MONITORAMENTO AMBIENTAL

Smart Cities

MongoDB

Resumo

Um novo paradigma com NOT ONLY SQL - Utilizando NOSQL
Implementado em web.api.asp.NET

Wagner Machado – RM99307
Alianne Reis dos Santos – RM99937
Nathalia de Campos Nasz – RM98962
Adriano Reis Gama Monteiro - RM98133

Lista de Figuras

Sumário

Um novo paradigma com NOT ONLY SQL - Utilizando NOSQL

Monitoramento Ambiental – Smart Cities

Decisão de Migrar

A escolha de migrar o projeto para MongoDB é justificada pelas características do MongoDB que beneficiam uma aplicação para cidades inteligentes:

Alta disponibilidade:

MongoDB oferece suporte a replicação, garantindo que os dados estejam sempre disponíveis.

Escalabilidade

A capacidade de escalar horizontalmente permite gerenciar grandes volumes de dados.

Flexibilidade de esquema

Permite a adaptação rápida às mudanças nos requisitos de dados.

Justificativa para a Escolha de Cada Collection e seu Papel no Projeto

A seguir, são descritas as 5 collections escolhidas e seu papel no projeto:

Collection: Sensores

Justificativa

Armazena informações detalhadas sobre os sensores instalados nos parques da cidade. Sensores são elementos críticos em um sistema de monitoramento ambiental, pois são responsáveis por coletar dados necessários para a análise de condições ambientais.

Papel no Projeto

Fornece dados essenciais sobre a localização, tipo e código dos sensores, permitindo a identificação e o gerenciamento eficiente dos dispositivos de monitoramento.

Collection: Leituras

Justificativa

Armazena os dados coletados pelos sensores. Estes dados são cruciais para entender as condições ambientais em tempo real e para a análise histórica.

Papel no Projeto

Permite o armazenamento e recuperação das leituras dos sensores, associando cada leitura a um sensor específico, facilitando a análise dos dados coletados.

Collection: Alertas

Justificativa

Armazena alertas gerados com base nas leituras dos sensores. Alertas são importantes para notificar os usuários sobre condições ambientais anormais ou potencialmente perigosas.

Papel no Projeto

Permite o armazenamento e recuperação das leituras dos sensores, associando cada leitura a um sensor específico, facilitando a análise dos dados coletados.

Collection: PrevisoesChuva

Justificativa

Armazena previsões de chuva obtidas de fontes externas. A previsão de chuva é fundamental para o planejamento e gestão de atividades relacionadas à água.

Papel no Projeto

Auxilia na tomada de decisões informadas sobre o controle de irrigação e outras atividades que dependem das condições meteorológicas.

Collection: ControleIrrigacoes

Justificativa

Armazena informações sobre o controle de sistemas de irrigação automatizada. O controle eficiente da irrigação é essencial para a conservação de água e a manutenção da saúde das plantas em áreas verdes urbanas.

Papel no Projeto

Permite o controle automatizado e otimizado da irrigação com base em previsões de chuva e outras condições ambientais, garantindo o uso eficiente dos recursos hídricos.

Comandos para criação das collections e todas as operações CRUD

Criação das Collections

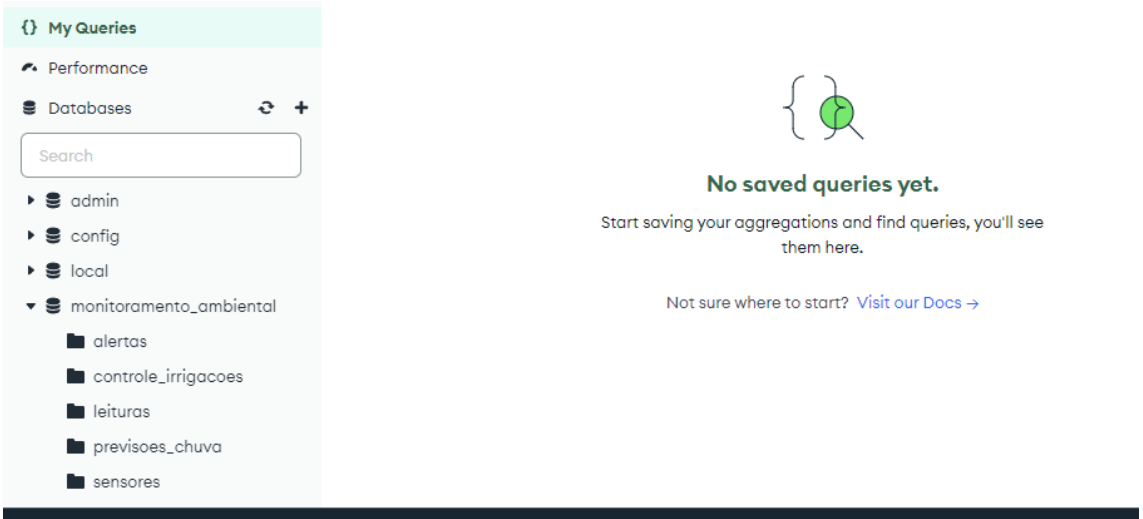
Primeiro Conectar ao MongoDB

use monitoramento_ambiental;

```
>_MONGOSH
> use monitoramento_ambiental;
< switched to db monitoramento_ambiental
monitoramento_ambiental> db.createCollection("alertas");
                         db.createCollection("sensores");
                         db.createCollection("leituras");
                         db.createCollection("controle_irrigacoes");
                         db.createCollection("previsoes_chuva");
```

Figura 1

Segundo Criar as Collections




```
>_MONGOSH
> use monitoramento_ambiental;
< switched to db monitoramento_ambiental
> db.createCollection("alertas");
  db.createCollection("sensores");
  db.createCollection("leituras");
  db.createCollection("controle_irrigacoes");
  db.createCollection("previsoes_chuva");
< { ok: 1 }
monitoramento_ambiental>
```

Figura 2

CRUD – Documentos na Collection Alertas

InsertMany

```
> db.alertas.insertMany([
    {
        tipo_alerta: "Alerta de Qualidade do Ar",
        descricao: "Poluição do ar acima do limite recomendado",
        localizacao: "Parque Ibirapuera",
        data_hora: new ISODate("2024-08-10T16:54:37.146Z")
    },
    {
        tipo_alerta: "Alerta de Umidade do Solo",
        descricao: "Umidade do solo abaixo do limite",
        localizacao: "Parque Villa-Lobos",
        data_hora: new ISODate("2024-08-10T17:00:00.000Z")
    }
]);
< {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('66b7dd43f1c061b2b27b66f2'),
    '1': ObjectId('66b7dd43f1c061b2b27b66f3')
  }
}
monitoramento_ambiental>
```

Figura 3

Find

```
> db.alertas.find({}).pretty()
< {
  _id: ObjectId('66b7dd43f1c061b2b27b66f2'),
  tipo_alerta: 'Alerta de Qualidade do Ar',
  descricao: 'Poluição do ar acima do limite recomendado',
  localizacao: 'Parque Ibirapuera',
  data_hora: 2024-08-10T16:54:37.146Z
}
{
  _id: ObjectId('66b7dd43f1c061b2b27b66f3'),
  tipo_alerta: 'Alerta de Umidade do Solo',
  descricao: 'Umidade do solo abaixo do limite',
  localizacao: 'Parque Villa-Lobos',
  data_hora: 2024-08-10T17:00:00.000Z
}
monitoramento_ambiental>
```

Figura 4

FindOne

```
> db.alertas.findOne({ _id: ObjectId("66b7dd43f1c061b2b27b66f2") })
< {
  _id: ObjectId('66b7dd43f1c061b2b27b66f2'),
  tipo_alerta: 'Alerta de Qualidade do Ar',
  descricao: 'Poluição do ar acima do limite recomendado',
  localizacao: 'Parque Ibirapuera',
  data_hora: 2024-08-10T16:54:37.146Z
}
monitoramento_ambiental>
```

Figura 5

UpdateOne

```
> db.alertas.updateOne(
      { _id: ObjectId("66b7dd43f1c061b2b27b66f2") },
      {
          $set: {
              tipo_alerta: "Alerta de Temperatura",
              descricao: "Temperatura elevada",
              localizacao: "Parque Ibirapuera",
              data_hora: new ISODate("2024-08-11T13:39:47Z")
          }
      }
  )
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0
  }
monitoramento_ambiental>
```

Figura 6

DeleteOne

```
> db.alertas.deleteOne({ _id: ObjectId("66b7dd43f1c061b2b27b66f3") })
< {
    acknowledged: true,
    deletedCount: 1
  }
monitoramento_ambiental>
```

Figura 7

CRUD – Documentos na Collection Sensores

InsertMany

```
> db.sensores.insertMany([
      {
          cod_sensor: 1,
          tipo_sensor: "Sensor de Temperatura",
          localizacao: "Parque Villa-Lobos"
      },
      {
          cod_sensor: 2,
          tipo_sensor: "Sensor de Umidade do Solo",
          localizacao: "Parque Ibirapuera"
      }
  ]);
< {
    acknowledged: true,
    insertedIds: {
      '0': ObjectId('66b7ddeaf1c061b2b27b66f4'),
      '1': ObjectId('66b7ddeaf1c061b2b27b66f5')
    }
  }
monitoramento_ambiental>
```

Figura 8

Find

```
> db.sensores.find({}).pretty()
< {
    _id: ObjectId('66b7ddeaf1c061b2b27b66f4'),
    cod_sensor: 1,
    tipo_sensor: 'Sensor de Temperatura',
    localizacao: 'Parque Villa-Lobos'
  }
  {
    _id: ObjectId('66b7ddeaf1c061b2b27b66f5'),
    cod_sensor: 2,
    tipo_sensor: 'Sensor de Umidade do Solo',
    localizacao: 'Parque Ibirapuera'
  }
monitoramento_ambiental>
```

Figura 9

FindOne

```
> db.sensores.findOne({ _id: ObjectId("66b7ddeaf1c061b2b27b66f5") })
< {
    _id: ObjectId('66b7ddeaf1c061b2b27b66f5'),
    cod_sensor: 2,
    tipo_sensor: 'Sensor de Umidade do Solo',
    localizacao: 'Parque Ibirapuera'
  }
monitoramento_ambiental>
```

Figura 10

UpdateOne

```
> db.sensores.updateOne(
      { _id: ObjectId("66b7ddeaf1c061b2b27b66f5") },
      {
          $set: {
              cod_sensor: 2,
              tipo_sensor: "Sensor de Umidade do Solo",
              localizacao: "Parque Villa-Lobos"
          }
      }
  )
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0
  }
monitoramento_ambiental>
```

Figura 11

DeleteOne

```
> db.sensores.deleteOne({ _id: ObjectId("66b7ddeaf1c061b2b27b66f5") })
< {
    acknowledged: true,
    deletedCount: 1
  }
monitoramento_ambiental>
```

Figura 12

CRUD – Documentos na Collection Leituras

InsertMany

```
> db.leituras.insertMany([
    {
        valor: 25.6,
        data_hora: new ISODate("2024-08-10T16:54:37.146Z"),
        cod_sensor: 1 // Obs: inserir um código sensor válido
    },
    {
        valor: 15.2,
        data_hora: new ISODate("2024-08-10T17:00:00.000Z"),
        cod_sensor: 2 // obs: inserir um código sensor válido
    }
]);
< {
    acknowledged: true,
    insertedIds: {
      '0': ObjectId('66b7de63f1c061b2b27b66f6'),
      '1': ObjectId('66b7de63f1c061b2b27b66f7')
    }
  }
monitoramento_ambiental>
```

Figura 13

Find

```
> db.leituras.find({}).pretty()
< {
    _id: ObjectId('66b7de63f1c061b2b27b66f6'),
    valor: 25.6,
    data_hora: 2024-08-10T16:54:37.146Z,
    cod_sensor: 1
  }
  {
    _id: ObjectId('66b7de63f1c061b2b27b66f7'),
    valor: 15.2,
    data_hora: 2024-08-10T17:00:00.000Z,
    cod_sensor: 2
  }
monitoramento_ambiental>
```

Figura 14

FindOne

```
> db.leituras.findOne({ _id: ObjectId("66b7de63f1c061b2b27b66f6") })
< {
    _id: ObjectId('66b7de63f1c061b2b27b66f6'),
    valor: 25.6,
    data_hora: 2024-08-10T16:54:37.146Z,
    cod_sensor: 1
  }
monitoramento_ambiental>
```

Figura 15

UpdateOne

```
> db.leituras.updateOne(
      { _id: ObjectId("66b7de63f1c061b2b27b66f7") },
      {
          $set: {
              valor: 25.6,
              data_hora: new ISODate("2024-08-10T17:00:00Z"),
              cod_sensor: 2
          }
      }
  )
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0
  }
monitoramento_ambiental>
```

Figura 16

DeleteOne

```
> db.leituras.deleteOne({ _id: ObjectId("66b7de63f1c061b2b27b66f6") })
< {
    acknowledged: true,
    deletedCount: 1
  }
monitoramento_ambiental>
```

Figura 17

CRUD – Documentos na Collection Previsões de Chuva

InsertMany

```
>_MONGOSH
> db.previsoes_chuva.insertMany([
      {
          data_hora: new ISODate("2024-08-10T16:54:37.146Z"),
          previsao_chuva: "Sem Chuva"
      },
      {
          data_hora: new ISODate("2024-08-10T17:00:00.000Z"),
          previsao_chuva: "Chuva"
      }
  ]);
< {
    acknowledged: true,
    insertedIds: {
      '0': ObjectId('66b7dee0f1c061b2b27b66f8'),
      '1': ObjectId('66b7dee0f1c061b2b27b66f9')
    }
  }
```

Figura 18

Find

```
> db.previsoes_chuva.find({}).pretty()
< {
    _id: ObjectId('66b7dee0f1c061b2b27b66f8'),
    data_hora: 2024-08-10T16:54:37.146Z,
    previsao_chuva: 'Sem Chuva'
  }
  {
    _id: ObjectId('66b7dee0f1c061b2b27b66f9'),
    data_hora: 2024-08-10T17:00:00.000Z,
    previsao_chuva: 'Chuva'
  }
monitoramento_ambiental>
```

Figura 19

FindOne

```
> db.previsoes_chuva.findOne({ _id: ObjectId("66b7dee0f1c061b2b27b66f8") })
< {
    _id: ObjectId('66b7dee0f1c061b2b27b66f8'),
    data_hora: 2024-08-10T16:54:37.146Z,
    previsao_chuva: 'Sem Chuva'
  }
monitoramento_ambiental>
```

Figura 20

UpdateOne

```
> db.previsoes_chuva.updateOne(
      { _id: ObjectId("66b7dee0f1c061b2b27b66f8") },
      {
          $set: {
              data_hora: new ISODate("2024-08-11T16:54:37Z"),
              previsao: "Sem Chuva"
          }
      }
  )
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0
  }
monitoramento_ambiental>
```

