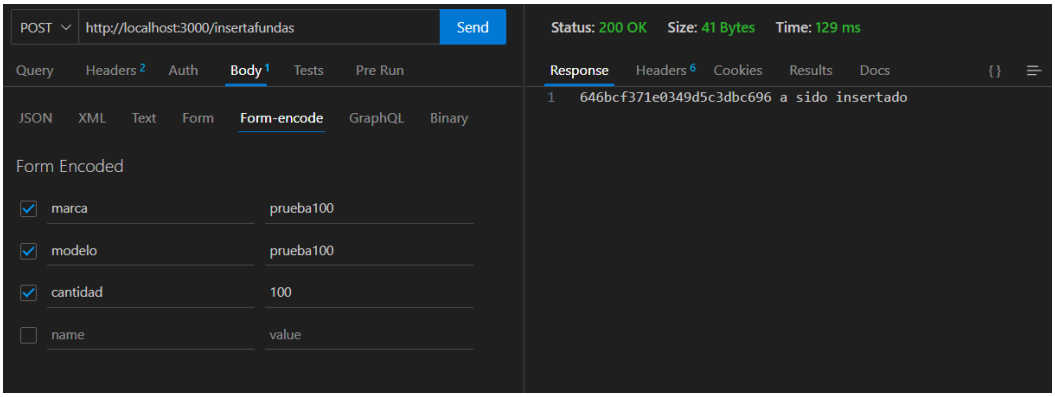
Para buscar en la base de datos mediante GET

```
//GET fundas/marca : Listado de fundas por nombre

app.get('/fundas/:marca?', async function (req, res) {

  let marca = req.params.marca || '';
  let fundas;

  if (marca == '') {
    fundas = await funda.find();
  } else {
    fundas = await funda.find({ marca: marca });
  }

  res.send(fundas);

});

//GET /cristales : Listado de cristales

app.get('/cristales', async function (req, res) {

  let cristal;

    cristal = await cristales.find();


  res.send(cristal);

});
```

Daniela Aragón Risso



Para inserta mediante POST:

