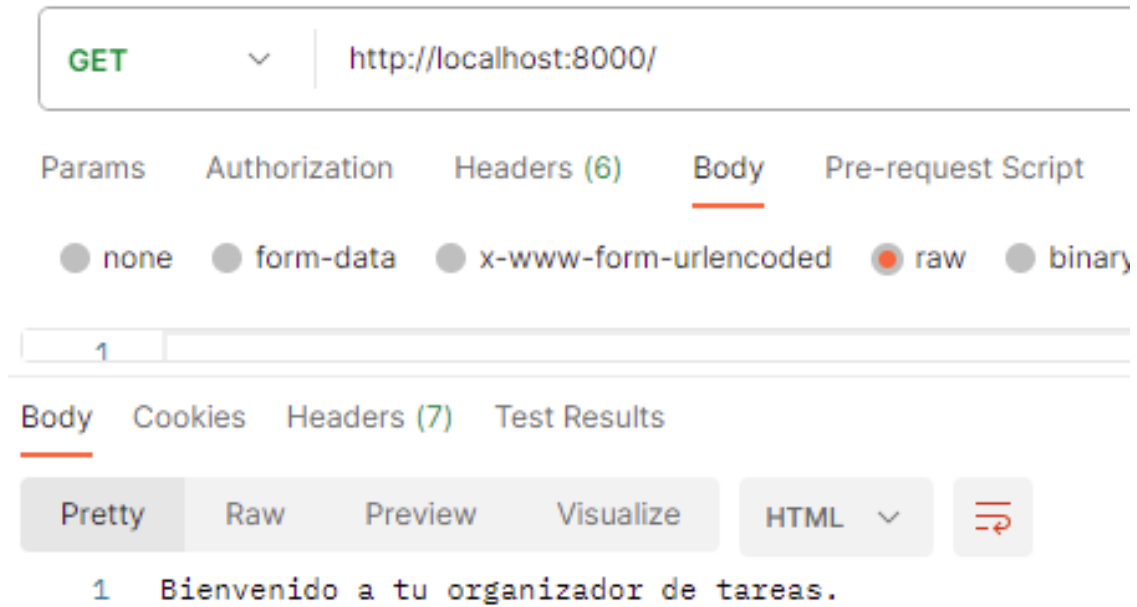


Organizador de tareas (Mongo Atlas)

El proyecto cuenta con varias funcionalidades, de acuerdo a los endpoints que se utilicen.

En la raíz ("/") desde el CRUD get, se muestra una bienvenida.



```
app.get('/', (req, res)=> {  
  res.send('Bienvenido a tu organizador de tareas.');
```

Para crear una tarea, se utiliza el endpoint a través del CRUD post: “/CreateTask”. Se debe indicar un título (“title”) y una descripción (“description”). La tarea se guarda por defecto como incompleta (“false”).