Figura 21

DeleteOne

```
> db.previsoes_chuva.deleteOne({ _id: ObjectId("66b7dee0f1c061b2b27b66f9") })
< {
    acknowledged: true,
    deletedCount: 1
  }
monitoramento_ambiental>
```

Figura 22

CRUD – Inserindo Documentos na Collection Controle Irrigações

InsertMany

```
>_MONGOSH
> db.controle_irrigacoes.insertMany([
    {
        localizacao: "Parque Ibirapuera",
        estado: "Ligado",
        data_hora: new ISODate("2024-08-10T16:54:37.146Z"),
        previsao_chuva_id: "63b6f54d8d95b97e3f8b4567",  // obs: inserir um ObjectId válido
        previsao_chuva: {
            data_hora: new ISODate("2024-08-10T16:54:37.146Z"),
            previsao: "Sem Chuva"
        }
    },
    {
        localizacao: "Parque Villa-Lobos",
        estado: "Desligado",
        data_hora: new ISODate("2024-08-10T17:00:00.000Z"),
        previsao_chuva_id: "63b6f54d8d95b97e3f8b4568",  // obs: inserir um ObjectId válido
        previsao_chuva: {
            data_hora: new ISODate("2024-08-10T17:00:00.000Z"),
            previsao: "Chuva"
        }
    }
]);
< {
    acknowledged: true,
    insertedIds: {
      '0': ObjectId('66b7df6cf1c061b2b27b66fa'),
      '1': ObjectId('66b7df6cf1c061b2b27b66fb')
    }
  }
monitoramento_ambiental>
```

Figura 23

Find

```
> db.controle_irrigacoes.find({}).pretty()
< {
    _id: ObjectId('66b7df6cf1c061b2b27b66fa'),
    localizacao: 'Parque Ibirapuera',
    estado: 'Ligado',
    data_hora: 2024-08-10T16:54:37.146Z,
    previsao_chuva_id: '63b6f54d8d95b97e3f8b4567',
    previsao_chuva: {
      data_hora: 2024-08-10T16:54:37.146Z,
      previsao: 'Sem Chuva'
    }
  }
  {
    _id: ObjectId('66b7df6cf1c061b2b27b66fb'),
    localizacao: 'Parque Villa-Lobos',
    estado: 'Desligado',
    data_hora: 2024-08-10T17:00:00.000Z,
    previsao_chuva_id: '63b6f54d8d95b97e3f8b4568',
    previsao_chuva: {
      data_hora: 2024-08-10T17:00:00.000Z,
      previsao: 'Chuva'
    }
  }
monitoramento_ambiental>
```

Figura 24

FindOne

```
> db.controle_irrigacoes.findOne({ _id: ObjectId("66b7df6cf1c061b2b27b66fa") })
< {
    _id: ObjectId('66b7df6cf1c061b2b27b66fa'),
    localizacao: 'Parque Ibirapuera',
    estado: 'Ligado',
    data_hora: 2024-08-10T16:54:37.146Z,
    previsao_chuva_id: '63b6f54d8d95b97e3f8b4567',
    previsao_chuva: {
      data_hora: 2024-08-10T16:54:37.146Z,
      previsao: 'Sem Chuva'
    }
  }
monitoramento_ambiental>
```

Figura 25

UpdateOne

```
> db.controle_irrigacoes.updateOne(
      { _id: ObjectId("66b7df6cf1c061b2b27b66fa") },
      {
          $set: {
              localizacao: "Parque Ibirapuera",
              estado: "Desligado",
              data_hora: new ISODate("2024-08-10T16:54:37Z"),
              previsao_chuva_id: "63b6f54d8d95b97e3f8b4567"
          }
      }
  )
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0
  }
monitoramento_ambiental>
```

Figura 26

DeleteOne

```
> db.controle_irrigacoes.deleteOne({ _id: ObjectId("66b7df6cf1c061b2b27b66fb") })
< {
    acknowledged: true,
    deletedCount: 1
  }
monitoramento_ambiental>
```

Figura 27

Evidências das Collections criadas

DataBase e Collections criados no MongoDB

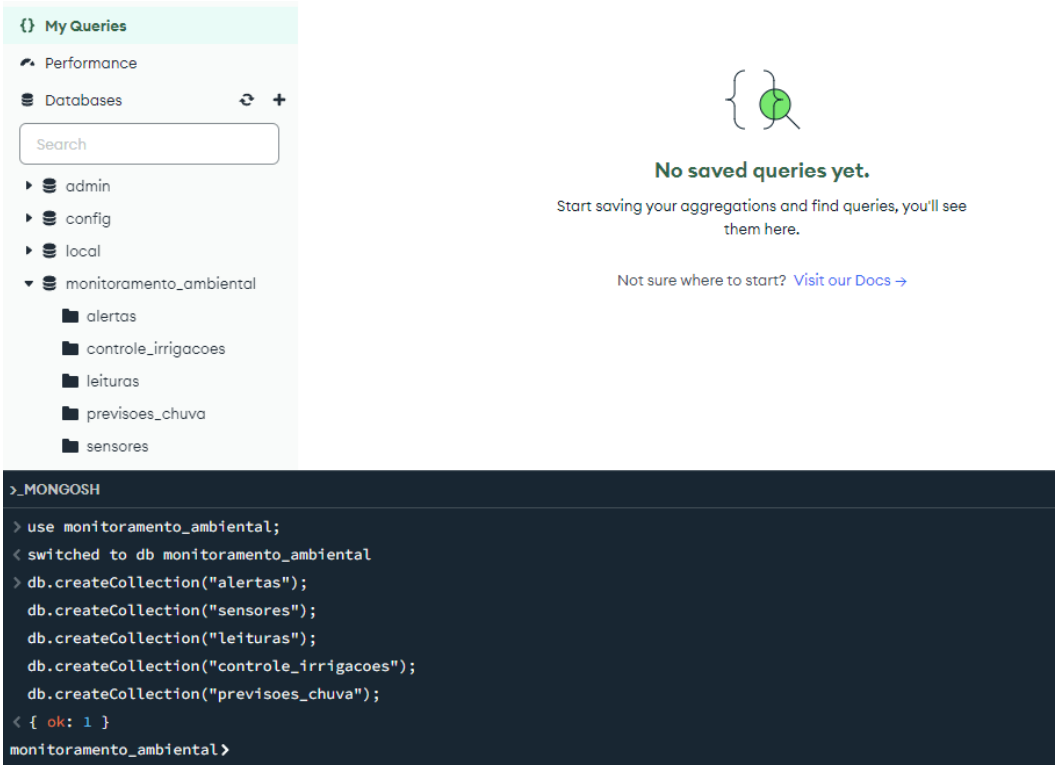


Figura 28

DataBase e Collections criados pela API Web Asp.NET

Configurações com MongoDB e Csharp e demais configurações das Services

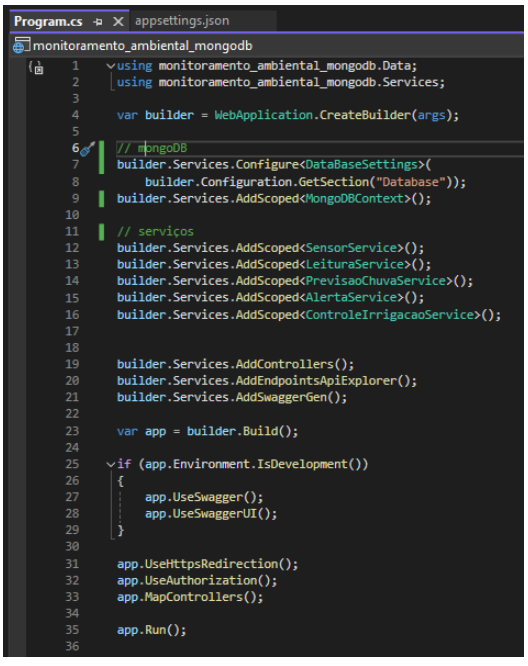


Figura 29

DataBase Context – Gerenciamento do DataBase e acesso as Collections

MongoDBContext.cs

monitoramento_ambiental_mongodb — monitoramento_ambiental_mongodb.Data.MongoDBContext

```
using Microsoft.Extensions.Options;
using MongoDB.Driver;
using monitoramento_ambiental_mongodb.Models;

namespace monitoramento_ambiental_mongodb.Data
{
    7 references
    public class MongoDBContext
    {
        private readonly IMongoDatabase _database;
        private readonly DataBaseSettings _settings;

        0 references
        public MongoDBContext(IOptions<DataBaseSettings> options)
        {
            _settings = options.Value;
            var client = new MongoClient(_settings.ConnectionURI);
            _database = client.GetDatabase(_settings.DatabaseName);
        }

        2 references
        public IMongoCollection<SensorModel> Sensores =>
            _database.GetCollection<SensorModel>(_settings.SensorCollectionName);

        1 reference
        public IMongoCollection<LeituraModel> Leituras =>
            _database.GetCollection<LeituraModel>(_settings.LeituraCollectionName);

        2 references
        public IMongoCollection<PrevisaoChuvaModel> PrevisoesChuva =>
            _database.GetCollection<PrevisaoChuvaModel>(_settings.PrevisaoChuvaCollectionName);

        1 reference
        public IMongoCollection<ControleIrrigacaoModel> ControleIrrigacoes =>
            _database.GetCollection<ControleIrrigacaoModel>(_settings.ControleIrrigacaoCollectionName);

        1 reference
        public IMongoCollection<AlertaModel> Alertas =>
            _database.GetCollection<AlertaModel>(_settings.AlertaCollectionName);
    }
}
```

Figura 30

DataBaseSettings

DataBaseSettings.cs | MongoDBContext.cs

monitoramento_ambiental_mongodb — monitoramento_ambiental_mongodb.Data.DataBaseSettings

```
namespace monitoramento_ambiental_mongodb.Data
{
    3 references
    public class DataBaseSettings
    {
        1 reference
        public string ConnectionURI { get; set; } = null!;
        1 reference
        public string DatabaseName { get; set; } = null!;
        1 reference
        public string SensorCollectionName { get; set; } = null!;
        1 reference
        public string LeituraCollectionName { get; set; } = null!;
        1 reference
        public string ControleIrrigacaoCollectionName { get; set; } = null!;
        1 reference
        public string AlertaCollectionName { get; set; } = null!;
        1 reference
        public string PrevisaoChuvaCollectionName { get; set; } = null!;

    }
}
```

Figura 31

Nome do DataBase e das Collections

appsettings.json

Schema: https://json.schemastore.org/appsettings.json

```
{
  "Database": {
    "ConnectionURI": "mongodb://localhost:27017",
    "DatabaseName": "monitoramento_ambiental",
    "SensorCollectionName": "sensores",
    "LeituraCollectionName": "leituras",
    "PrevisaoChuvaCollectionName": "previsoes_chuva",
    "ControleIrrigacaoCollectionName": "controle_irrigacoes",
    "AlertaCollectionName": "alertas"
  }
}
```

Figura 32

Models / Collections

Alerta

AlertaModel.cs

monitoramento_ambiental_mongodb

```
using MongoDB.Bson.Serialization.Attributes;
using MongoDB.Bson;

namespace monitoramento_ambiental_mongodb.Models
{
    12 references
    public class AlertaModel
    {
        [BsonId]
        [BsonRepresentation(BsonType.ObjectId)]
        [BsonElement("_id")]
        7 references
        public string? Id { get; set; }

        [BsonElement("tipo_alerta")]
        0 references
        public string? TipoAlerta { get; set; }

        [BsonElement("descricao")]
        0 references
        public string? Descricao { get; set; }

        [BsonElement("localizacao")]
        0 references
        public string? Localizacao { get; set; }

        [BsonElement("data_hora")]
        6 references
        public DateTime DataHora { get; set; }

    }
}
```

Figura 33

Sensor

SensorModel.cs

monitoramento_ambiental_mongodb

```
using MongoDB.Bson.Serialization.Attributes;
using MongoDB.Bson;

namespace monitoramento_ambiental_mongodb.Models
{
    14 references
    public class SensorModel
    {

        [BsonId]
        [BsonRepresentation(BsonType.ObjectId)]
        [BsonElement("_id")]
        6 references
        public string? Id { get; set; }

        [BsonElement("cod_sensor")]
        4 references
        public int CodSensor { get; set; }

        [BsonElement("tipo_sensor")]
        0 references
        public string? TipoSensor { get; set; } = null;

        [BsonElement("localizacao")]
        0 references
        public string? Localizacao { get; set; } = null;
    }

}
```

Figura 34

Previsão de Chuva

PrevisaoChuvaModel.cs

monitoramento_ambiental_mongodb

```
using MongoDB.Bson.Serialization.Attributes;
using MongoDB.Bson;

namespace monitoramento_ambiental_mongodb.Models
{
    14 references
    public class PrevisaoChuvaModel
    {
        [BsonId]
        [BsonRepresentation(BsonType.ObjectId)]
        11 references
        public string? Id { get; set; }

        [BsonElement("data_hora")]
        6 references
        public DateTime DataHora { get; set; }

        [BsonElement("previsao")]
        1 reference
        public string? PrevisaoChuva { get; set; }
    }

}
```

Figura 35

Leitura

LeituraModel.cs

monitoramento_ambiental_mongodb

```
using MongoDB.Bson.Serialization.Attributes;
using MongoDB.Bson;
using System.Text.Json.Serialization;
using Amazon.Auth.AccessControlPolicy;

namespace monitoramento_ambiental_mongodb.Models
{
    12 references
    public class LeituraModel
    {
        [BsonId]
        [BsonRepresentation(BsonType.ObjectId)]
        [BsonIgnoreIfDefault]
        [BsonElement("_id")]
        7 references
        public string? Id { get; set; }

        private decimal valor;

        [BsonElement("valor")]
        [BsonRepresentation(BsonType.Decimal128)]
        0 references
        public decimal Valor
        {
            get => valor;
            set => valor = Math.Round(value, 2);
        }

        [BsonElement("data_hora")]
        6 references
        public DateTime DataHora { get; set; }

        [BsonElement("cod_sensor")]
        2 references
        public int CodSensor { get; set; }

        2 references
        public SensorModel? Sensor { get; set; }
    }

}
```

Figura 36

Controle de Irrigação

ControleIrrigacaoModel.cs

monitoramento_ambiental_mongodb

```
using MongoDB.Bson.Serialization.Attributes;
using MongoDB.Bson;

namespace monitoramento_ambiental_mongodb.Models
{
    12 references
    public class ControleIrrigacaoModel
    {
        [BsonId]
        [BsonRepresentation(BsonType.ObjectId)]
        [BsonElement("_id")]
        5 references
        public string? Id { get; set; }

        [BsonElement("localizacao")]
        0 references
        public string? Localizacao { get; set; }

        [BsonElement("estado")]
        0 references
        public string? Estado { get; set; }

        [BsonElement("data_hora")]
        0 references
        public DateTime DataHora { get; set; }

        [BsonElement("previsao_chuva_id")]
        3 references
        public string PrevisaoChuvaId { get; set; }

        2 references
        public PrevisaoChuvaModel? PrevisaoChuva { get; set; }
    }

}
```

