CRUD Nodejs y Mongodb
Fazt Code

https://www.youtube.com/watch?v=3J925fRl_UE

Resumo dos vídeos feito por Roberto Pinheiro

Criando arquivo de dependências

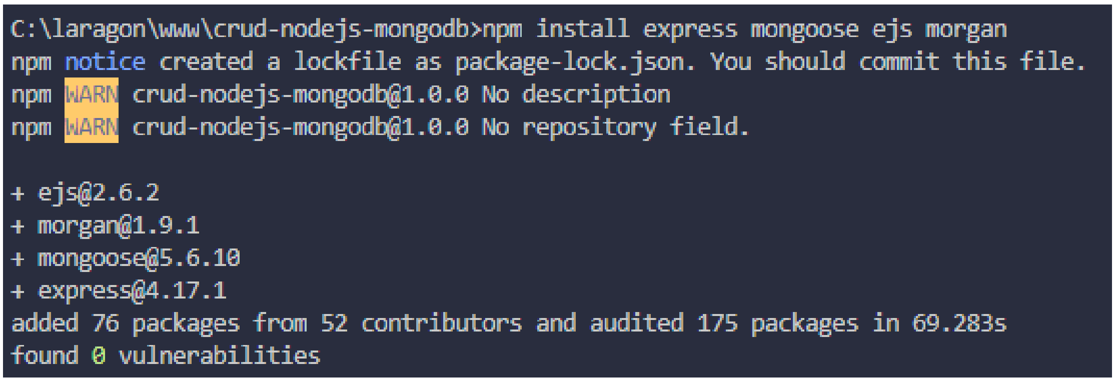
```
npm init --yes
```

package.json

```
{
  "name": "crud-nodejs-mongodb",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}
```

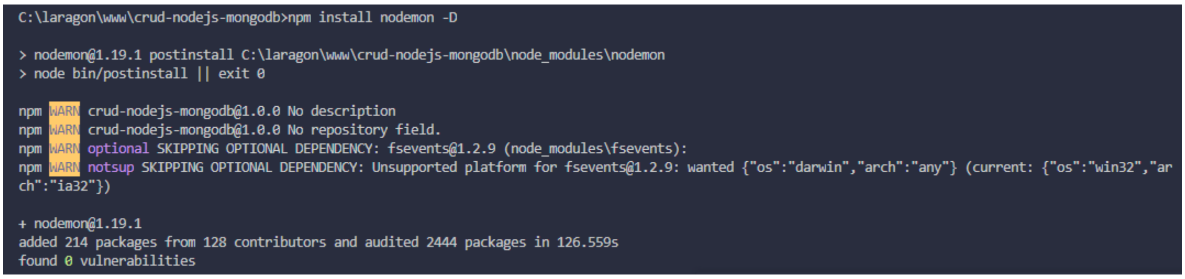
Instalando pacotes

```
npm install express mongoose ejs morgan
```



```
C:\laragon\www\crud-nodejs-mongodb>npm install express mongoose ejs morgan
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN crud-nodejs-mongodb@1.0.0 No description
npm WARN crud-nodejs-mongodb@1.0.0 No repository field.

+ ejs@2.6.2
+ morgan@1.9.1
+ mongoose@5.6.10
+ express@4.17.1
added 76 packages from 52 contributors and audited 175 packages in 69.283s
found 0 vulnerabilities
```

```
npm install nodemon -D
```



```
C:\laragon\www\crud-nodejs-mongodb>npm install nodemon -D

> nodemon@1.19.1 postinstall C:\laragon\www\crud-nodejs-mongodb\node_modules\nodemon
> node bin/postinstall || exit 0

npm WARN crud-nodejs-mongodb@1.0.0 No description
npm WARN crud-nodejs-mongodb@1.0.0 No repository field.
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.9 (node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.9: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","ar
ch":"ia32"})

+ nodemon@1.19.1
added 214 packages from 128 contributors and audited 2444 packages in 126.559s
found 0 vulnerabilities
```

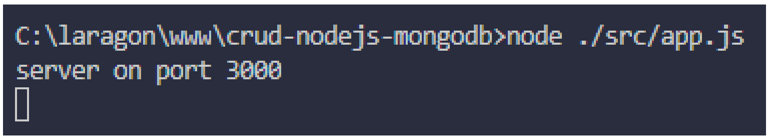
package.json

```
{
  "name": "crud-nodejs-mongodb",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "dependencies": {
    "ejs": "^2.6.2",
    "express": "^4.17.1",
    "mongoose": "^5.6.10",
    "morgan": "^1.9.1"
  },
  "devDependencies": {
    "nodemon": "^1.19.1"
  }
}
```