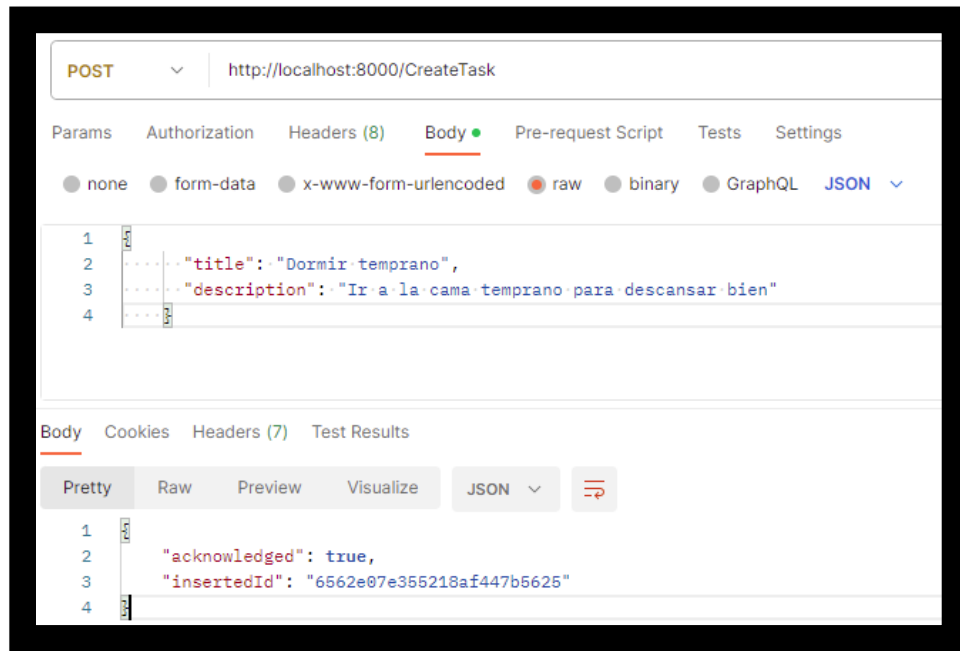


Imagen de postman.

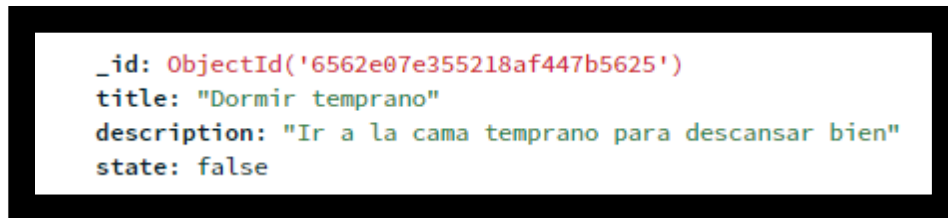


Imagen de Mongo Atlas.

```
app.post("/CreateTask", async (req, res) => {
  try {
    const { title } = req.body;
    const { description } = req.body;
    const newTask = {
      title,
      description,
      state: false,
    };
    await client.connect();
    const db = client.db("TasksList");
    const collection = db.collection("Tasks");
    const result = await collection.insertOne(newTask);
    res.send(result);
  } catch (error) {
    console.error(error);
    res.status(500).send({ error: "Failed to create task" });
  } finally {
    await client.close();
  }
});
```

Para obtener la lista completa de tareas, se puede acceder desde el CRUD get al endpoint “/TasksList”.