Figura 37

Evidencias da Manipulação dos Dados Via Swagger UI

Collection Sensores

Insert

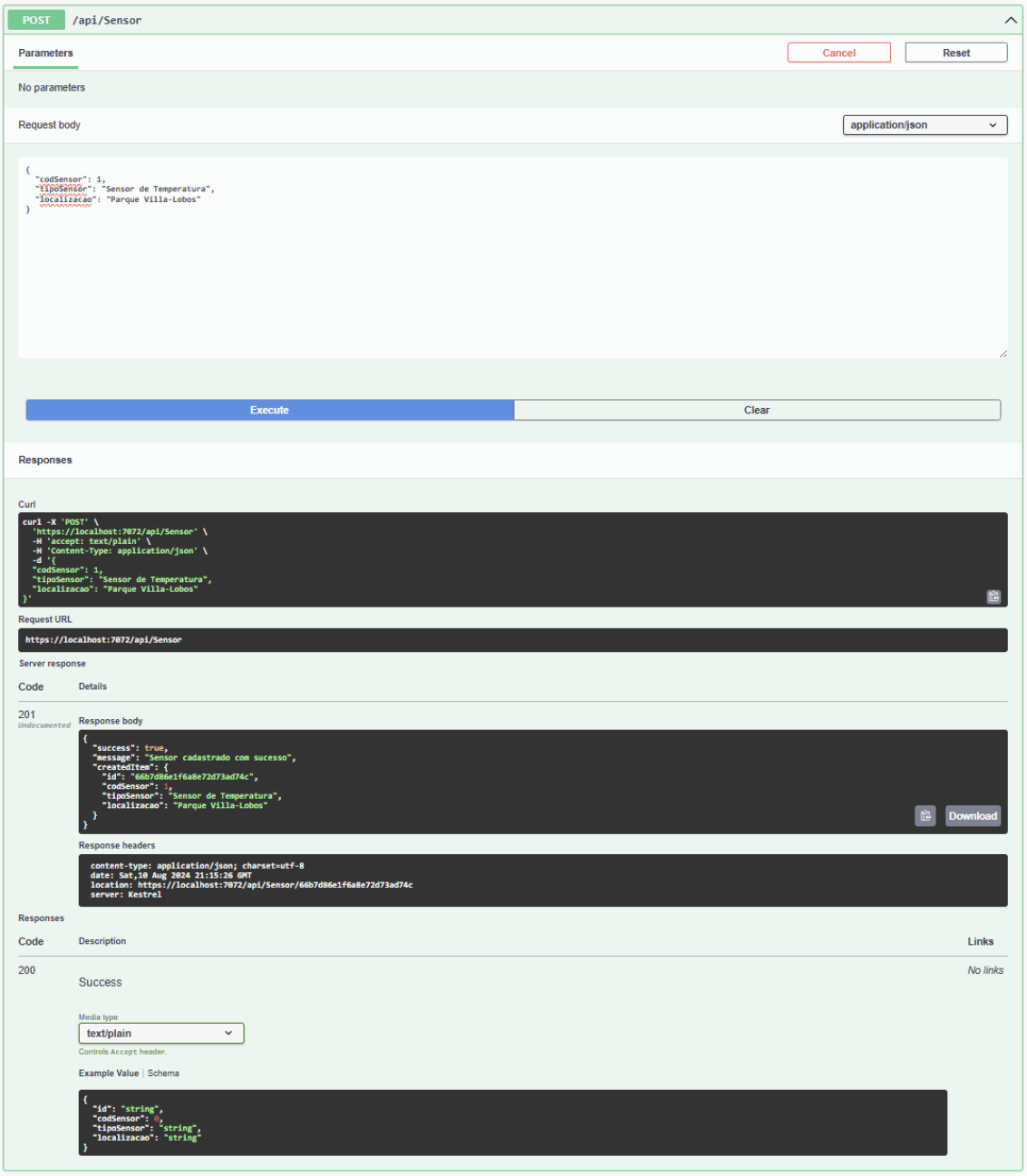


Figura 38

GetAll

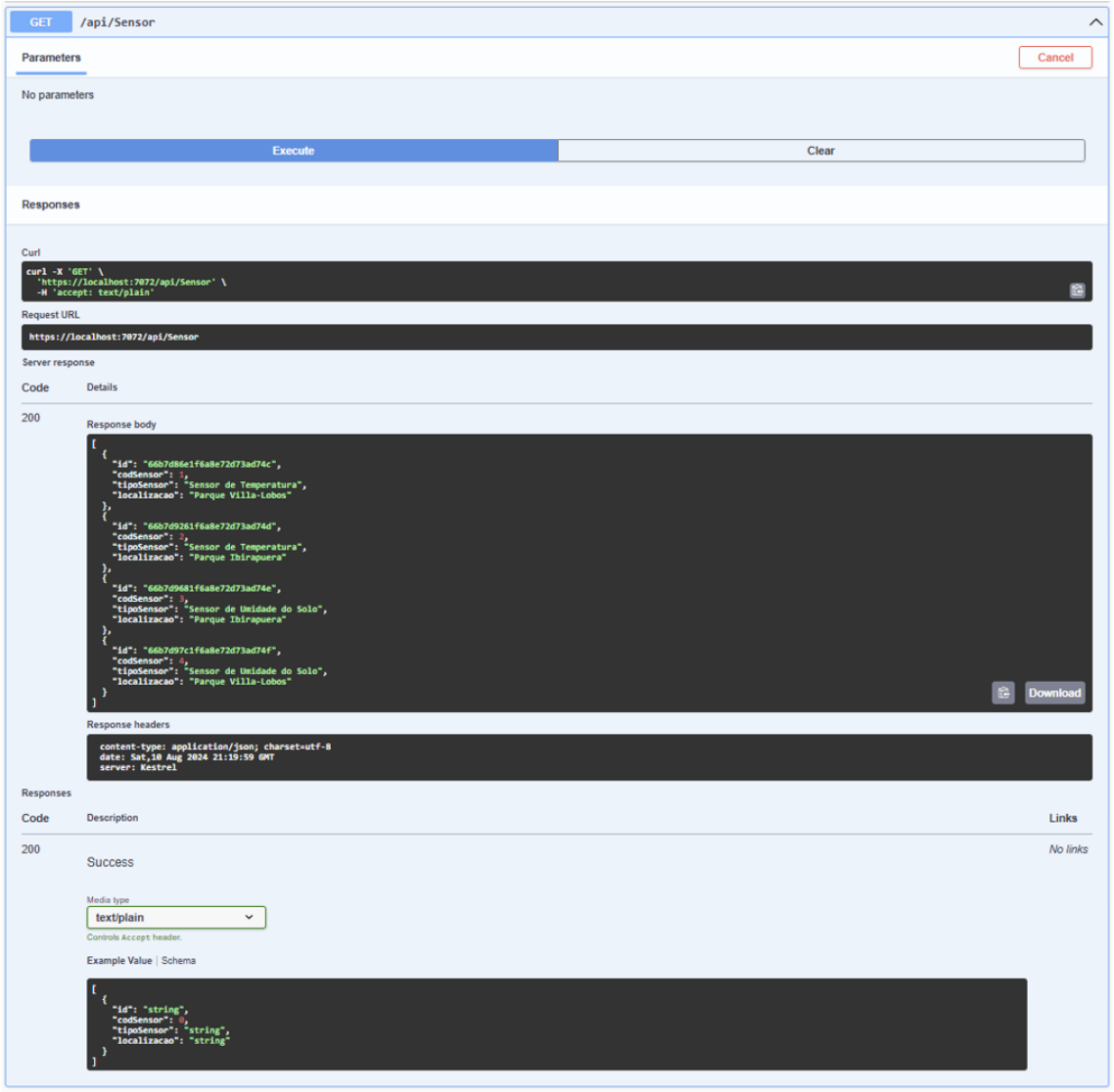


Figura 39

GetById

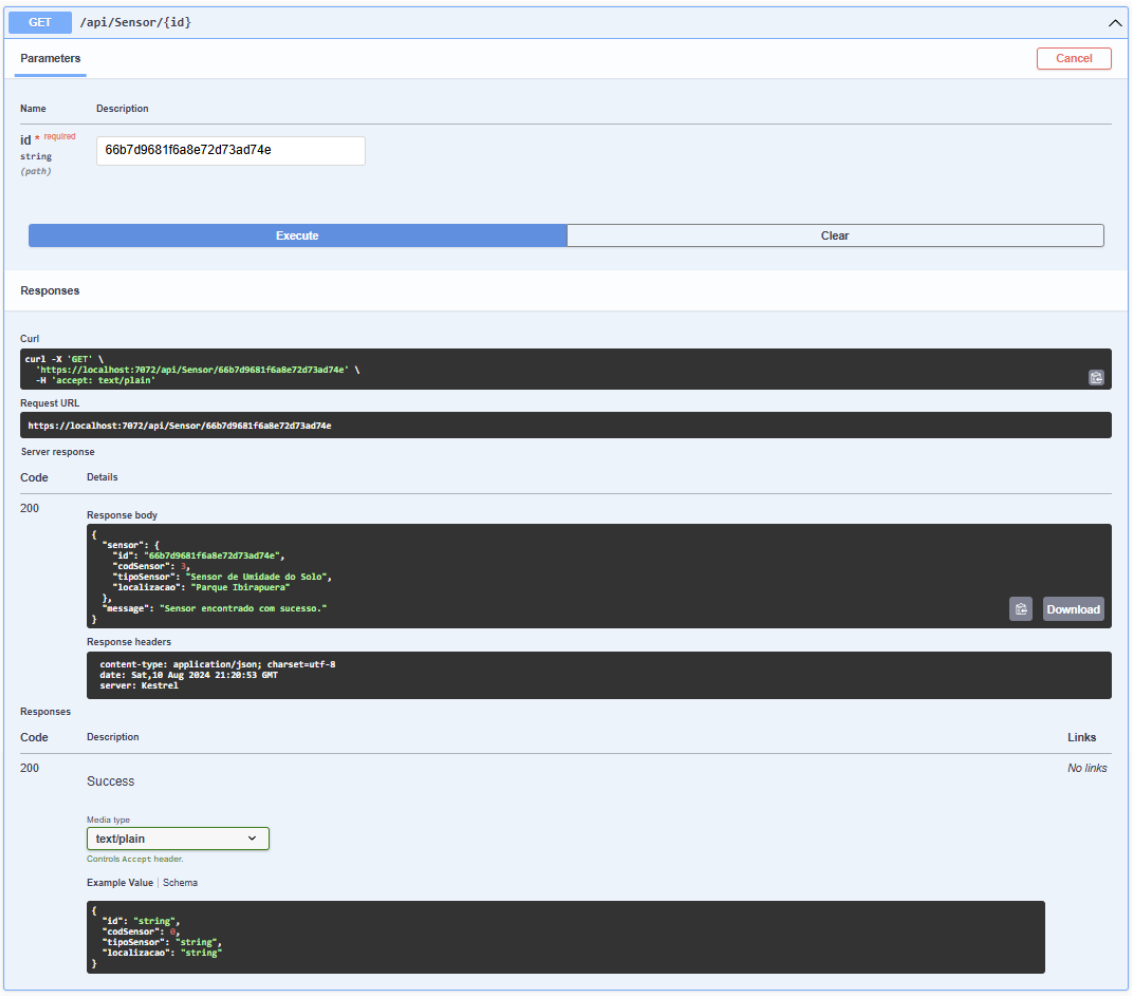


Figura 40

Update