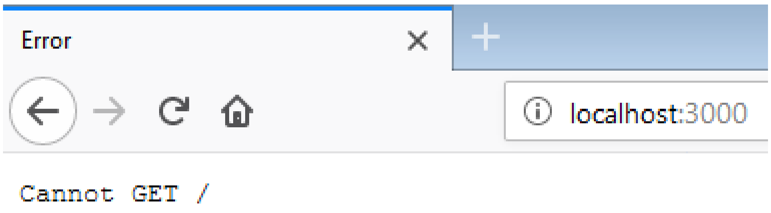
Configurando o servidor

app.js

```
const express = require('express');
const app = express();

// settings
app.set('port', process.env.PORT || 3000);

app.listen(app.get('port'), () => {
  console.log(`server on port ${app.get('port')}`);
});
```



```
C:\laragon\www\crud-nodejs-mongodb>node ./src/app.js
server on port 3000
```



```
Cannot GET /
```

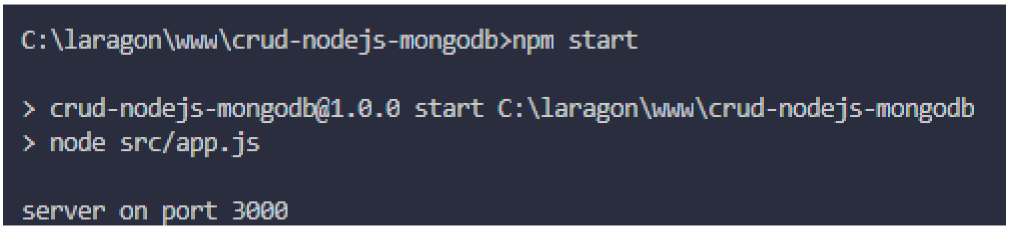
O erro especificado é porque ainda não foi definida nenhuma rota para a página principal.

package.json

```
{
  "name": "crud-nodejs-mongodb",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "start": "node src/app.js",
    "dev": "nodemon src/app.js"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "dependencies": {
    "ejs": "^2.6.2",
    "express": "^4.17.1",
    "mongoose": "^5.6.10",
    "morgan": "^1.9.1"
  },
  "devDependencies": {
    "nodemon": "^1.19.1"
  }
}
```

Em produção (production), no servidor de hospedagem será executado:

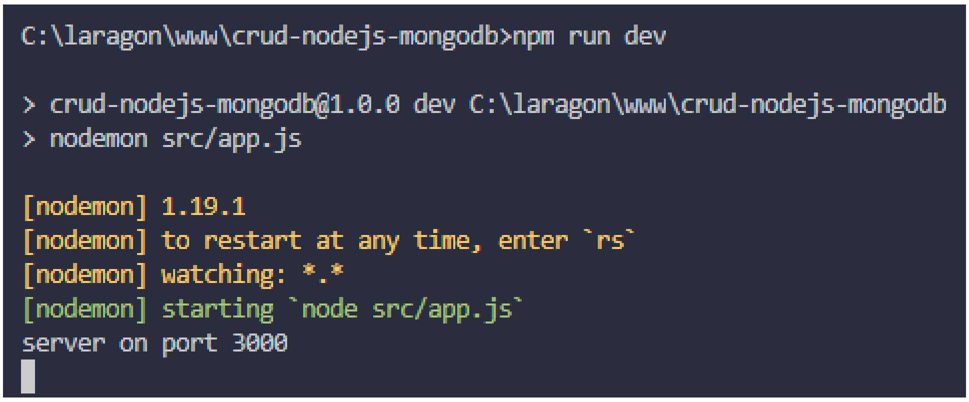
npm start



```
C:\laragon\www\crud-nodejs-mongodb>npm start

> crud-nodejs-mongodb@1.0.0 start C:\laragon\www\crud-nodejs-mongodb
> node src/app.js

server on port 3000
```

Em desenvolvimento (devlopment), no servidor local será executado:

npm run dev



```
C:\laragon\www\crud-nodejs-mongodb>npm run dev

> crud-nodejs-mongodb@1.0.0 dev C:\laragon\www\crud-nodejs-mongodb
> nodemon src/app.js

[nodemon] 1.19.1
[nodemon] to restart at any time, enter `rs`
[nodemon] watching: *.*
[nodemon] starting `node src/app.js`
server on port 3000
```