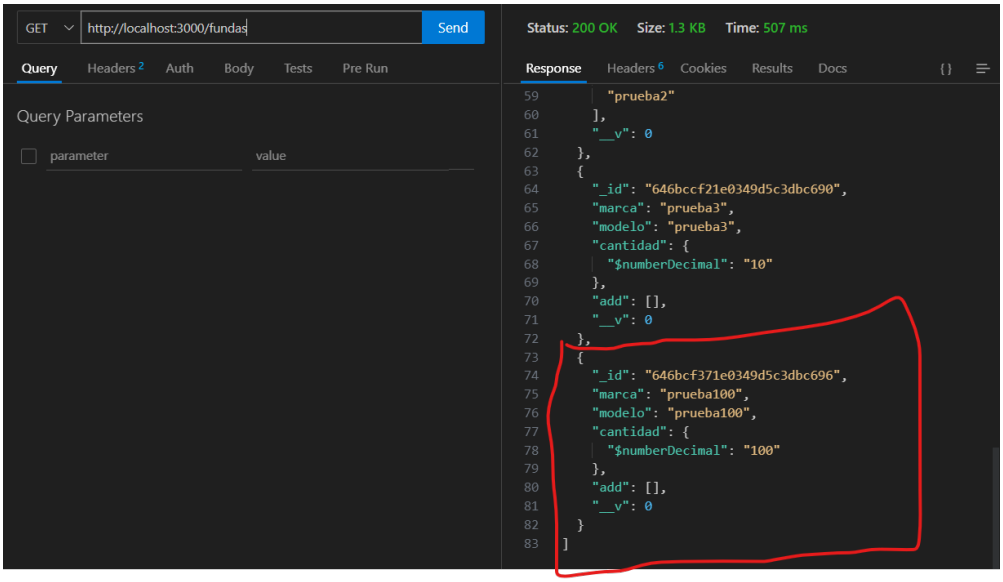


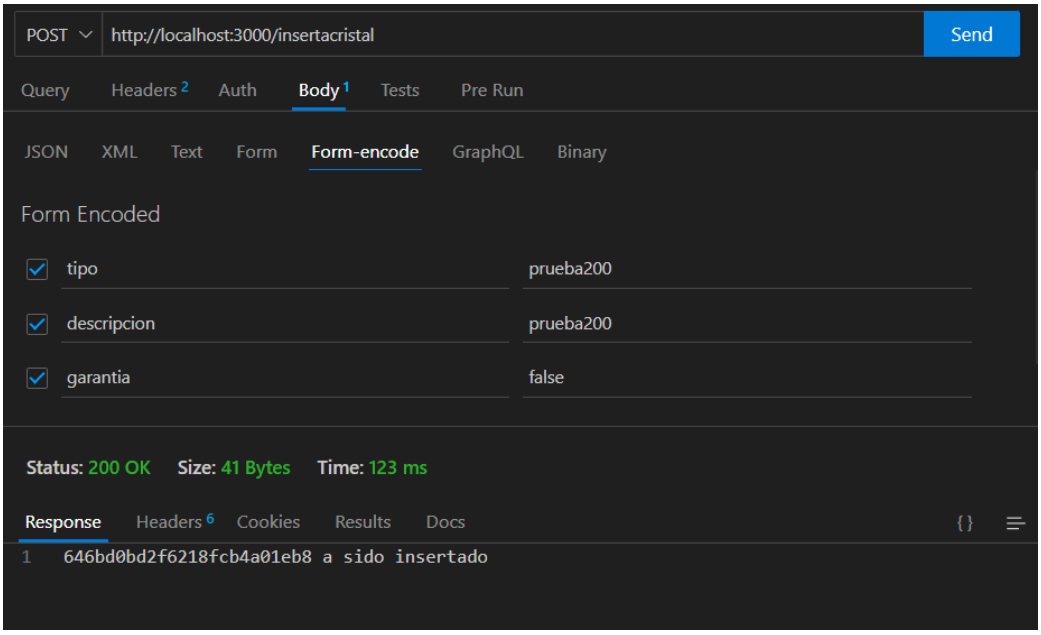
Daniela Aragón Risso

```
//POST insertafundas

app.post('/insertafundas', async function (req, res) {

  //Recoge datos formulario

  let marca = req.body.marca;
  let modelo = req.body.modelo;
  let cantidad = req.body.cantidad;
  let precio = req.body.precio;
  let fecha_envio = req.body.fecha_envio;
  let descripcion_producto = req.body.descripcion_producto;
  let cristal = req.body.cristal;
  let seguro = req.body.seguro;
  let add = req.body.add;


  //crea una instancia de usuarios con esos datos

  let nuevafunda = new funda(
    {
      marca: marca,
      modelo: modelo,
      cantidad: cantidad,
      precio: precio,
      fecha_envio: fecha_envio,
      descripcion_producto: descripcion_producto,
      cristal: cristal,
      seguro: seguro,
      add: add
    }
```