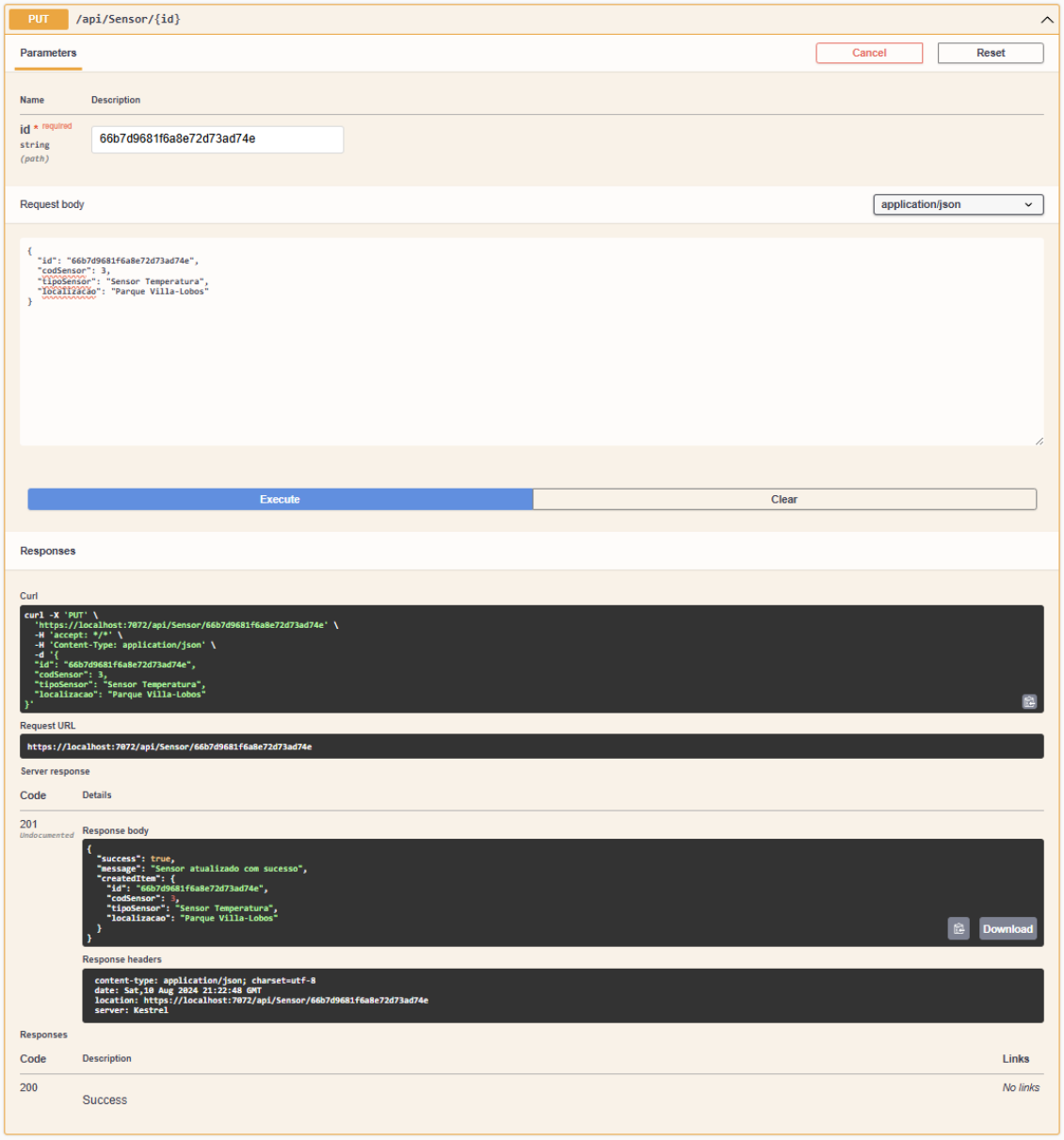


Figura 41

Delete

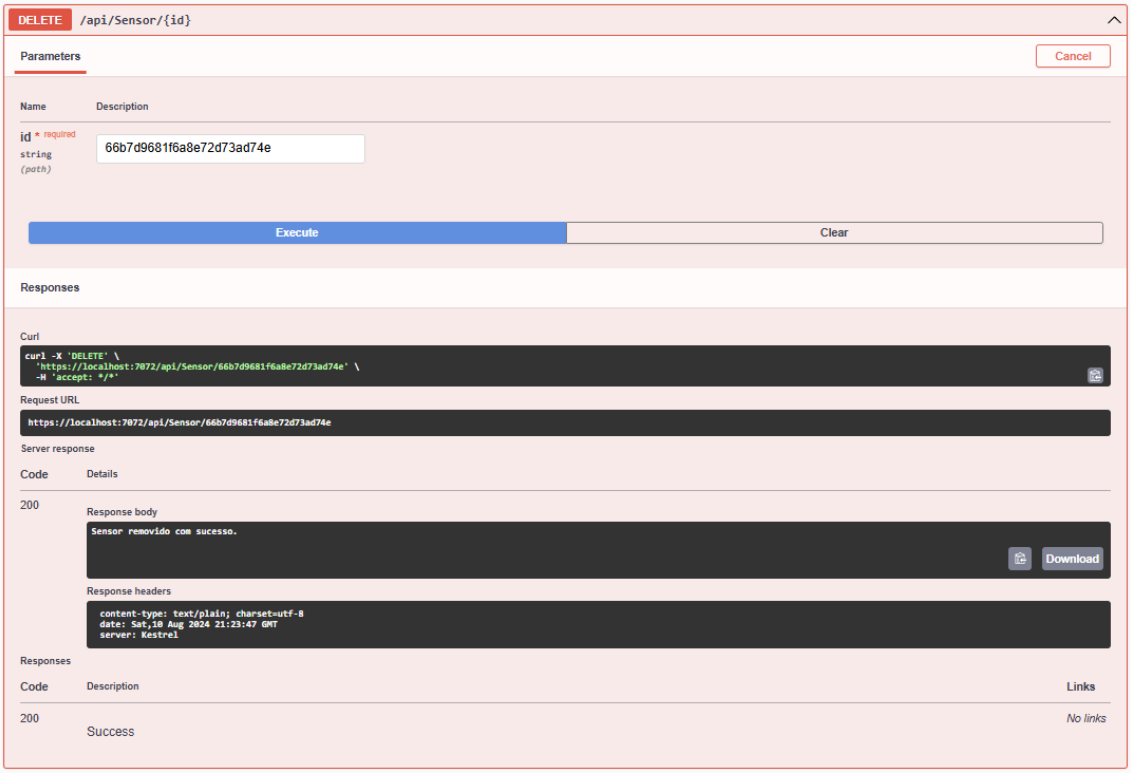


Figura 42

Collection Alertas

Insert

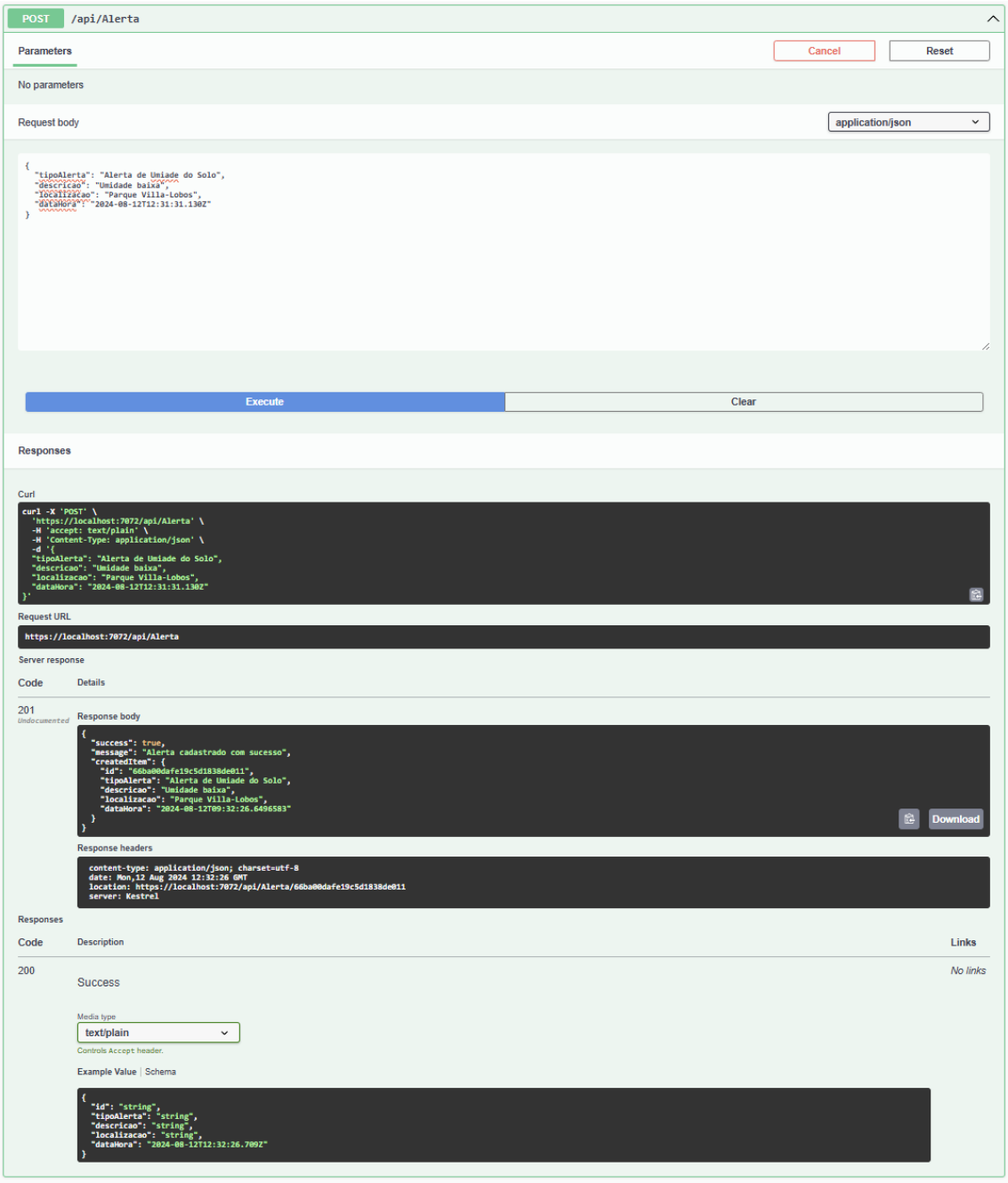


Figura 43

GetAll

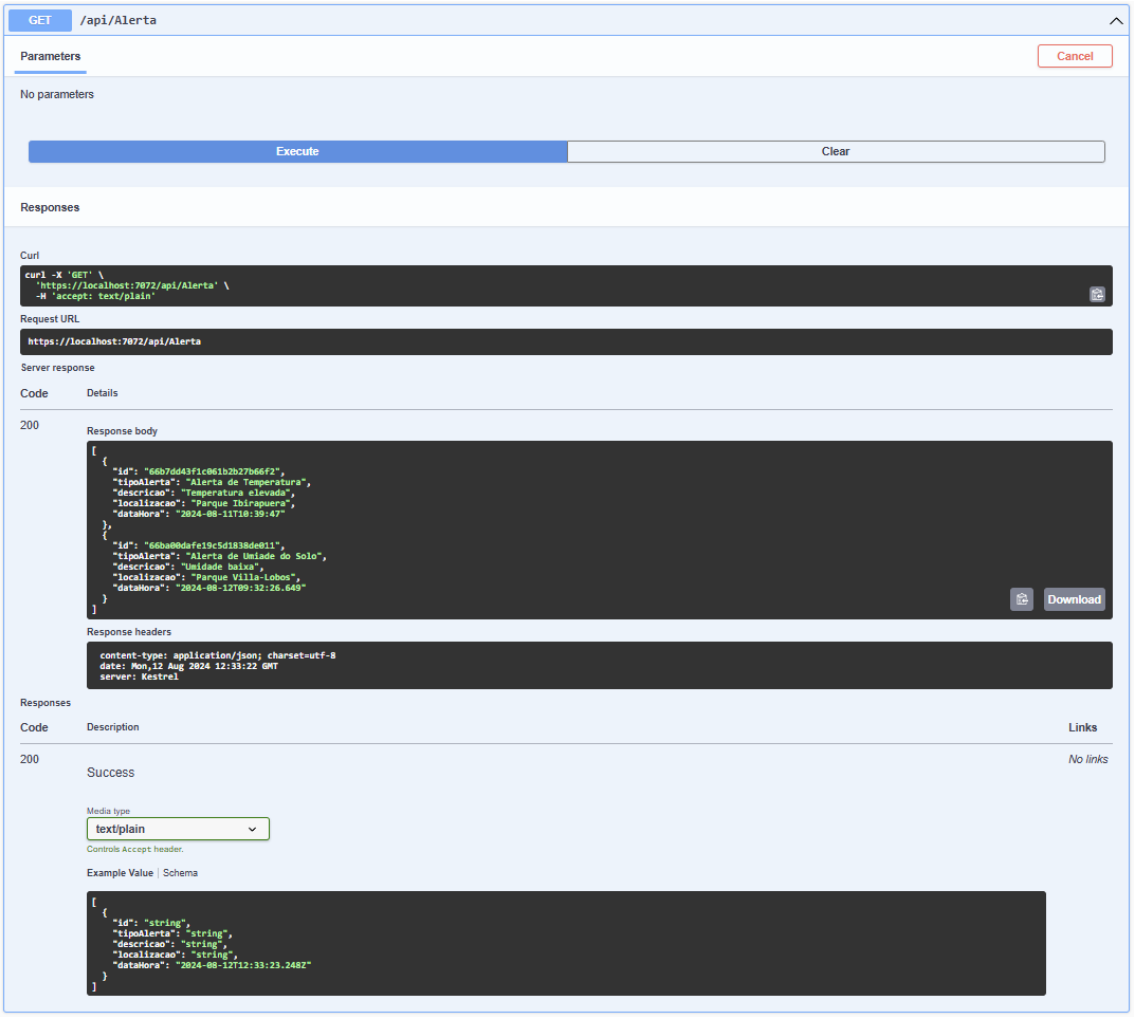


Figura 44

GetById

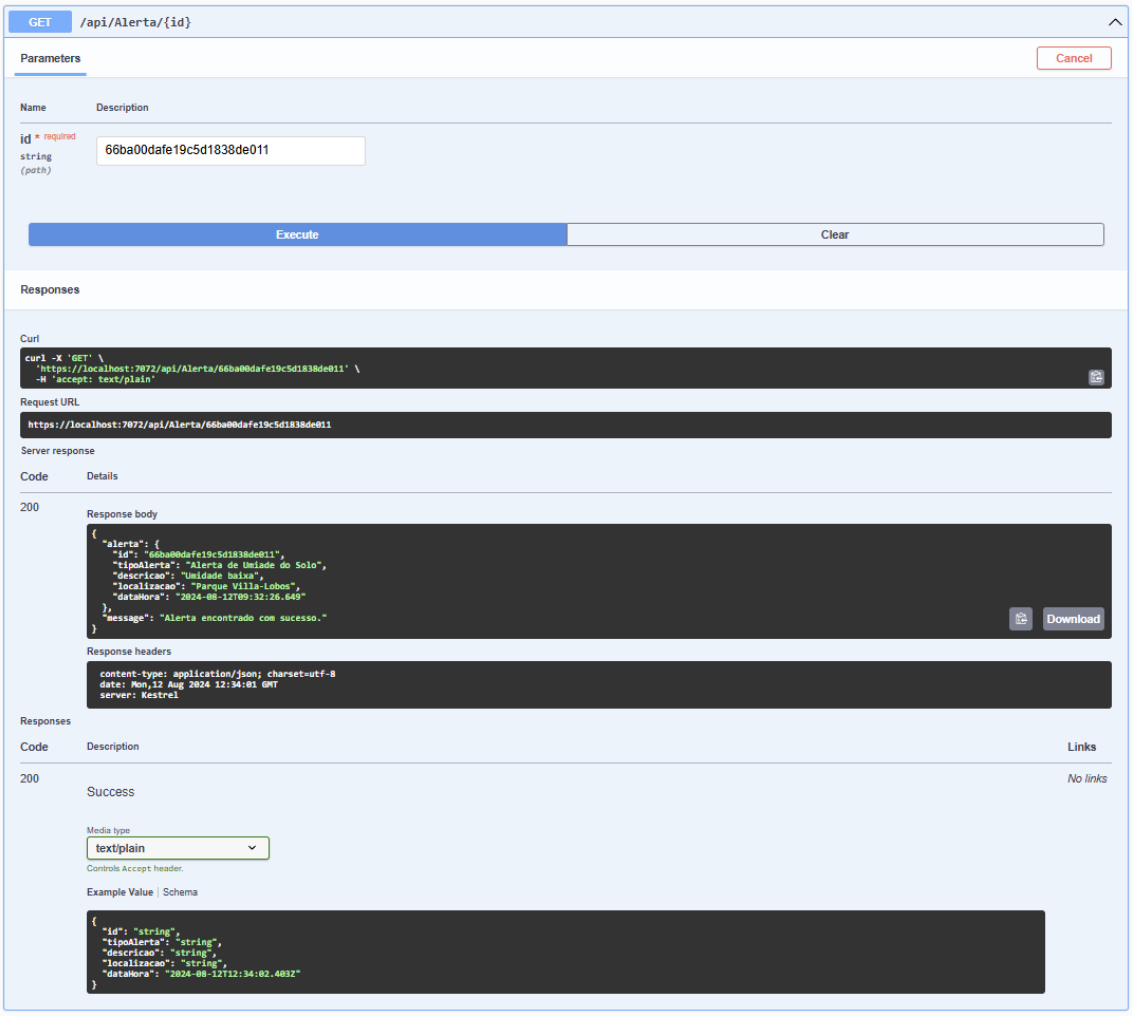