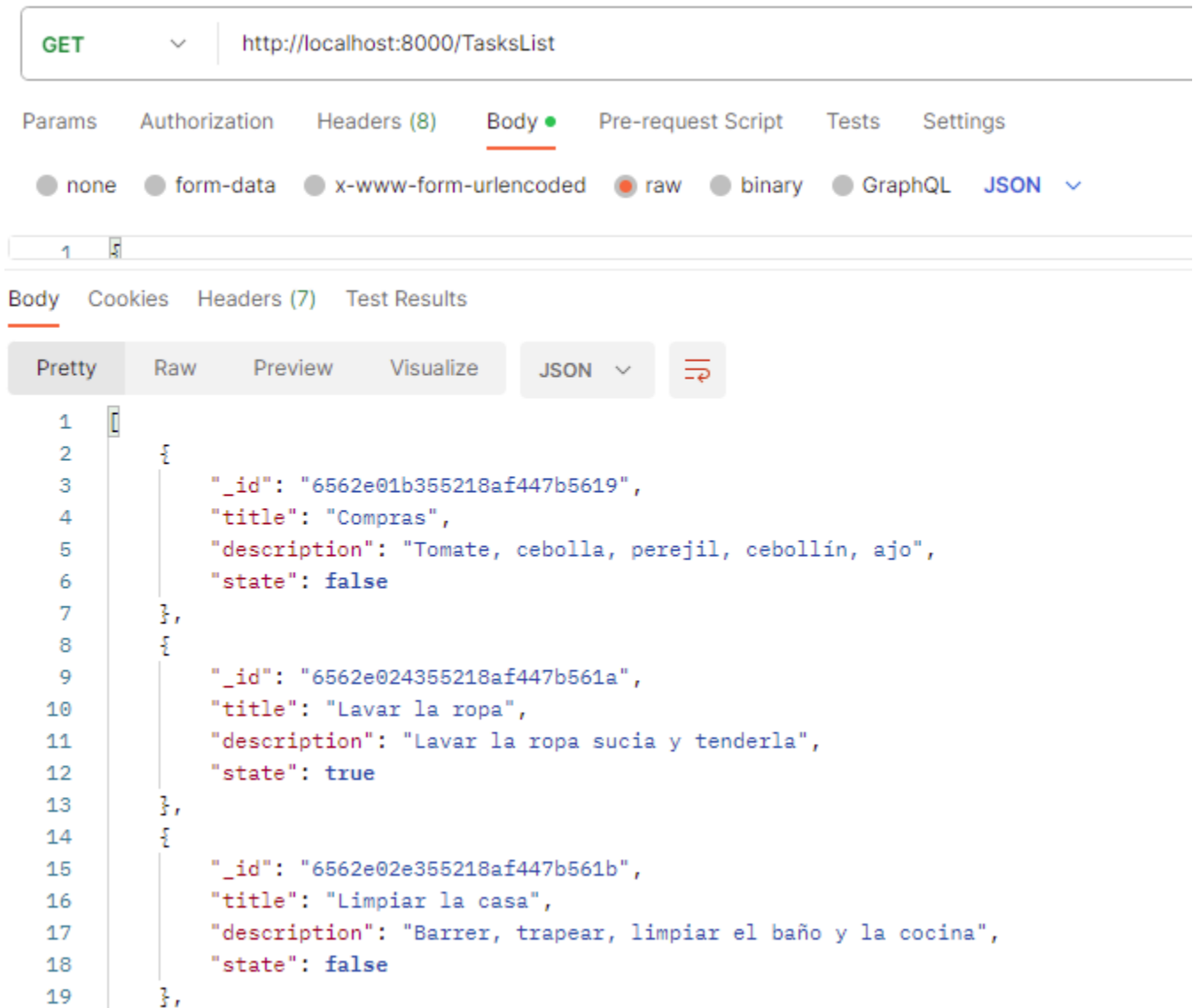


Imagen de postman.

```
app.get("/TasksList", async (req, res)=> {
  try {
    await client.connect()
    const db = client.db('TasksList')
    const collection = db.collection("Tasks")
    const result = await collection.find().toArray();
    console.log(result)
    res.send(result)
  }catch(error){
    return error;
  }
});
```

También se puede filtrar la lista entre tareas completadas (“/TasksList/Complete”).