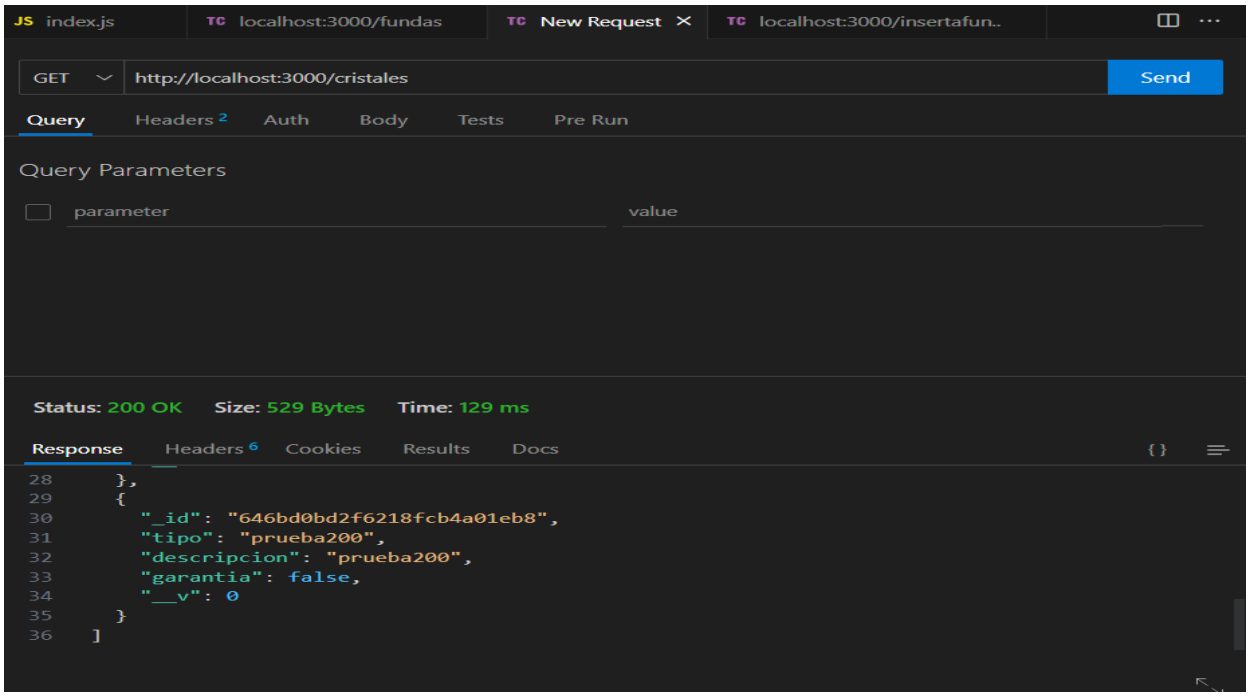
```
  //hace un 'save' de esa instancia

  await nuevafunda.save();

  //informa al cliente

  res.send(nuevafunda.id+' a sido insertado');

});



  //POST insertacristales

app.post('/insertacristal', async function (req, res) {

  //Recoge datos formulario

  let tipo = req.body.tipo;
  let descripcion = req.body.descripcion;
  let garantia = req.body.garantia;



  //crea una instancia de usuarios con esos datos

  let nuevocristal = new cristales(
    {
      tipo: tipo,
      descripcion: descripcion,
      garantia: garantia
    }
  );
```