Configurando views

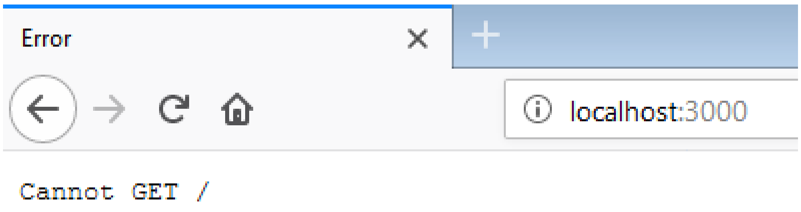
app.js

```
const path = require('path');
const express = require('express');
const app = express();

/* settings */

// configurando a porta do servidor
app.set('port', process.env.PORT || 3000);

// configurando o diretório de views
app.set('views', path.join(__dirname, 'views'));

// definindo o motor de templates
app.set('view engine', 'ejs');

/* starting the server */
app.listen(app.get('port'), () => {
  console.log(`server on port ${app.get('port')}`);
});
```

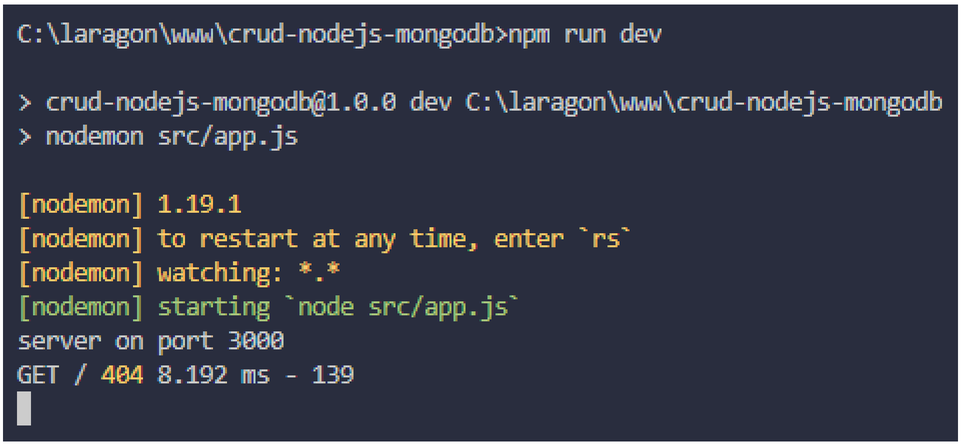
Configurando Middlewares

app.js

```
const path = require('path');
const express = require('express');
const morgan = require('morgan');

const app = express();

/* settings */

// configurando a porta do servidor
app.set('port', process.env.PORT || 3000);

// configurando o diretório de views
app.set('views', path.join(__dirname, 'views'));

// definindo o motor de templates
app.set('view engine', 'ejs');

/* middlewares */
app.use(morgan('dev'));


/* starting the server */
app.listen(app.get('port'), () => {
  console.log(`server on port ${app.get('port')}`);
});
```



```
Cannot GET /
```



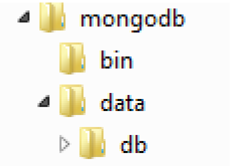
```
C:\laragon\www\crud-nodejs-mongodb>npm run dev

> crud-nodejs-mongodb@1.0.0 dev C:\laragon\www\crud-nodejs-mongodb
> nodemon src/app.js

[nodemon] 1.19.1
[nodemon] to restart at any time, enter `rs`
[nodemon] watching: *.*
[nodemon] starting `node src/app.js`
server on port 3000
GET / 404 8.192 ms - 139
```

Conectando com o banco de dados

Rodando o servidor do Mongo

- Criar a pasta c:\mongodb e copiar os arquivos do Mongo para dentro dela.

- Dentro da pasta mongodb criar a subpasta data e dentro dela criar a subpasta db.



- Inserir o caminho da pasta bin do MongoDB no path do Windows.