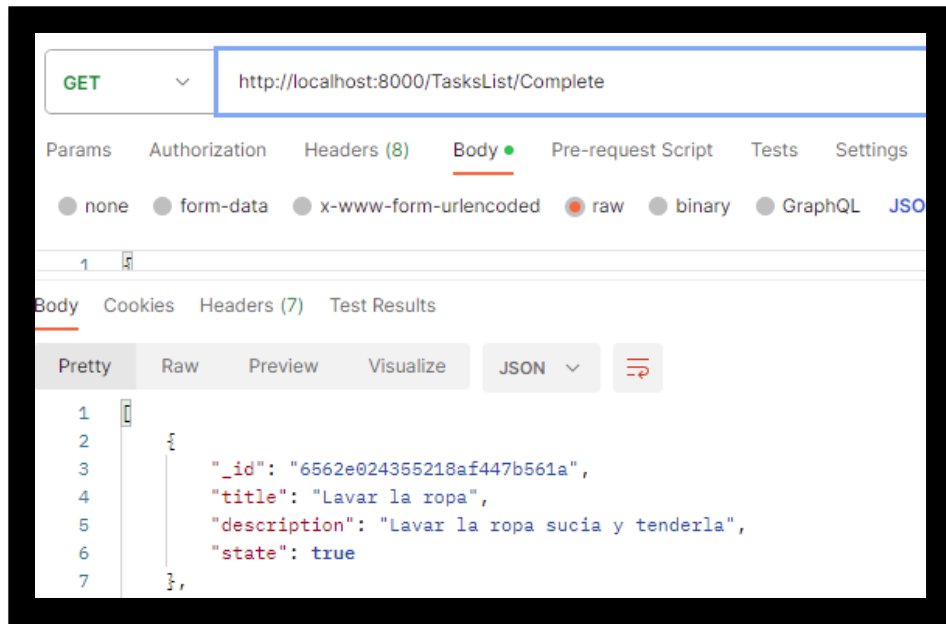


Imagen de postman.

```
app.get("/TasksList/Complete", async (req, res) => {
  try {
    await client.connect();
    const db = client.db("TasksList");
    const collection = db.collection("Tasks");
    const filter = { state: true };
    const result = await collection.find(filter).toArray();
    console.log(result);
    res.send(result);
  } catch (error) {
    return error;
  }
});
```

0 incompletas ("/TasksList/Incomplete").

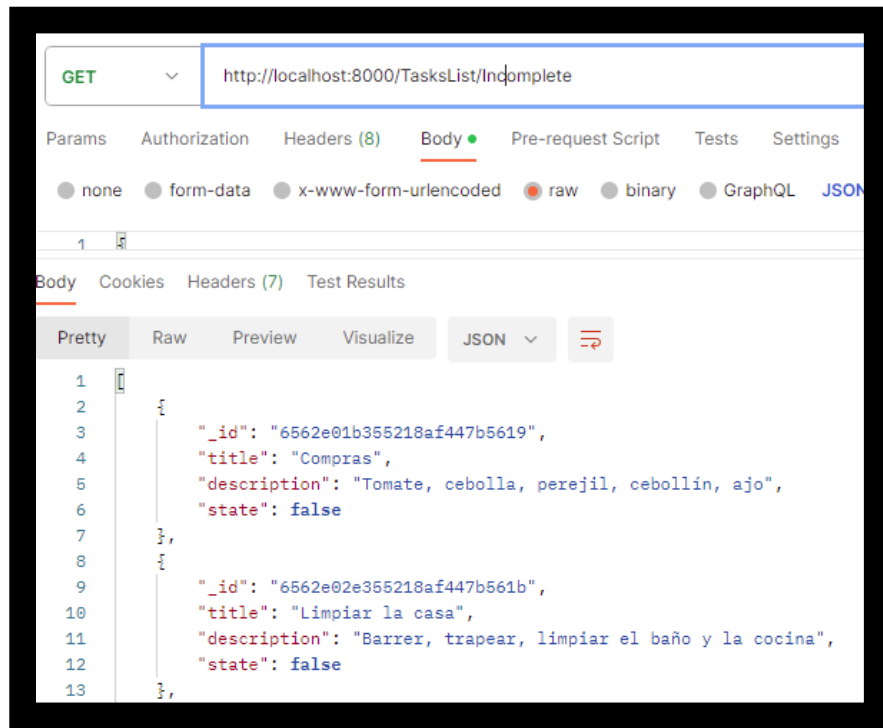


Imagen de postman.

```
app.get("/TasksList/Incomplete", async (req, res) => {
  try {
    await client.connect();
    const db = client.db("TasksList");
    const collection = db.collection("Tasks");
    const filter = { state: false };
    const result = await collection.find(filter).toArray();
    console.log(result);
    res.send(result);
  } catch (error) {
    return error;
  }
});
```

En caso de que se desee editar una tarea, se pueden utilizar los siguientes endpoints desde el CRUD put, teniendo en cuenta que “:id” debe ser reemplazado por el identificador de la tarea que se desea modificar:

```
_id: ObjectId('6562e01b355218af447b5619')
title: "Compras"
description: "Tomate, cebolla, perejil, cebollín, ajo"
state: false
```

Imagen de Mongo Atlas de la tarea que se va a editar.