Daniela Aragón Risso

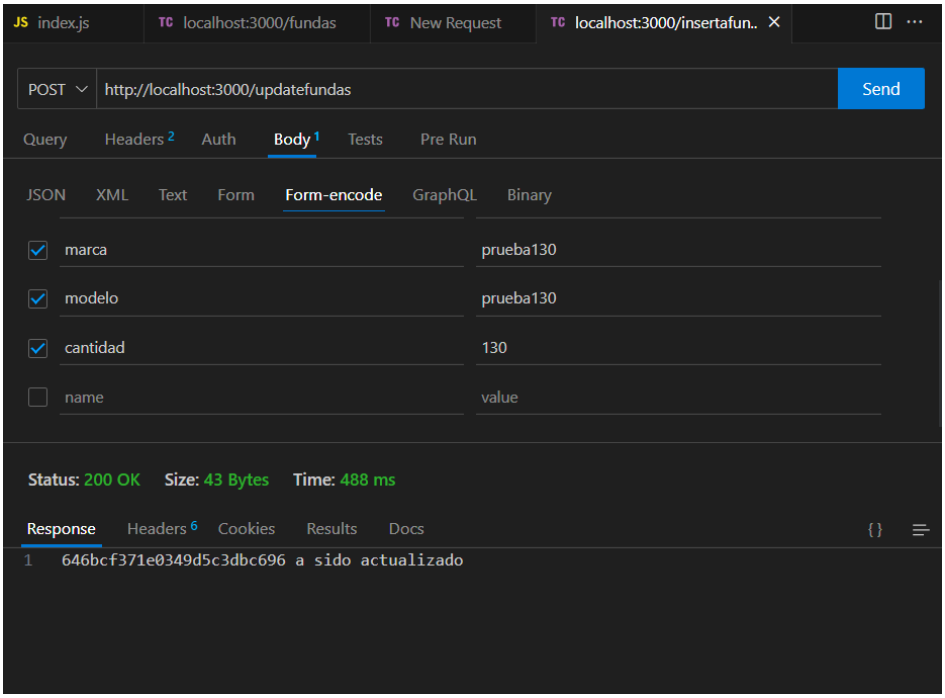




Daniela Aragón Risso



Update mediante POST:

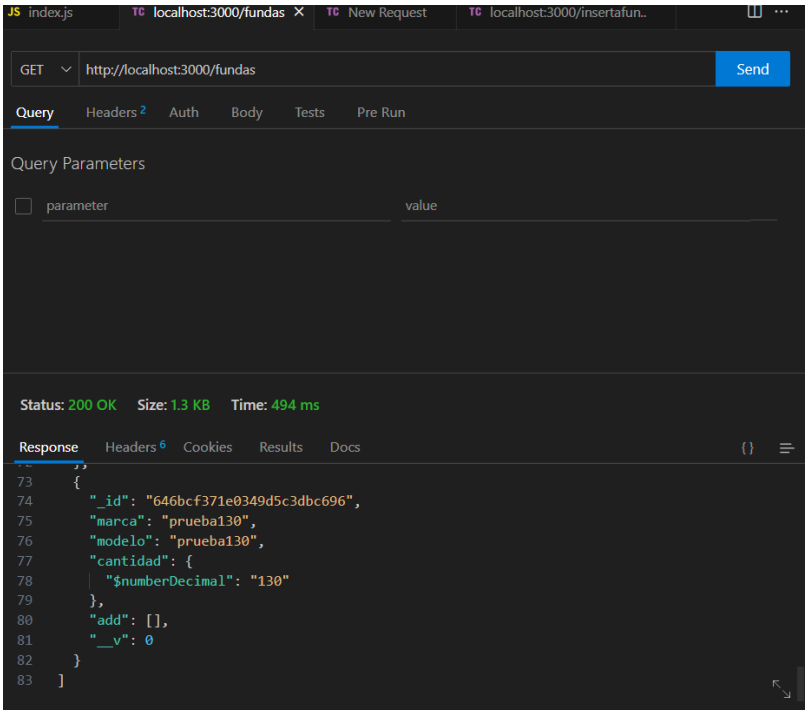


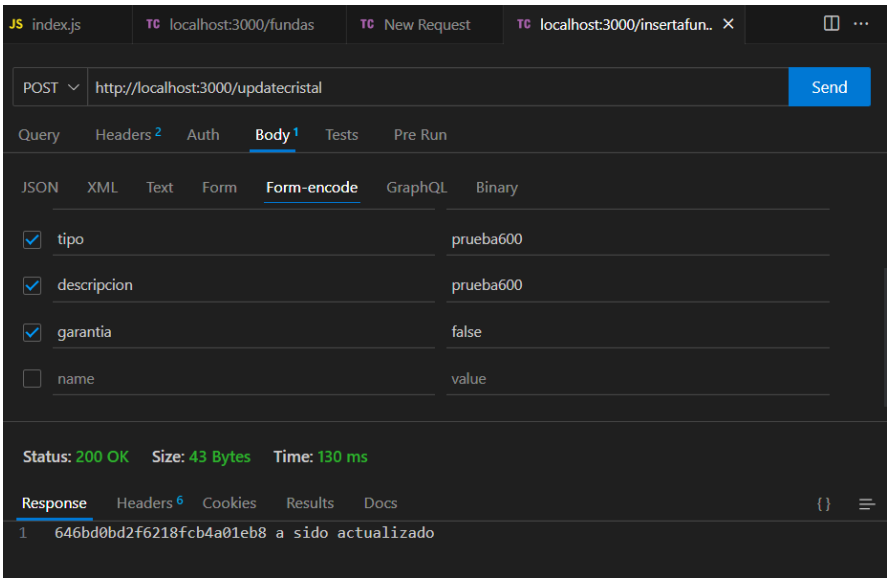
Daniela Aragón Risso

```
// update == POST
app.post('/updatefundas', async function (req, res) {
  //coger la id

  let id = req.body.id;


  //coger los datos

    let marca = req.body.marca;
    let modelo = req.body.modelo;
    let cantidad = req.body.cantidad;
    let precio = req.body.precio;
    let fecha_envio = req.body.fecha_envio;
    let descripcion_producto = req.body.descripcion;
    let cristal = req.body.cristal;
    let seguro = req.body.seguro;
    let add = req.body.add;

  //fundasfindByIdandAndUpdate
                                                  let id: any
  let fundas = await funda.findByIdAndUpdate(id,

    {
      marca: marca,
      modelo: modelo,
      cantidad: cantidad,
      precio: precio,
      fecha_envio: fecha_envio,
      descripcion_producto: descripcion_producto,
      cristal: cristal,
      seguro: seguro,
      add: add
```