Figura 45

Update

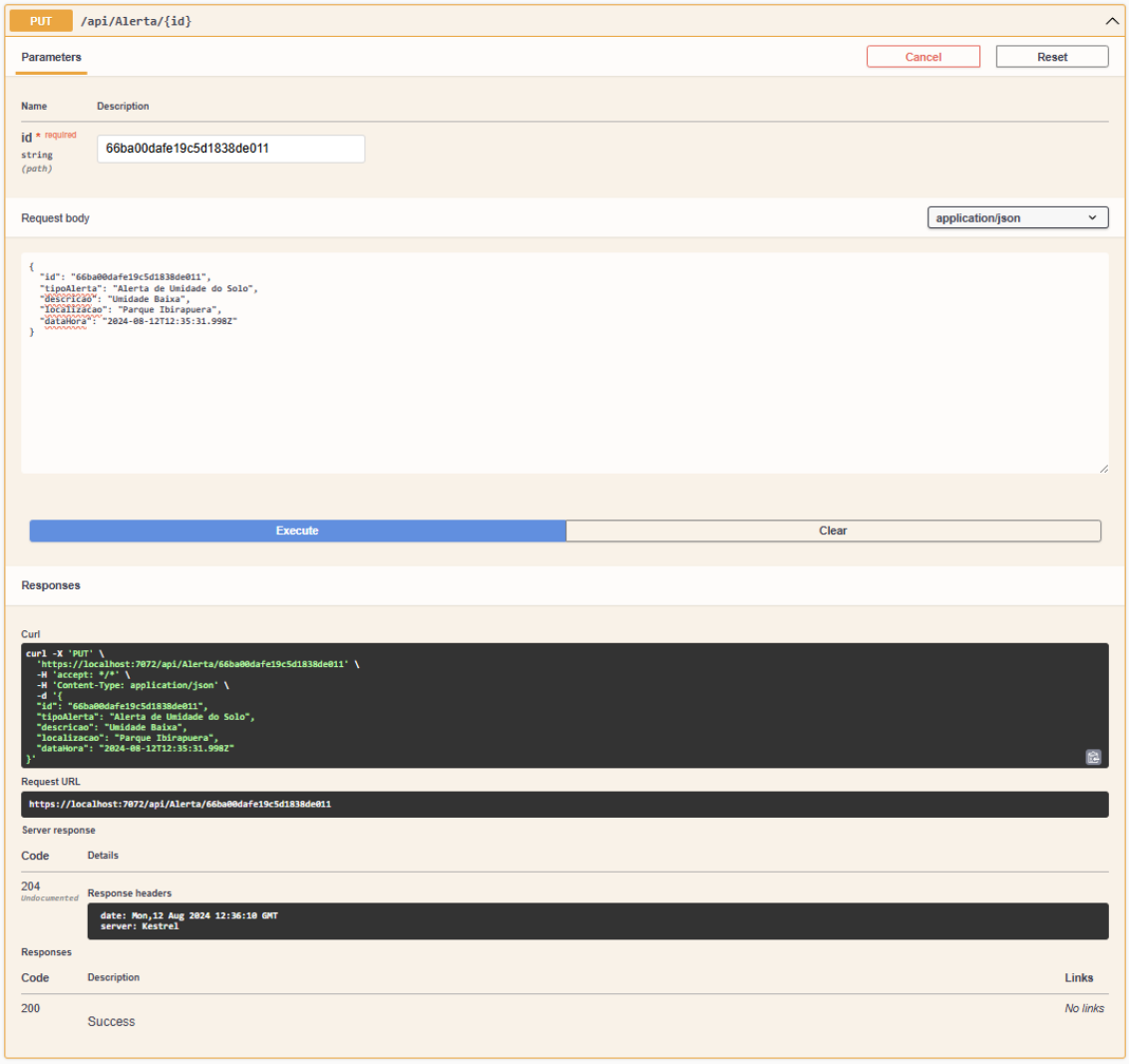


Figura 46

Delete

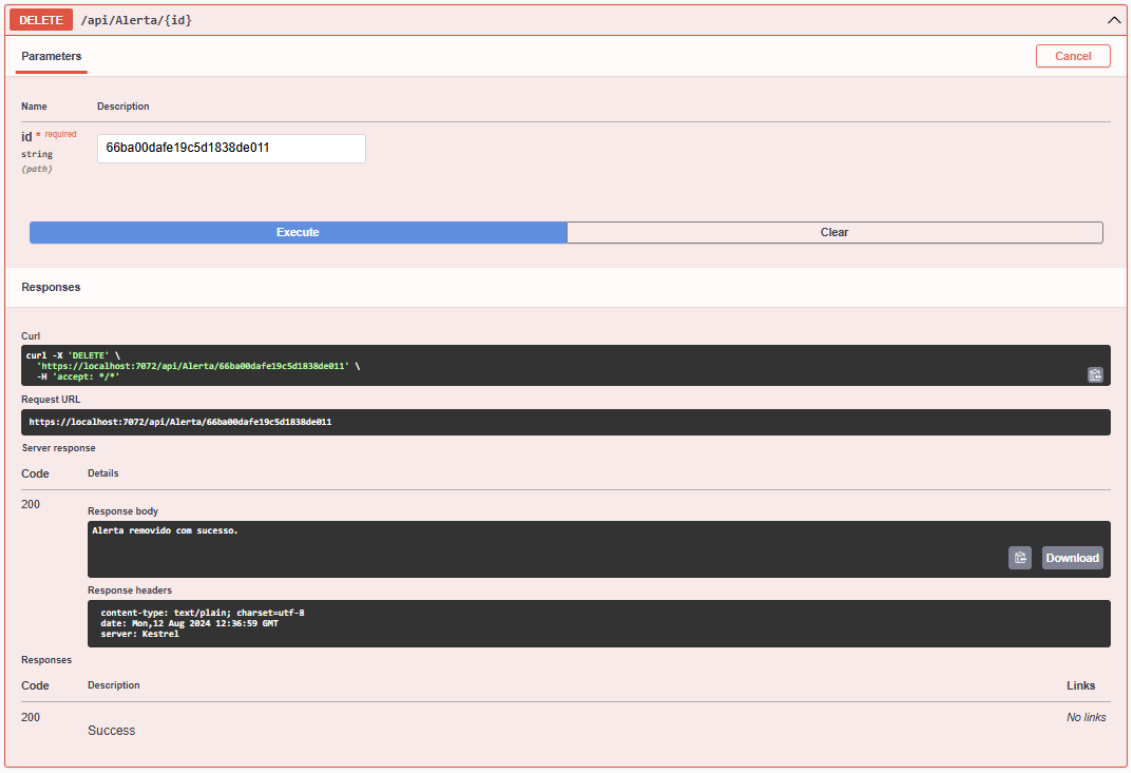


Figura 47

Collection Leituras

Insert

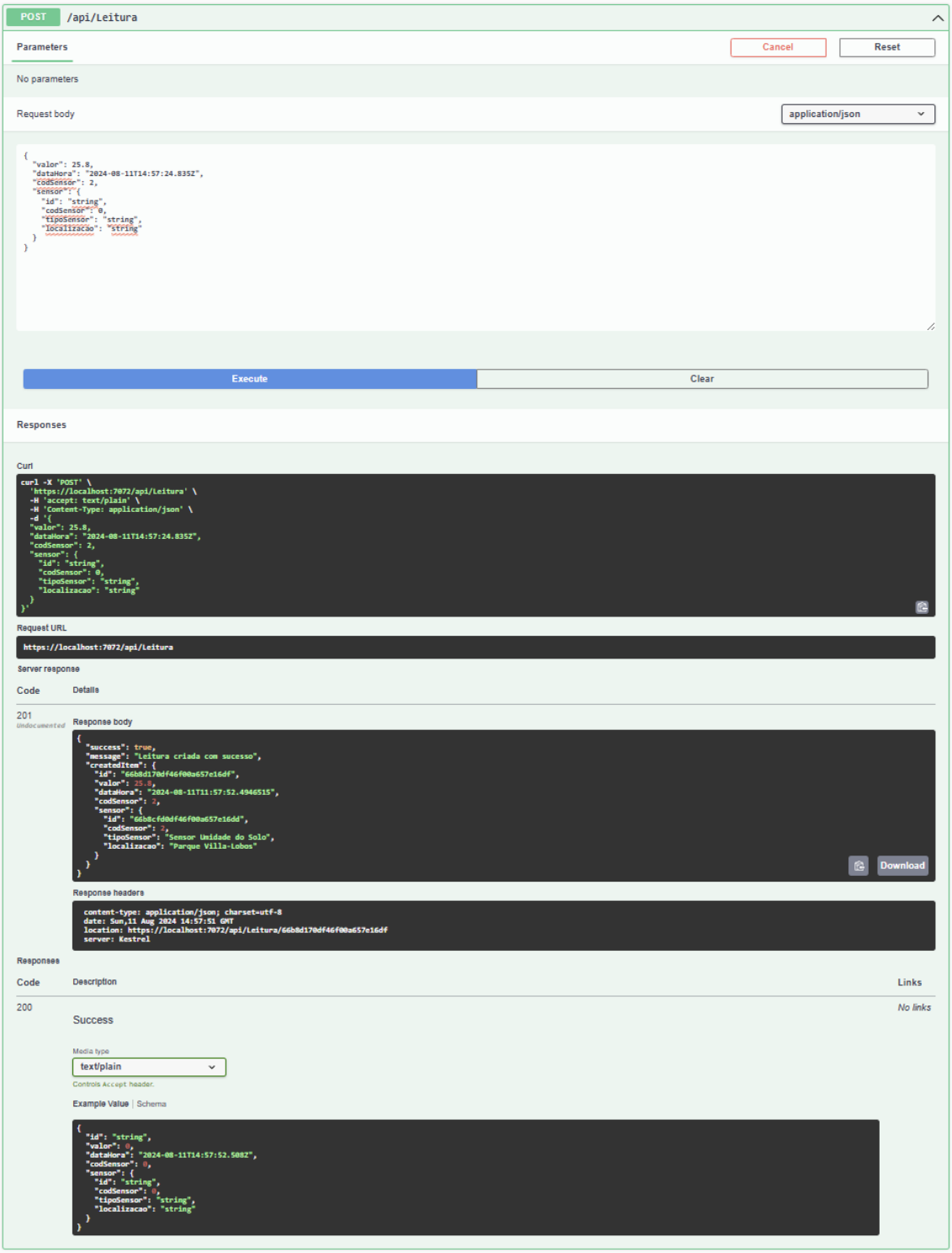


Figura 48

GetAll

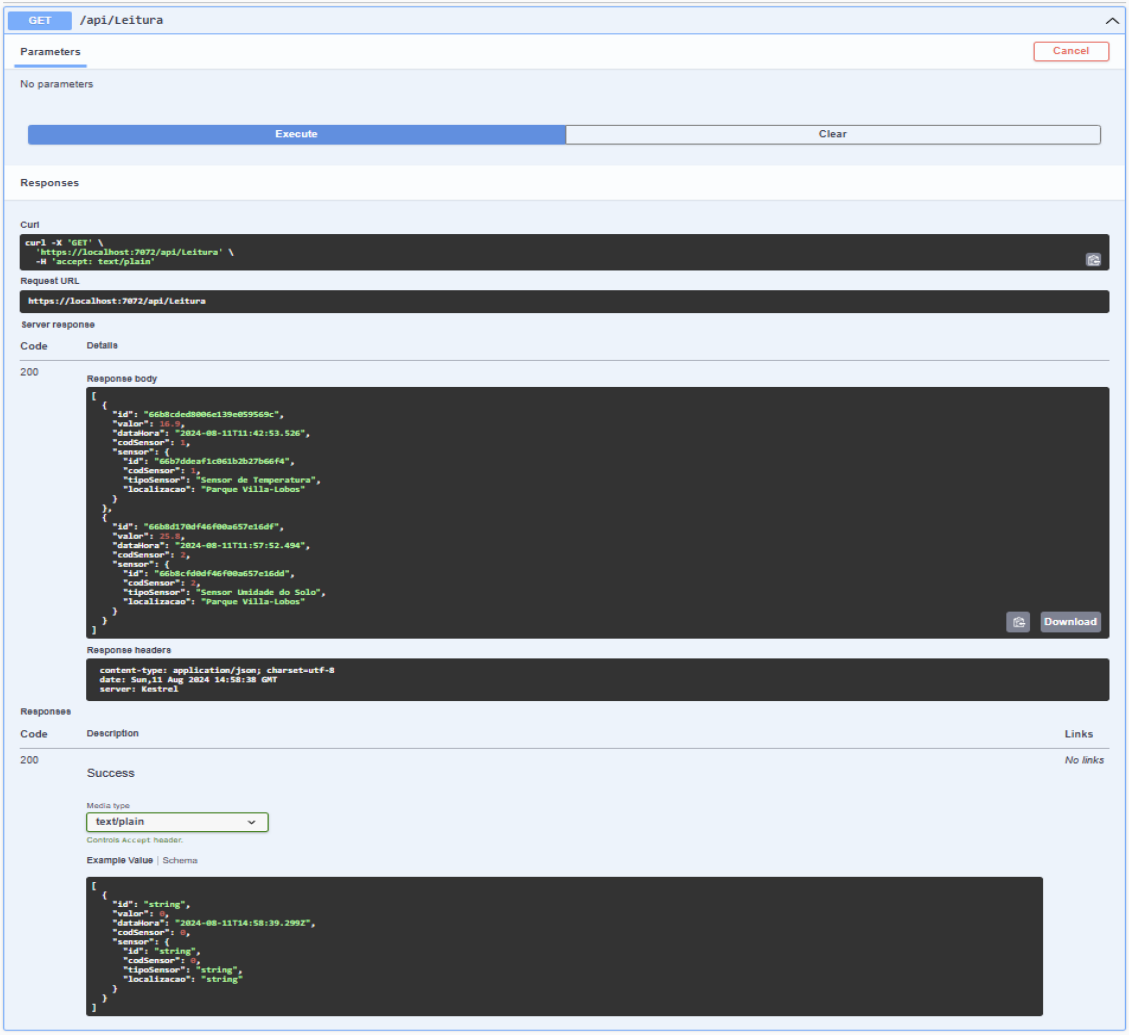


Figura 49