Para editar el título, se utiliza el endpoint “/Update/Title/:id”. En el cuerpo de la solicitud, se debe indicar el nuevo título.

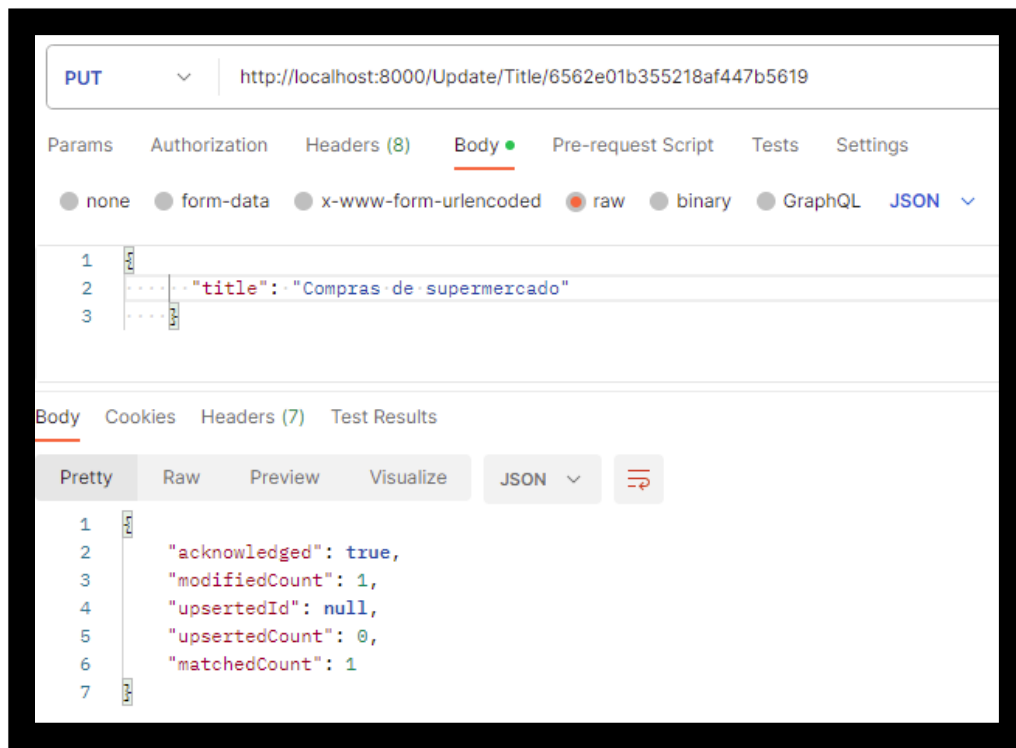


Imagen de postman.

```
app.put('/Update/Title/:id', async (req, res)=> {
  const id = req.params.id
  const {title} = req.body
  try{
    const db = client.db('TasksList');
    const collection = db.collection('Tasks')
    const result = await collection.updateOne({_id: new ObjectId(id)}, {$set:{title}})
    res.send(result)
  }catch(error){
    return error;
  }
});
```

Para editar la descripción, se utiliza el endpoint “/Update/Description/:id”. En el cuerpo de la solicitud, se debe indicar la nueva descripción.