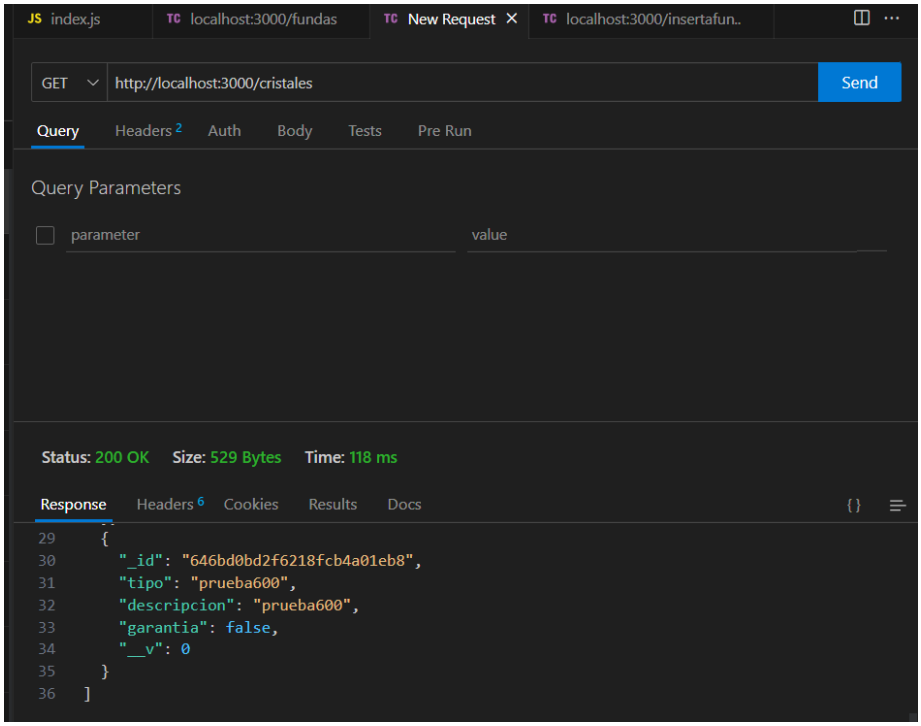
```
      seguro: seguro,
      add: add
    }

  );


  // responder al cliente

  res.send(fundas.id+' a sido actualizado');

});




// update == POST
app.post('/updatecristal', async function (req, res) {
  //coger la id

  let id = req.body.id;


  //coger los datos

  let tipo = req.body.tipo;
  let descripcion = req.body.descripcion;
  let garantia = req.body.garantia;

  //CristalesfindByIdandAndUpdate

  let cristal = await cristales.findByIdAndUpdate(id,
```