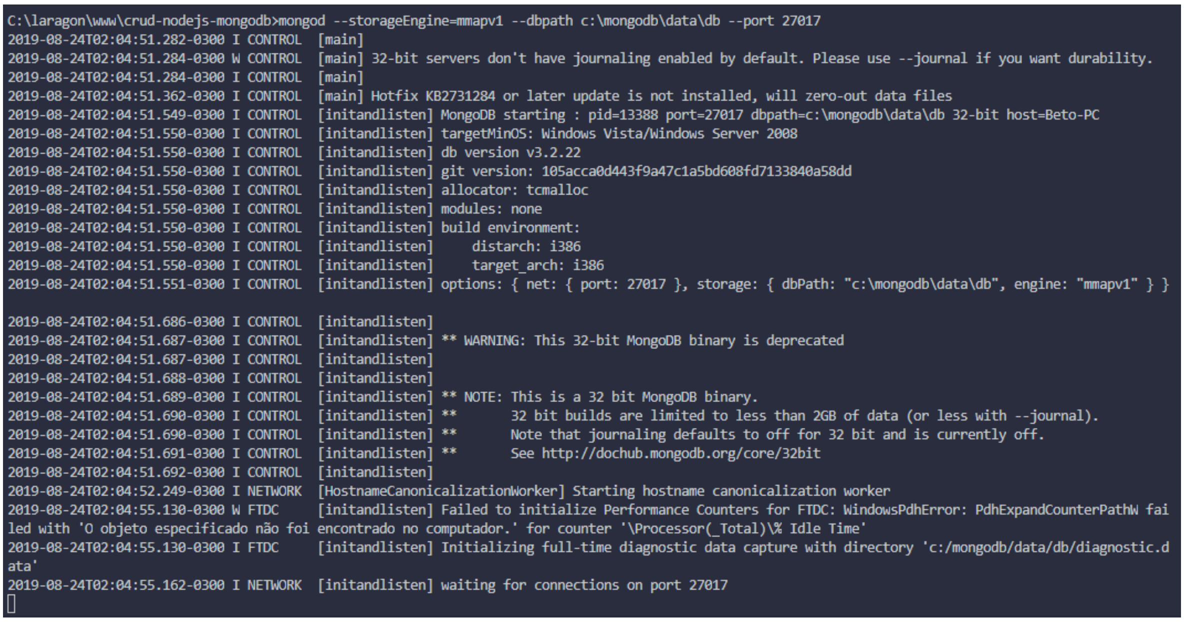
C:\mongodb\bin

Importante: É dentro desta pasta mongodb/data/db que serão salvos os arquivos criados pelo Mongo.

Abrindo o servidor do MongoDB no Windows 7 - 32 bits

Sempre que for trabalhar com o MongoDB é necessário abrir o seu servidor. Para isso, entre no seu terminal com o seguinte comando:

```
mongod --storageEngine=mmapv1 --dbpath c:\mongodb\data\db --port 27017
```



```
C:\laragon\www\crud-nodejs-mongodb>mongod --storageEngine=mmapv1 --dbpath c:\mongodb\data\db --port 27017
2019-08-24T02:04:51.282-0300 I CONTROL  [main]
2019-08-24T02:04:51.284-0300 W CONTROL  [main] 32-bit servers don't have journaling enabled by default. Please use --journal if you want durability.
2019-08-24T02:04:51.284-0300 I CONTROL  [main]
2019-08-24T02:04:51.362-0300 I CONTROL  [main] Hotfix KB2731284 or later update is not installed, will zero-out data files
2019-08-24T02:04:51.549-0300 I CONTROL  [initandlisten] MongoDB starting : pid=13388 port=27017 dbpath=c:\mongodb\data\db 32-bit host=Beto-PC
2019-08-24T02:04:51.550-0300 I CONTROL  [initandlisten] targetMinOS: Windows Vista/Windows Server 2008
2019-08-24T02:04:51.550-0300 I CONTROL  [initandlisten] db version v3.2.22
2019-08-24T02:04:51.550-0300 I CONTROL  [initandlisten] git version: 105acca0d443f9a47c1a5bd608fd7133840a58dd
2019-08-24T02:04:51.550-0300 I CONTROL  [initandlisten] allocator: tcmalloc
2019-08-24T02:04:51.550-0300 I CONTROL  [initandlisten] modules: none
2019-08-24T02:04:51.550-0300 I CONTROL  [initandlisten] build environment:
2019-08-24T02:04:51.550-0300 I CONTROL  [initandlisten]     distarch: i386
2019-08-24T02:04:51.550-0300 I CONTROL  [initandlisten]     target_arch: i386
2019-08-24T02:04:51.551-0300 I CONTROL  [initandlisten] options: { net: { port: 27017 }, storage: { dbPath: "c:\mongodb\data\db", engine: "mmapv1" } }

2019-08-24T02:04:51.686-0300 I CONTROL  [initandlisten]
2019-08-24T02:04:51.687-0300 I CONTROL  [initandlisten] ** WARNING: This 32-bit MongoDB binary is deprecated
2019-08-24T02:04:51.687-0300 I CONTROL  [initandlisten]
2019-08-24T02:04:51.688-0300 I CONTROL  [initandlisten]
2019-08-24T02:04:51.689-0300 I CONTROL  [initandlisten] ** NOTE: This is a 32 bit MongoDB binary.
2019-08-24T02:04:51.690-0300 I CONTROL  [initandlisten] **       32 bit builds are limited to less than 2GB of data (or less with --journal).
2019-08-24T02:04:51.690-0300 I CONTROL  [initandlisten] **       Note that journaling defaults to off for 32 bit and is currently off.
2019-08-24T02:04:51.691-0300 I CONTROL  [initandlisten] **       See http://dochub.mongodb.org/core/32bit
2019-08-24T02:04:51.692-0300 I CONTROL  [initandlisten]
2019-08-24T02:04:52.249-0300 I NETWORK  [HostnameCanonicalizationWorker] Starting hostname canonicalization worker
2019-08-24T02:04:55.130-0300 W FTDC     [initandlisten] Failed to initialize Performance Counters for FTDC: WindowsPdhError: PdhExpandCounterPathW fai
led with 'O objeto especificado não foi encontrado no computador.' for counter '\Processor(_Total)\% Idle Time'
2019-08-24T02:04:55.130-0300 I FTDC     [initandlisten] Initializing full-time diagnostic data capture with directory 'c:/mongodb/data/db/diagnostic.d
ata'
2019-08-24T02:04:55.162-0300 I NETWORK  [initandlisten] waiting for connections on port 27017
```