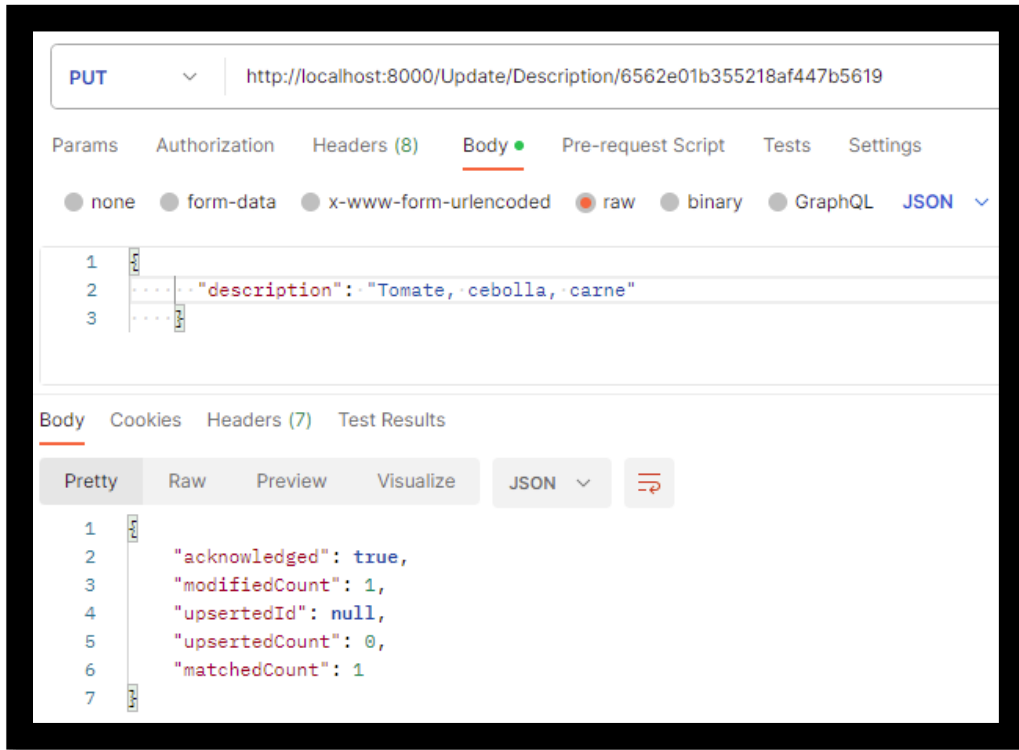


Imagen de postman.

```
app.put('/Update/Description/:id', async (req, res)=> {
  const id = req.params.id
  const {description} = req.body
  try{
    const db = client.db('TasksList');
    const collection = db.collection('Tasks')
    const result = await collection.updateOne({_id: new ObjectId(id)},
{$set:{description}})
    res.send(result)
  }catch(error){
    return error;
  }
});
```

Para editar el estado de la tarea, se utiliza el endpoint “/Update/State/:id”. En este caso, solo se necesita indicar el identificador de la tarea en la ruta. El estado de la tarea se invertirá.