GeyById

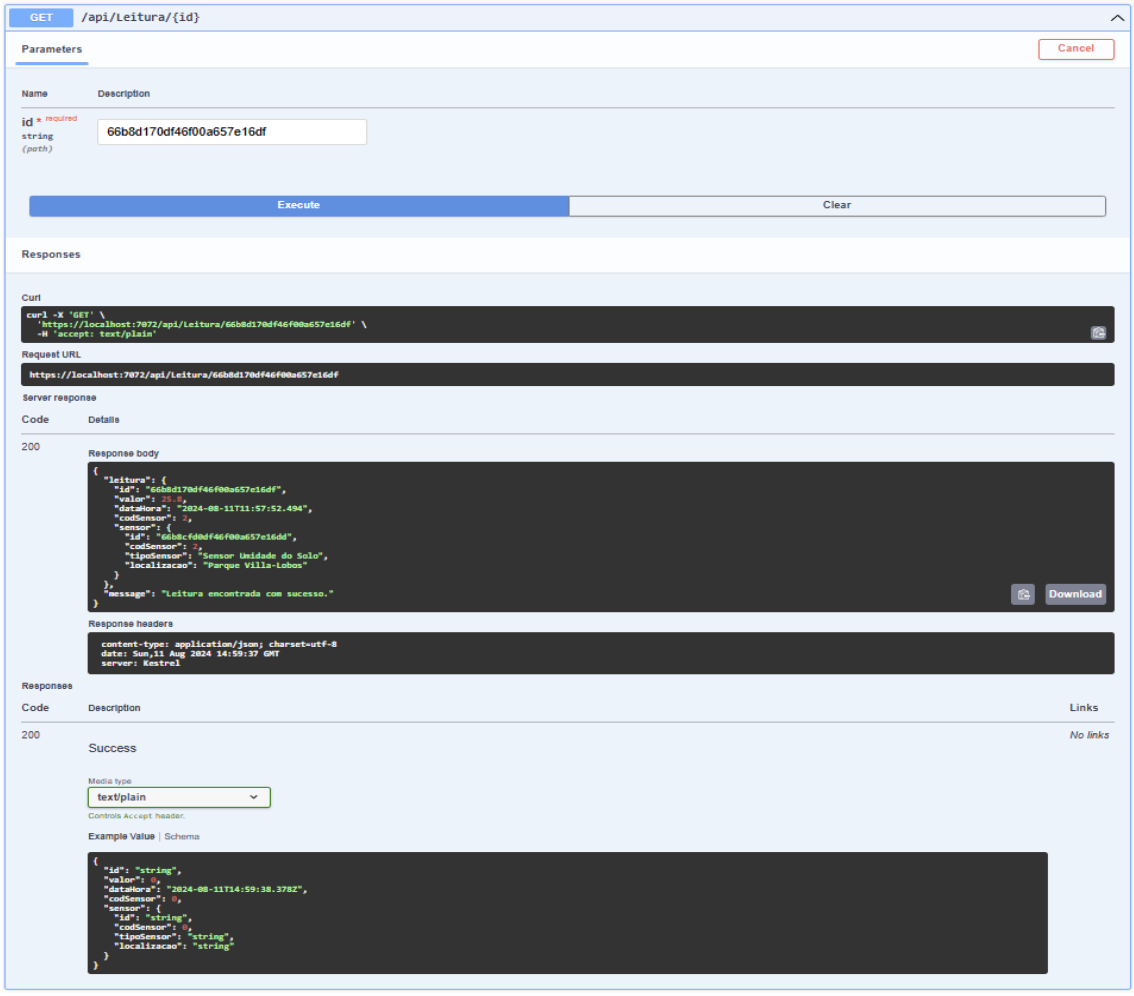


Figura 50

Update

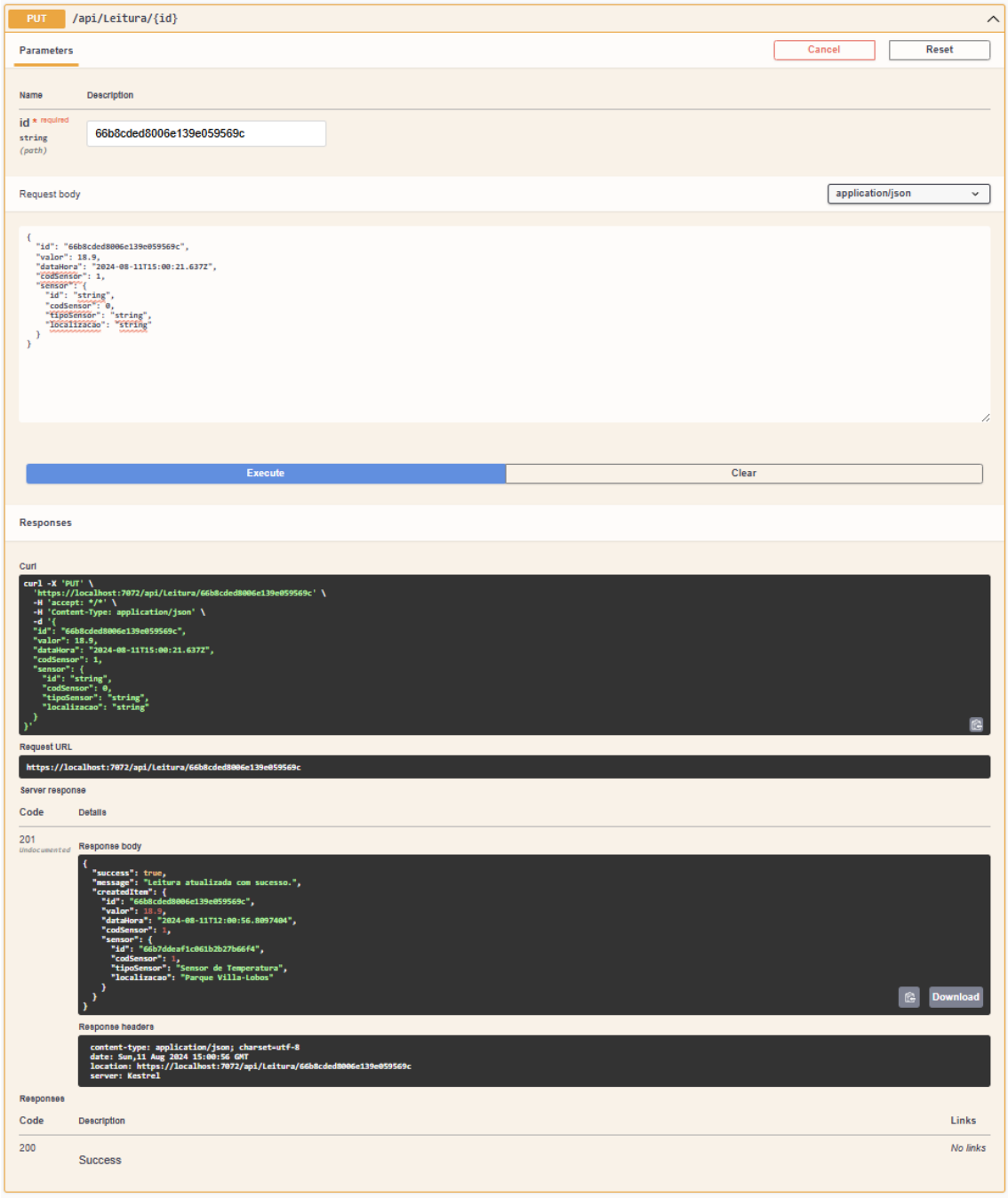


Figura 51

Delete

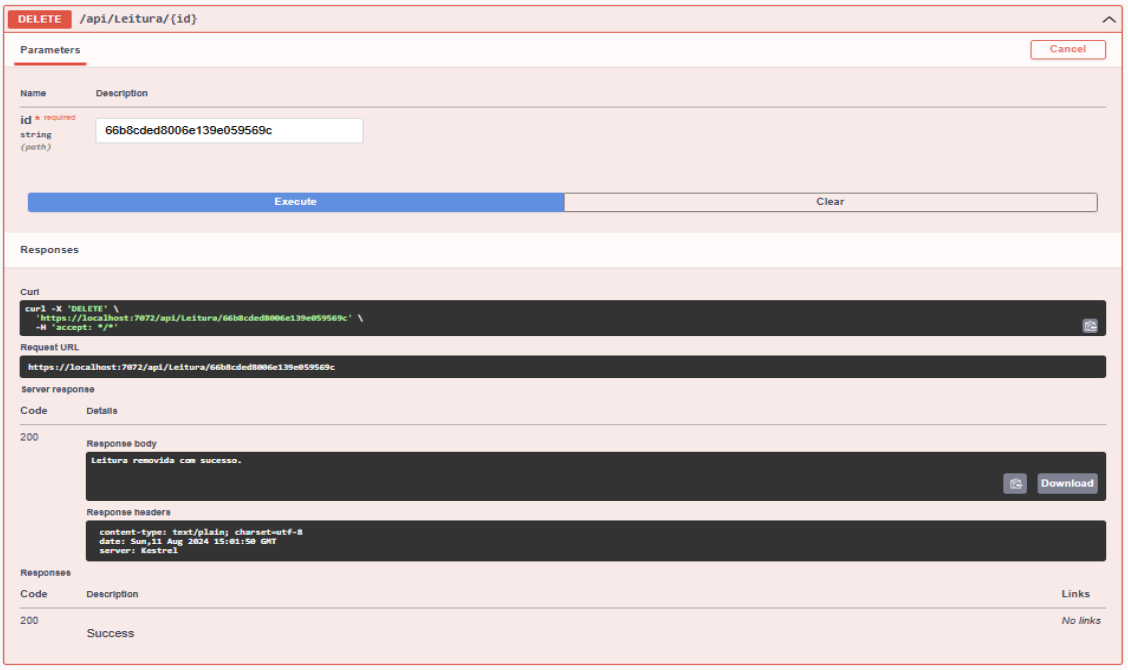


Figura 52

Collection Previsões Chuva

Insert

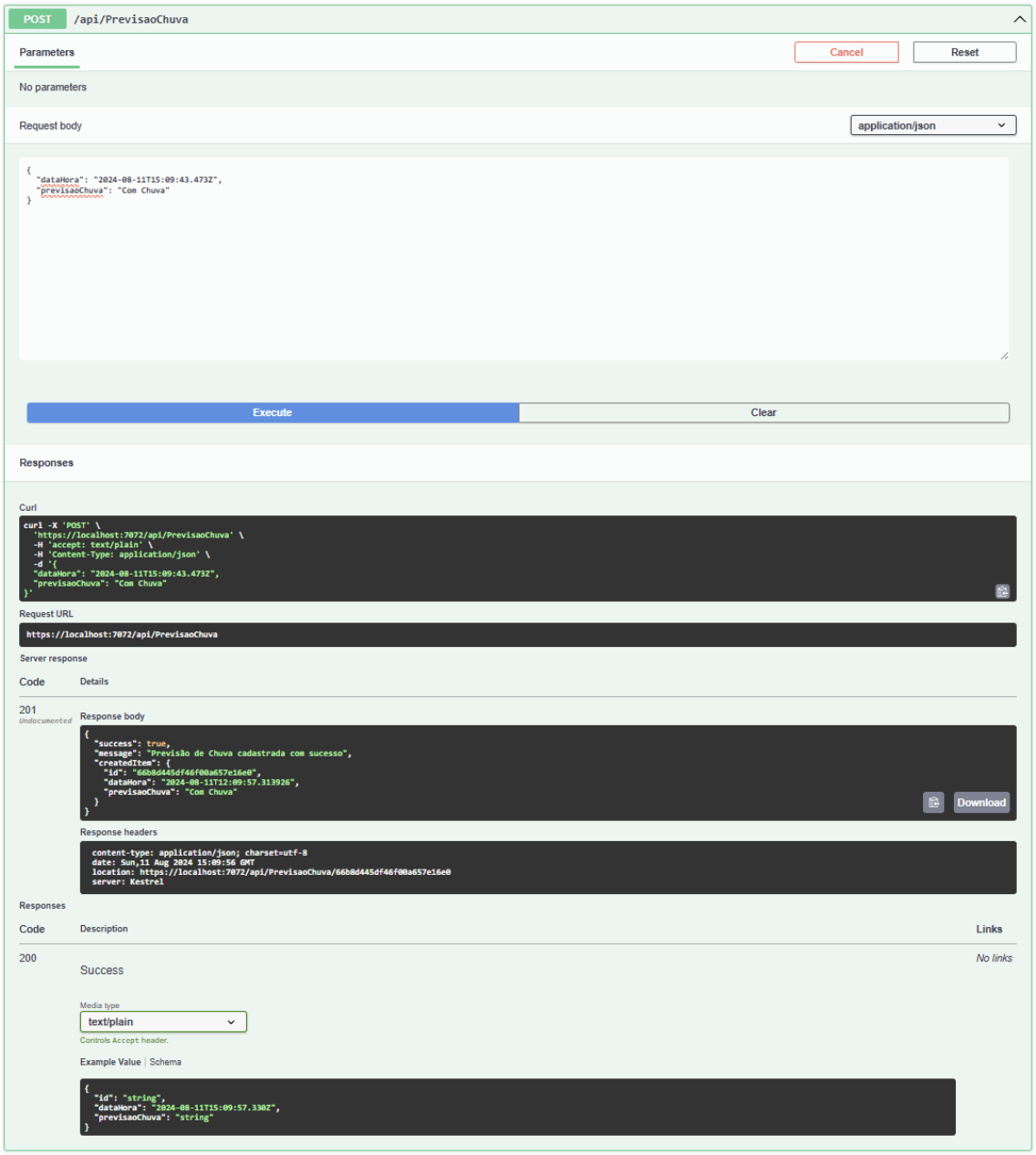


Figura 53

GetAll