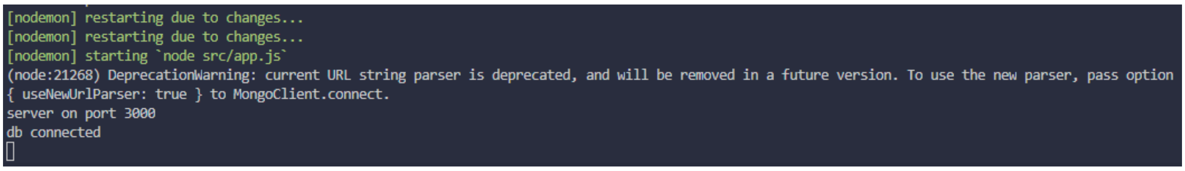
Configurando as rotas e a conexão com o BD

app.js

```
const path = require('path');
const express = require('express');
const morgan = require('morgan');
const mongoose = require('mongoose');

const app = express();

/* connection to db */

mongoose.connect('mongodb://localhost/crud-mongo')
  .then(db => console.log('db connected'))
  .catch(err => console.log(err));


/* Importing routes */

const indexRoutes = require('./routes/index');


/* settings */

// configurando a porta do servidor
app.set('port', process.env.PORT || 3000);

// configurando o diretório de views
app.set('views', path.join(__dirname, 'views'));

// definindo o motor de templates
app.set('view engine', 'ejs');


/* middlewares */

app.use(morgan('dev'));

// para entender os dados que vem de um formulário
app.use(express.urlencoded({extended: false}));


/* routes */

app.use('/', indexRoutes);
```

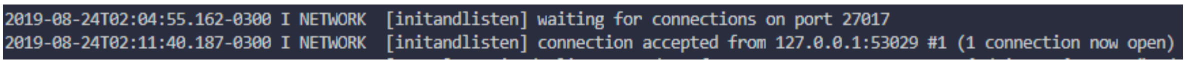
```
/* starting the server */

app.listen(app.get('port'), () => {
  console.log(`server on port ${app.get('port')}`);
});
```

Não é necessário criar a BD porque se ela não existe será criada automaticamente com o nome dado. Nesse caso, o nome do database criado será crud-mongo.



```
[nodemon] restarting due to changes...
[nodemon] restarting due to changes...
[nodemon] starting `node src/app.js`
(node:21268) DeprecationWarning: current URL string parser is deprecated, and will be removed in a future version. To use the new parser, pass option
{ useNewUrlParser: true } to MongoClient.connect.
server on port 3000
db connected
```

Uma conexão agora está aberta:



```
2019-08-24T02:04:55.162-0300 I NETWORK  [initandlisten] waiting for connections on port 27017
2019-08-24T02:11:40.187-0300 I NETWORK  [initandlisten] connection accepted from 127.0.0.1:53029 #1 (1 connection now open)
```