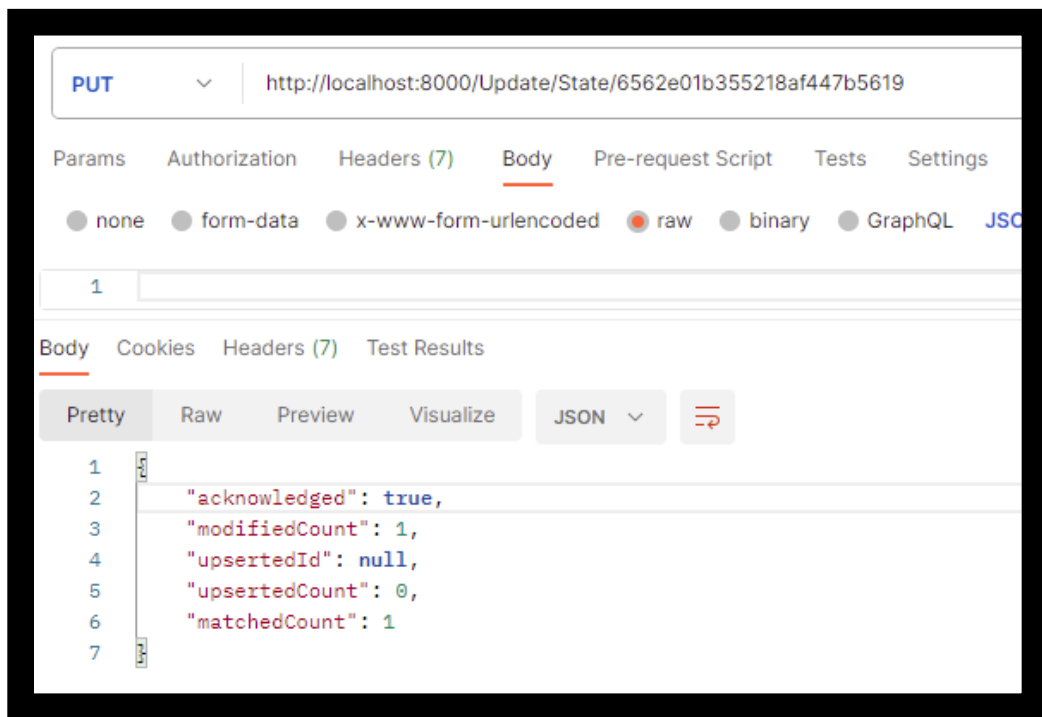


Imagen de postman.

```
app.put('/Update/State/:id', async (req, res) => {
  const id = req.params.id;

  try {
    const db = client.db('TasksList');
    const collection = db.collection('Tasks');
    const task = await collection.findOne({ _id: new ObjectId(id) });
    const currentState = task.state;
    const updatedState = !currentState;
    const result = await collection.updateOne({ _id: new ObjectId(id) }, { $set: { state:
updatedState } });

    res.send(result);
  } catch (error) {
    return error;
  }
});
```

```
_id: ObjectId('6562e01b355218af447b5619')
title: "Compras de supermercado"
description: "Tomate, cebolla, carne"
state: true
```

Imagen de Mongo Atlas de la tarea que se editó.

Para eliminar una tarea, desde delete el CRUD usamos el endpoint “/Delete/Task/:id”. El valor de “:id” debe ser reemplazado por el identificador de la tarea a eliminar.