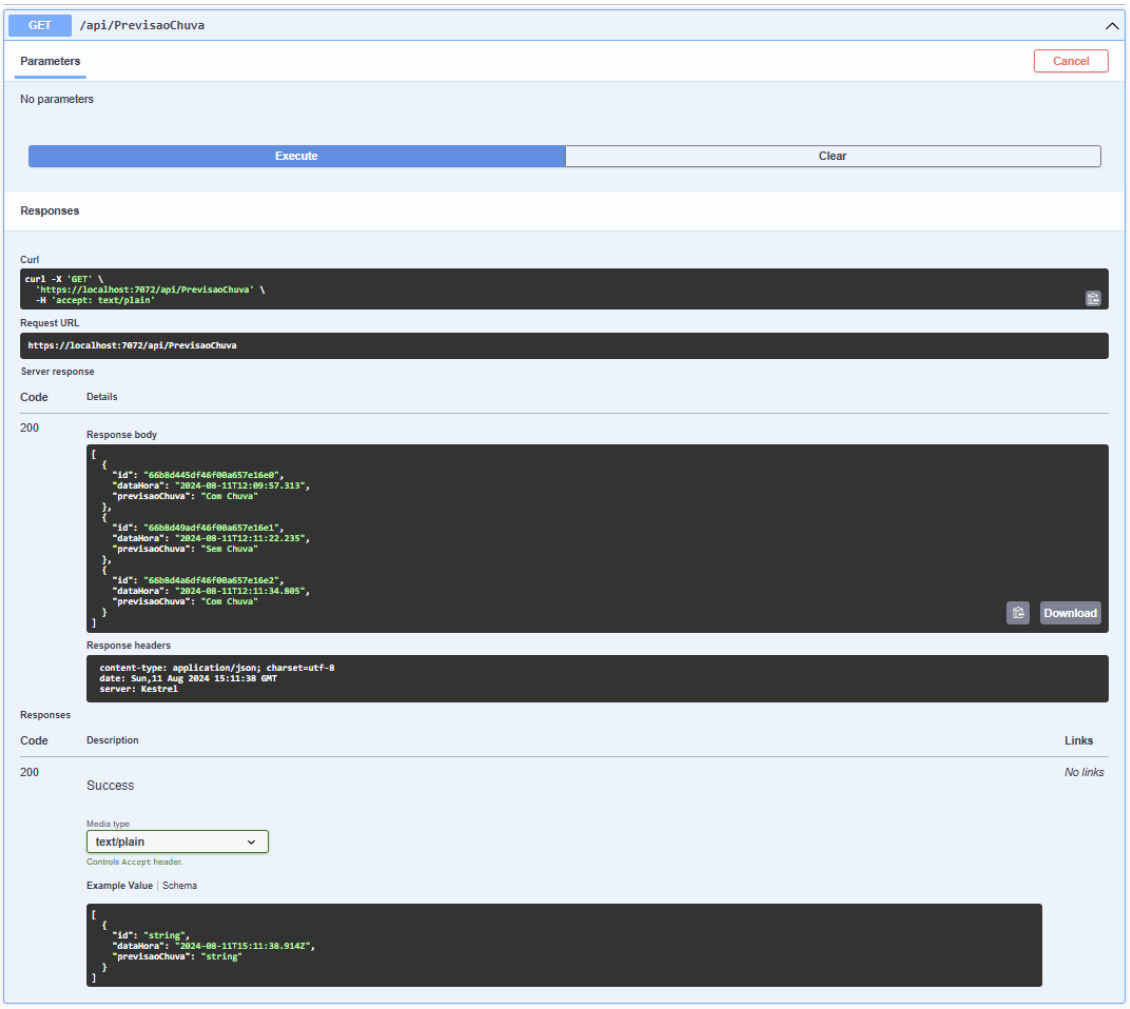


Figura 54

GetById

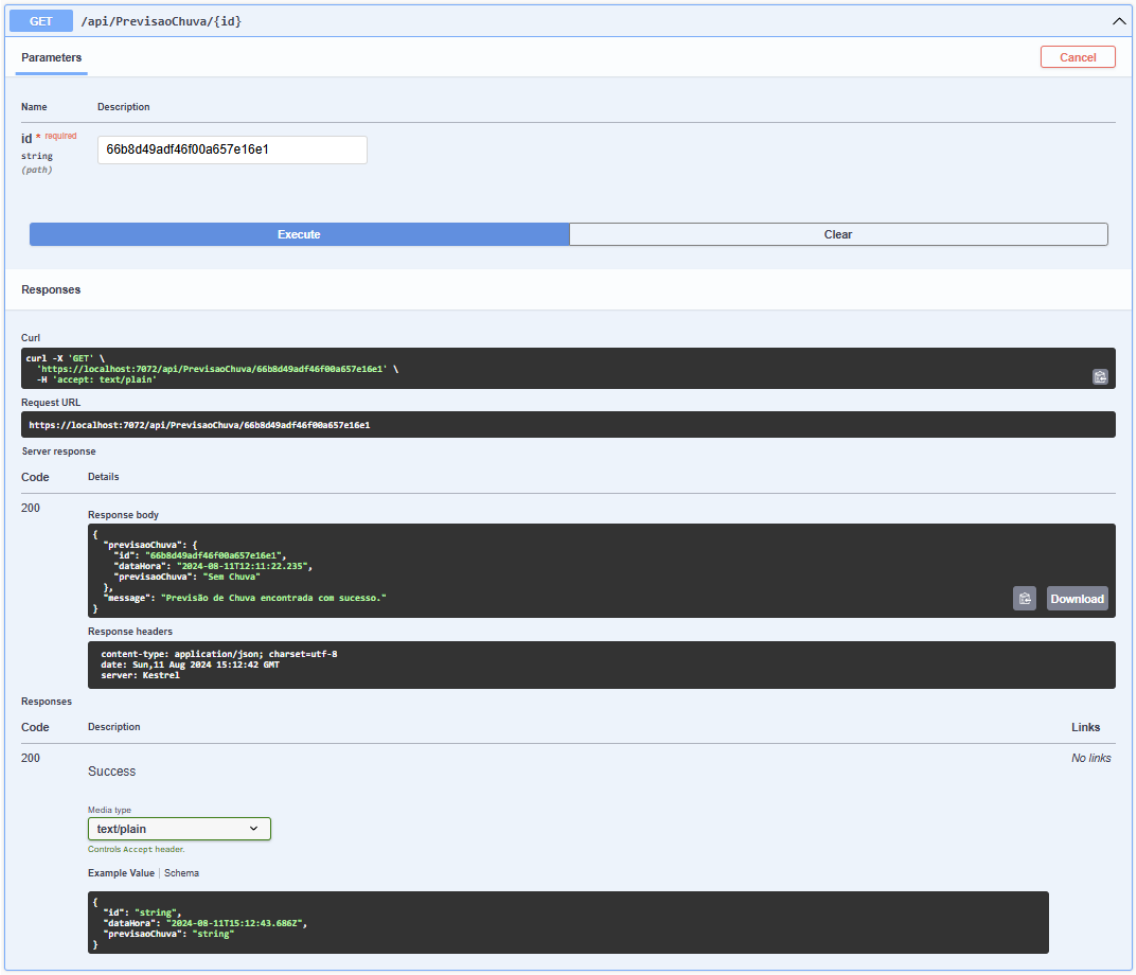


Figura 55

Update

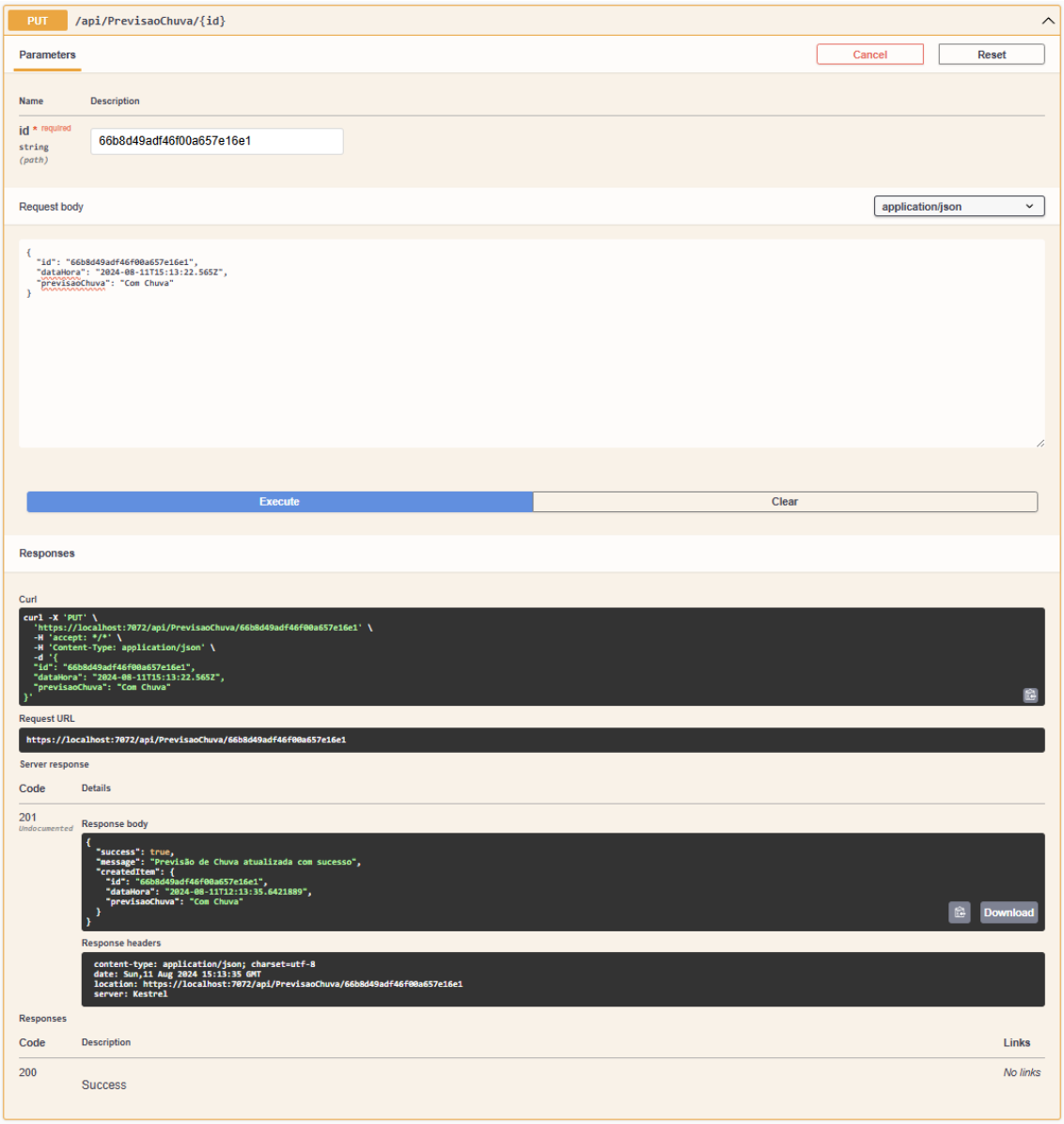


Figura 56

Delete

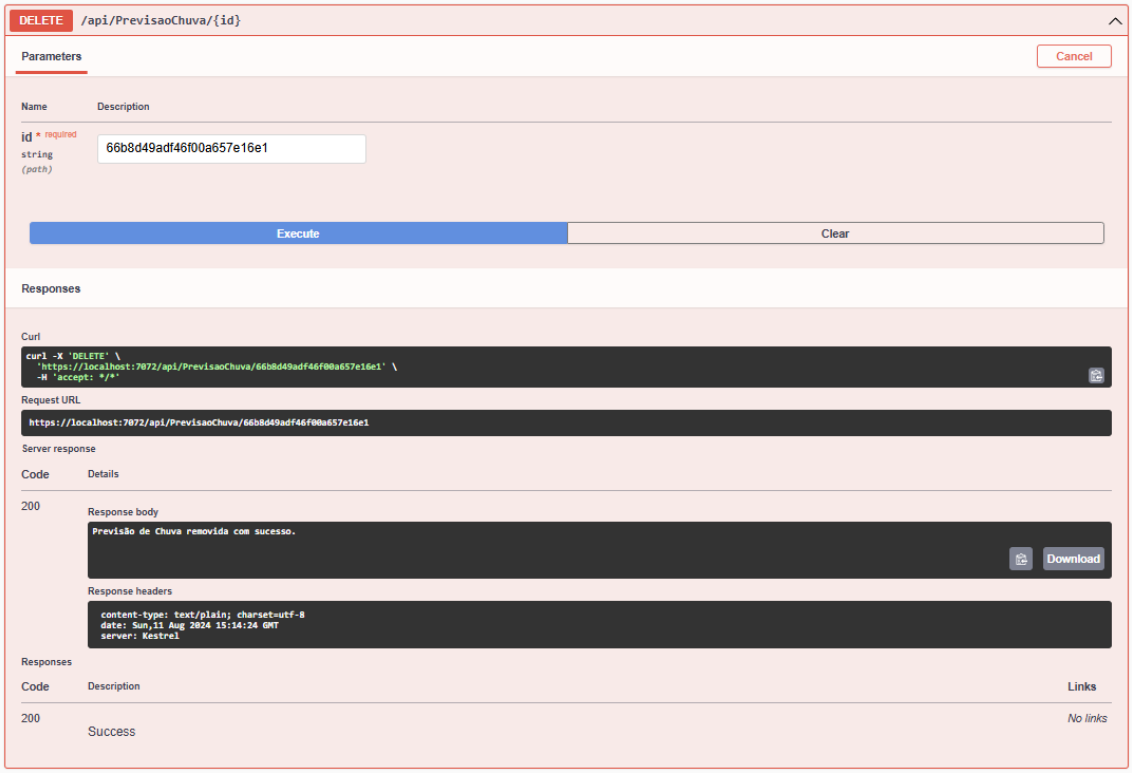


Figura 57

Collection Controle de Irrigações

Insert