Daniela Aragón Risso





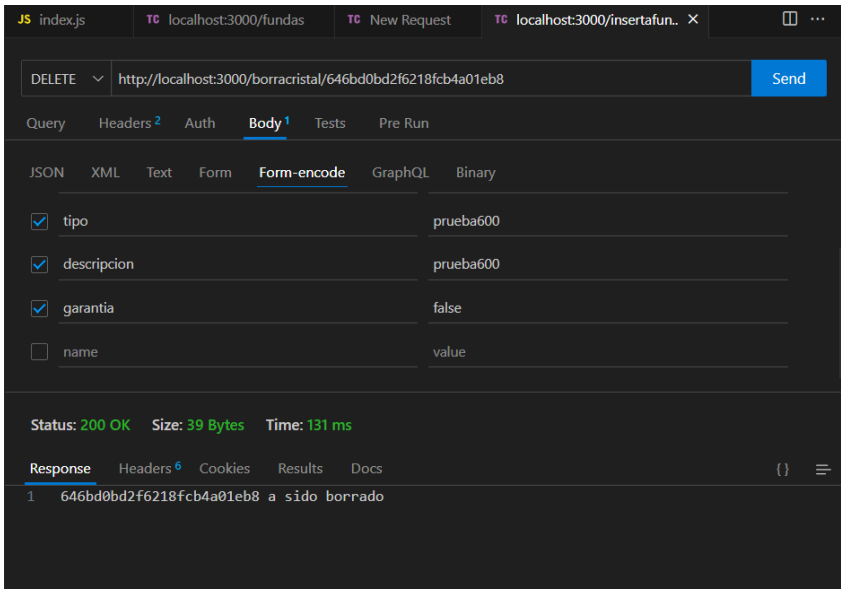
Daniela Aragón Risso

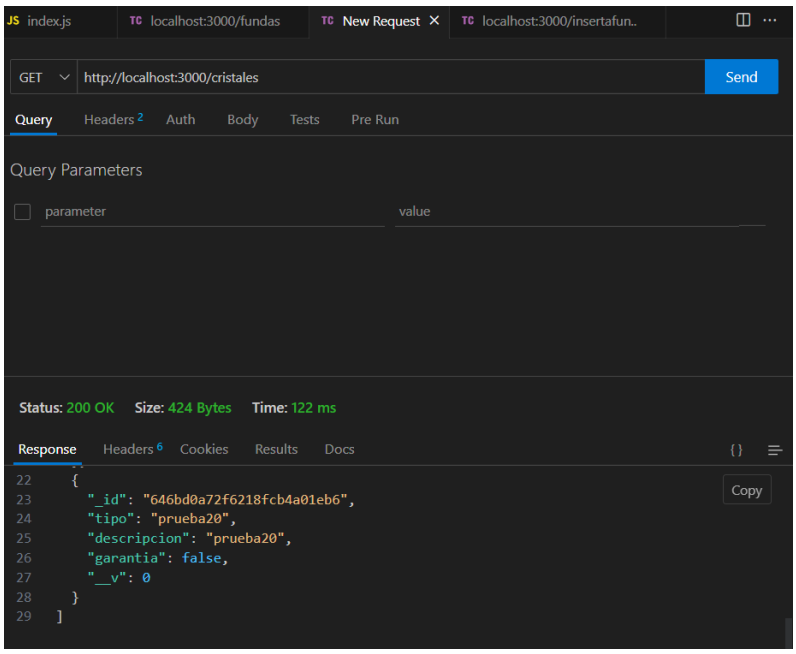


El borrado mediante DELETE:

```
//DELETE (GET) borrafundas/:id
app.delete('/borrafundas/:id', async function (req, res) {
  let id = req.params.id;
  let borrafundas = await funda.findByIdAndDelete(id);
  res.send(borrafundas.id+' a sido borrado');
});
//DELETE (GET) borracristal/:id
app.delete('/borracristal/:id', async function (req, res) {
  let id = req.params.id;
  let borracristal = await cristales.findByIdAndDelete(id);
  res.send(borracristal.id+' a sido borrado');
});
```

Daniela Aragón Risso





Daniela Aragón Risso