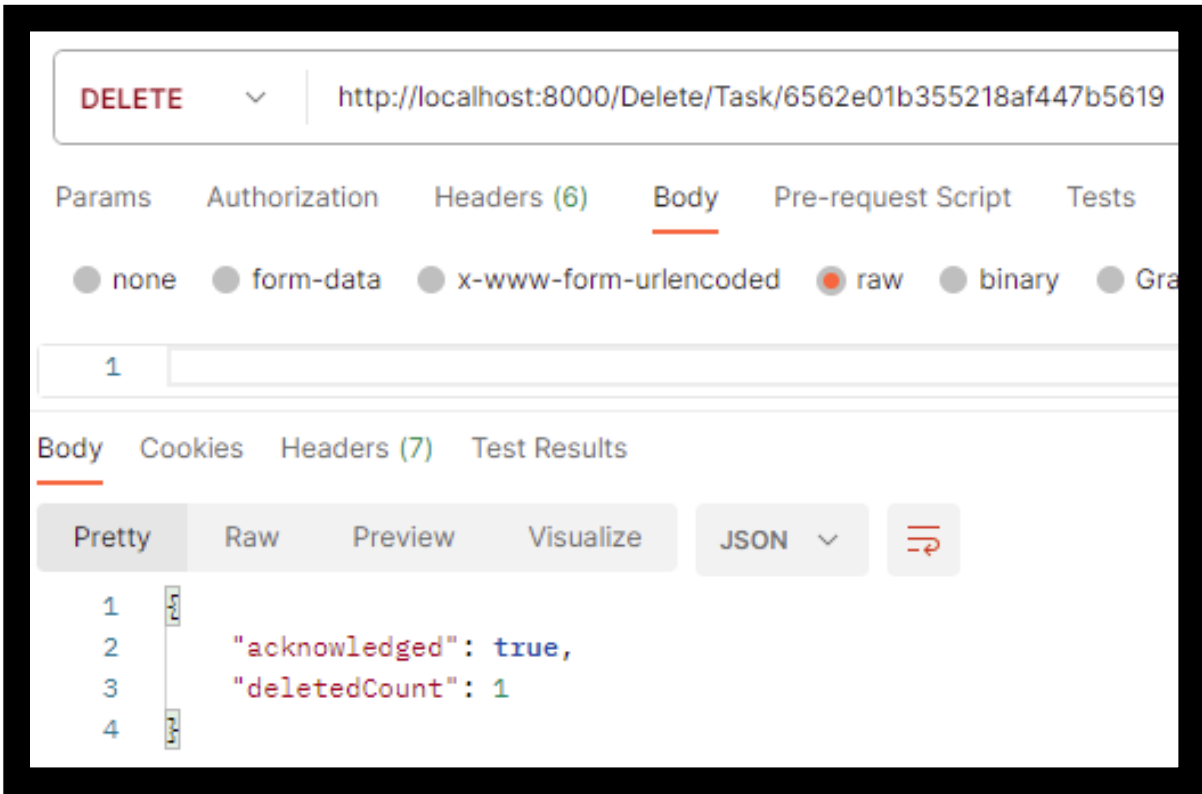


Imagen de postman.

```
app.delete('/Delete/Task/:id', async (req, res)=> {
  const id = req.params.id
  try{
    const db = client.db('TasksList');
    const collection = db.collection('Tasks')
    const result = await collection.deleteOne({_id: new ObjectId(id)})
    res.send(result)
  }catch(error){
    return error;
  }
});
```

Si se desea eliminar todas las tareas, se debe utilizar: “/Delete/AllTasks”.

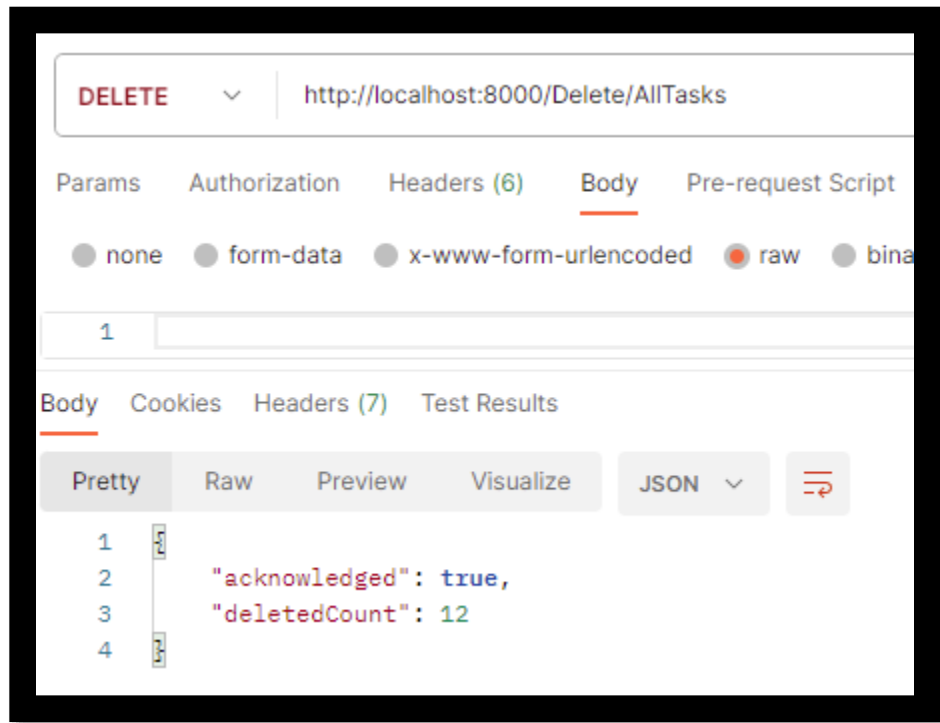


Imagen de postman.

```
app.delete('/Delete/AllTasks', async (req, res) => {  
  
  try {  
    const db = client.db('TasksList');  
    const collection = db.collection('Tasks');  
    const result = await collection.deleteMany({});  
  
    res.send(result);  
  } catch (error) {  
    return error;  
  }  
});
```