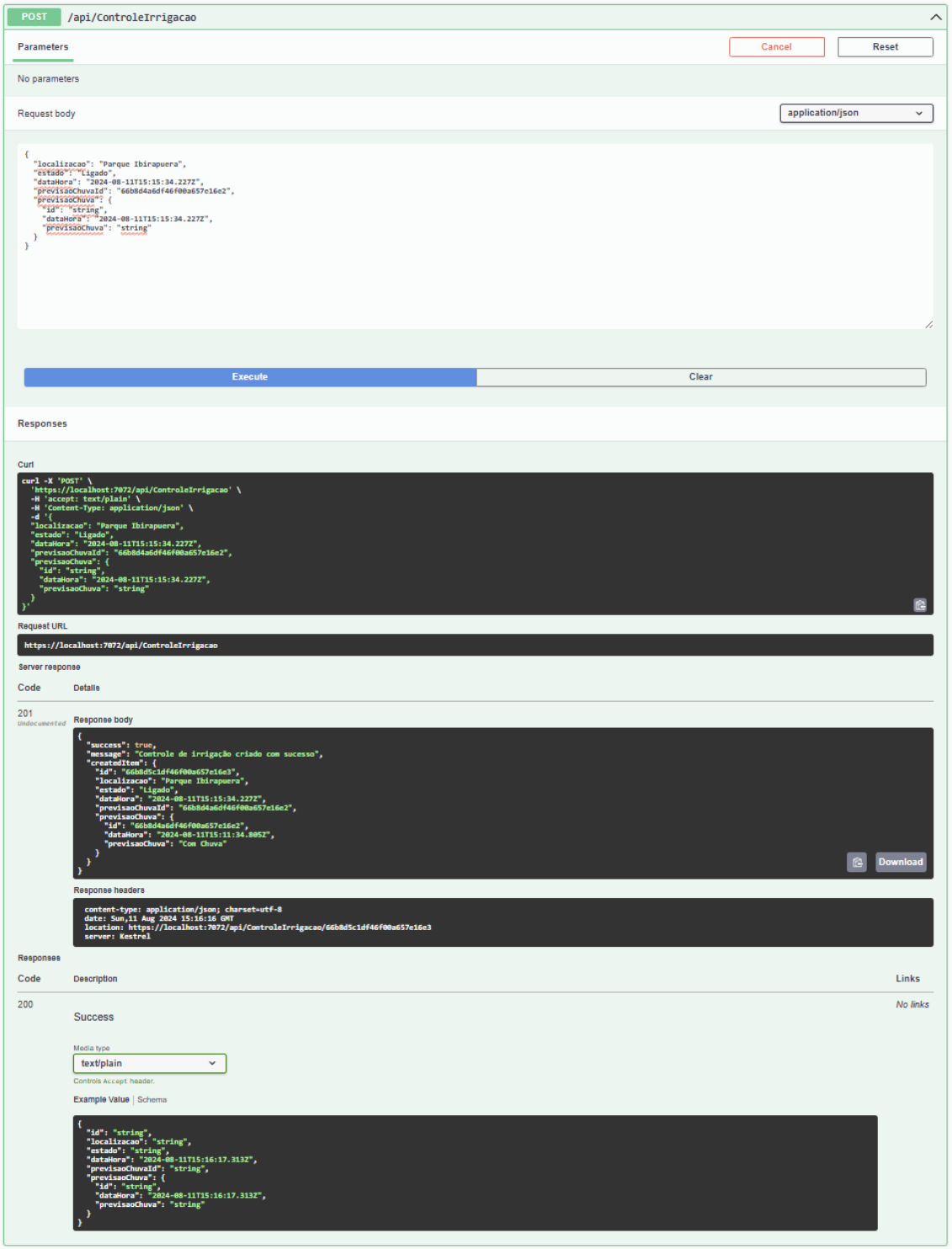


Figura 58

GetAll

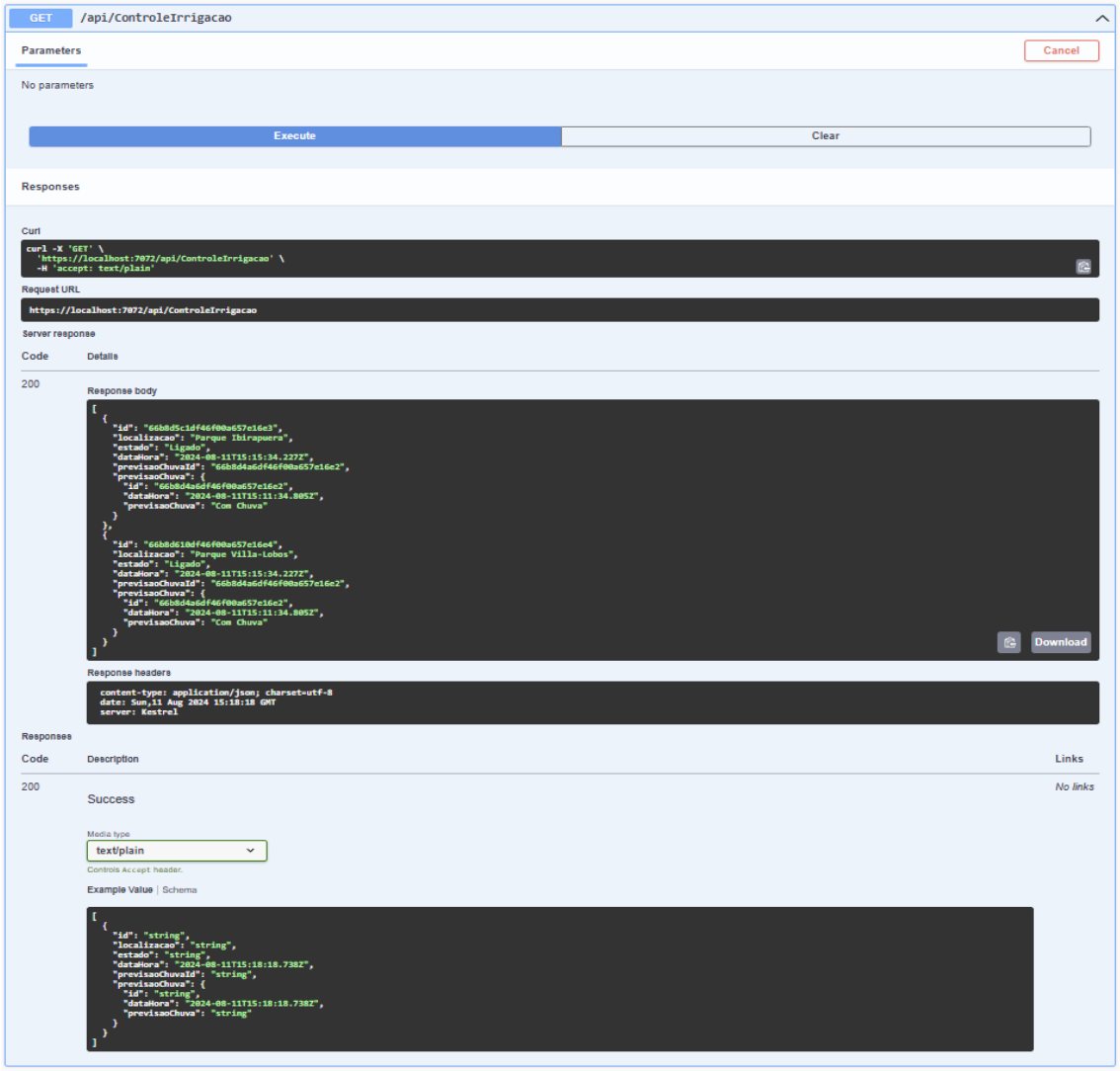


Figura 59

GetById

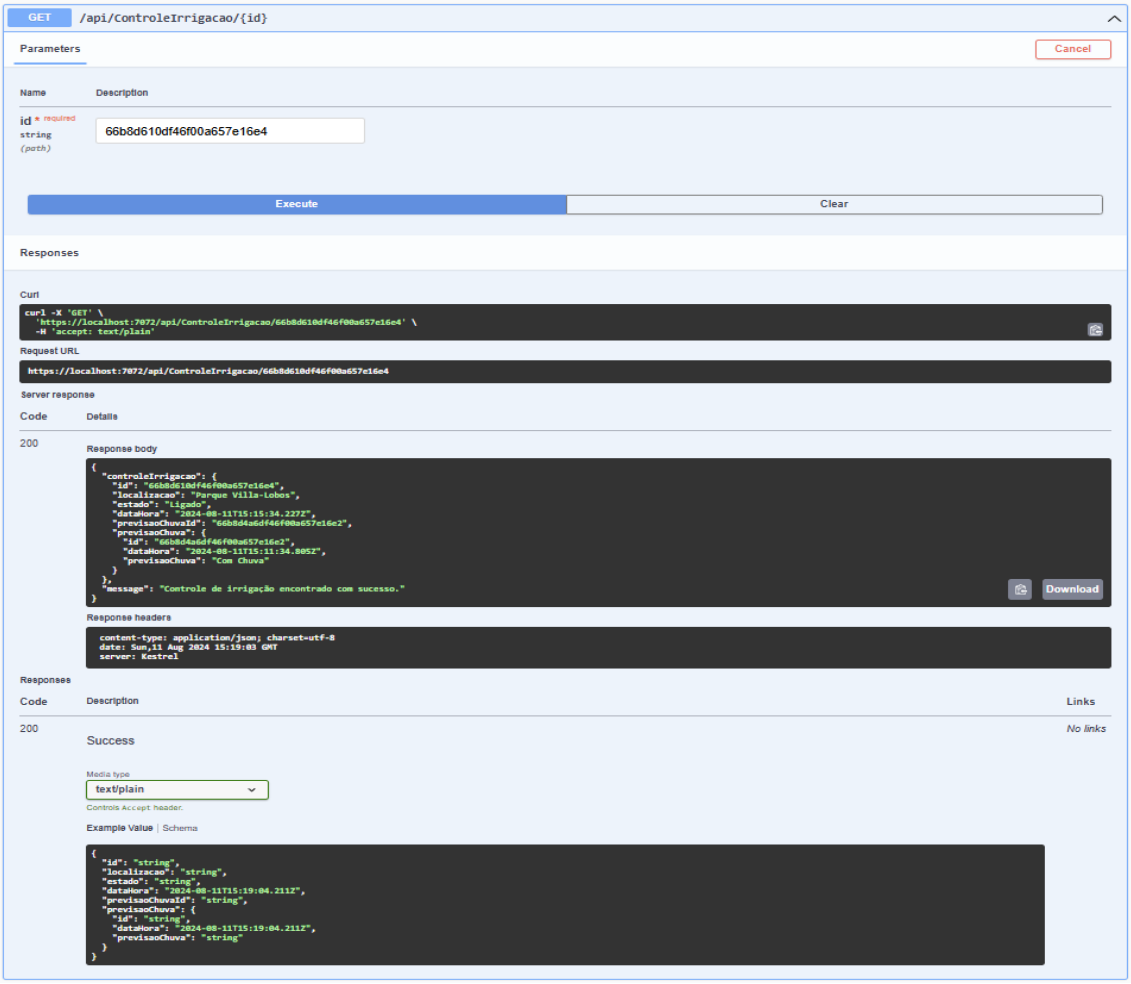


Figura 60

Update

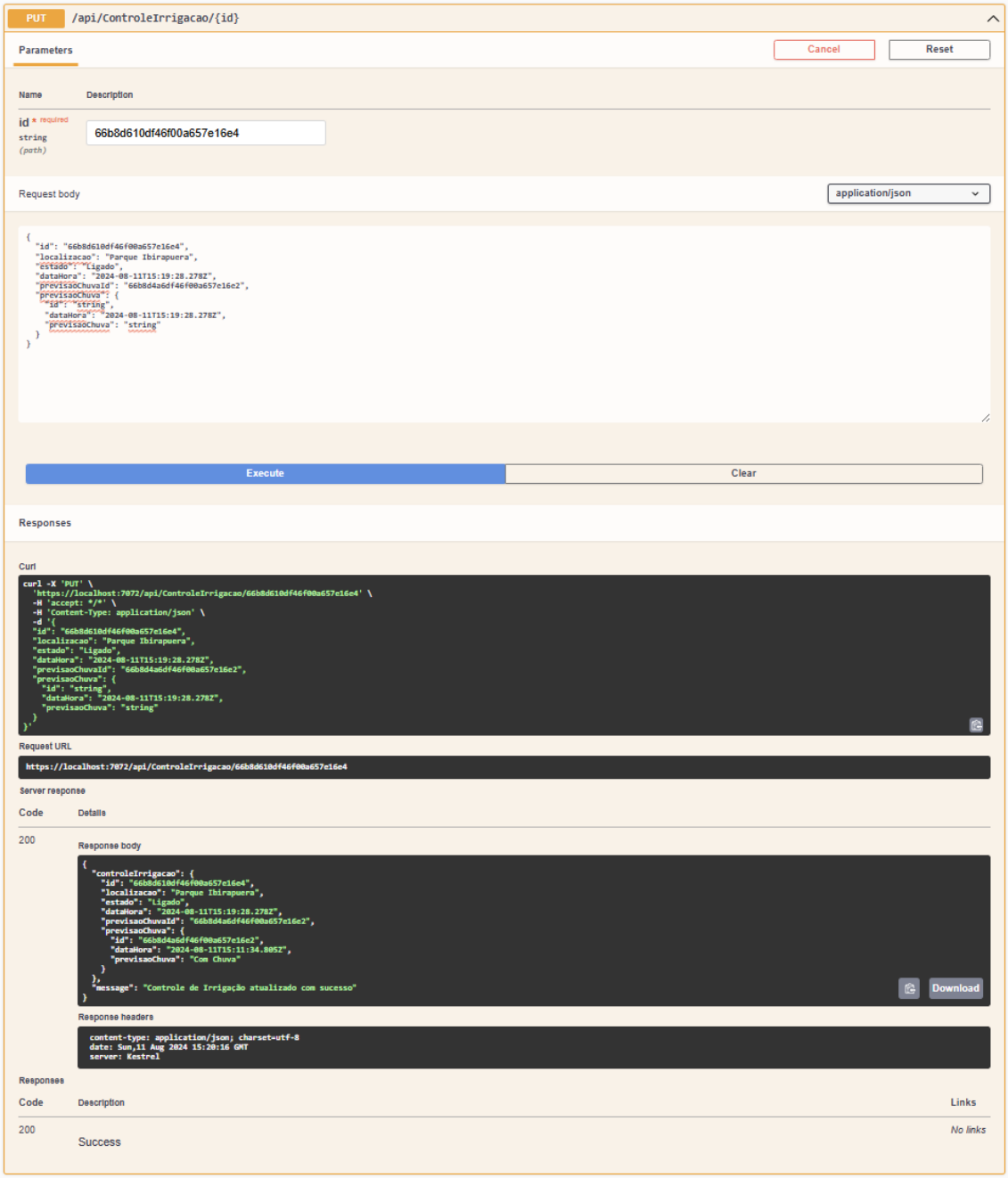


Figura 61

Delete

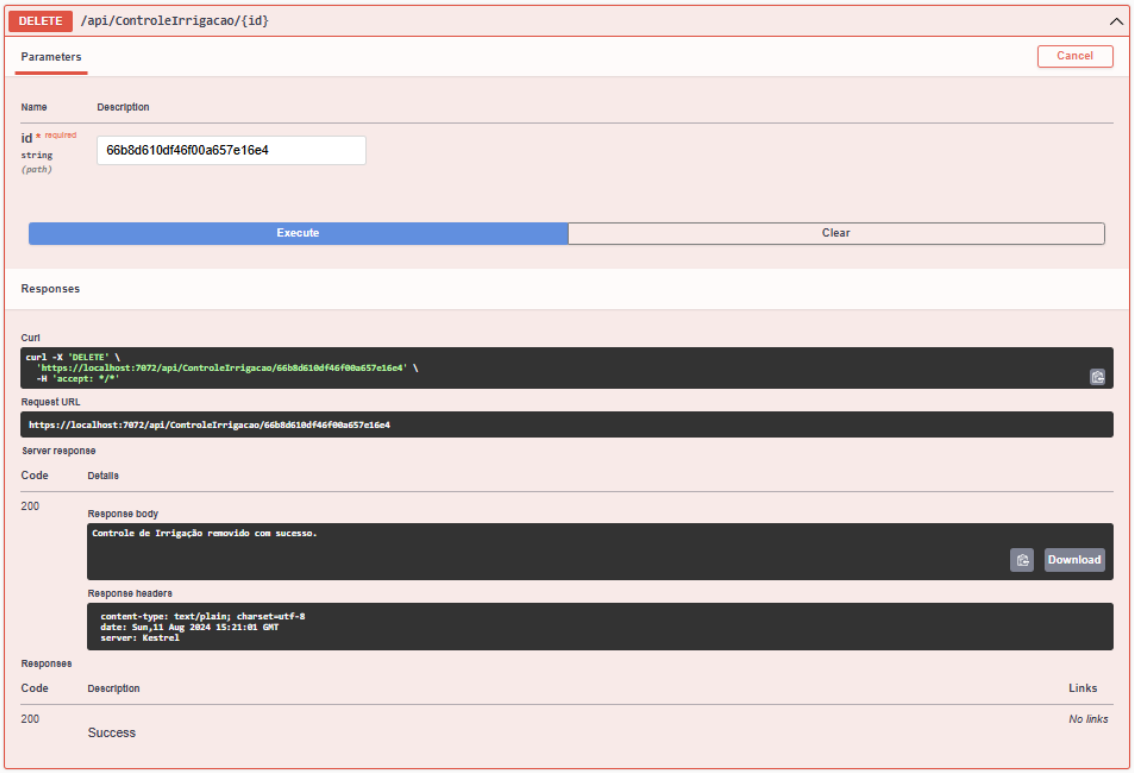


Figura 62