Configurando Models

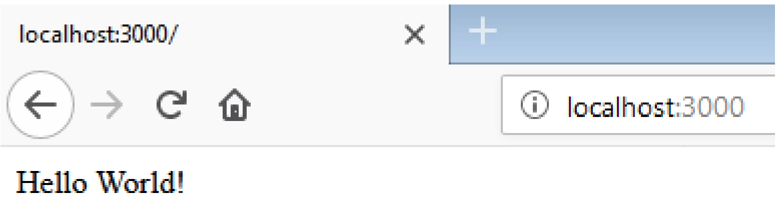
src/models/task.js

```
const mongoose = require('mongoose');
const Schema = mongoose.Schema;

const TaskSchema = Schema({
  title: String,
  description: String,
  status: {
    type: Boolean,
    default: false
  }
});

module.exports = mongoose.model('tasks', TaskSchema);
```

Configurando a rota principal

src/routes/index.js

```
const express = require('express');
const router = express.Router();

router.get('/', (req, res) => {
   res.send('Hello World!');
});

module.exports = router;
```



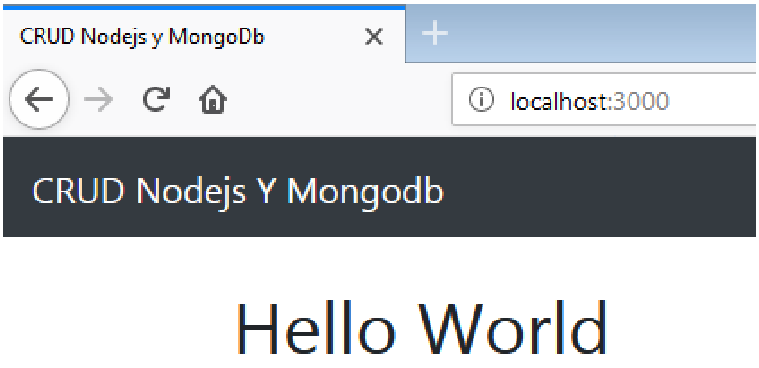
Hello World!

src/routes/index.js

```
const express = require('express');
const router = express.Router();

router.get('/', (req, res) => {
  res.render('index');
});

module.exports = router;
```

src/views/index.ejs

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>CRUD Nodejs y MongoDb</title>
    <!-- BOOTSTRAP CDN -->
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.0/css/bootstrap.min.css" integrity="sha384-9gVQ4dYFwwWSjIDZnLEWnxCjeSWFphJiwGPXr1jddIhOegiu1FwO5qRGvFXOdJZ4" crossorigin="anonymous">
  </head>
  <body>

    <nav class="navbar navbar-dark bg-dark mb-4">
      <a class="navbar-brand" href="#">CRUD Nodejs Y Mongodb</a>
    </nav>

    <div class="container">
        <h1>Hello World</h1>
    </div>

</body>
</html>
```



CRUD Nodejs Y Mongodb

Hello World

src/views/index.ejs

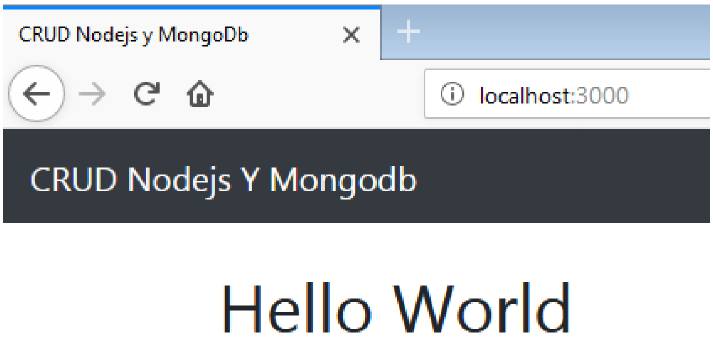
```
<% include partials/_header %>

    <div class="container">
        <h1>Hello World</h1>
    </div>

<% include partials/_footer %>
```

src/views/partials/_header.js

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>CRUD Nodejs y MongoDb</title>
    <!-- BOOTSTRAP CDN -->
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.0/css/bootstrap.min.css" integrity="sha384-
9gVQ4dYFwwWSjIDZnLEWnxCjeSWFphJiwGPXr1jddIhOegiu1FwO5qRGvFXOdJZ4"
crossorigin="anonymous">
  </head>
  <body>

    <nav class="navbar navbar-dark bg-dark mb-4">
      <a class="navbar-brand" href="#">CRUD Nodejs Y Mongodb</a>
    </nav>
```

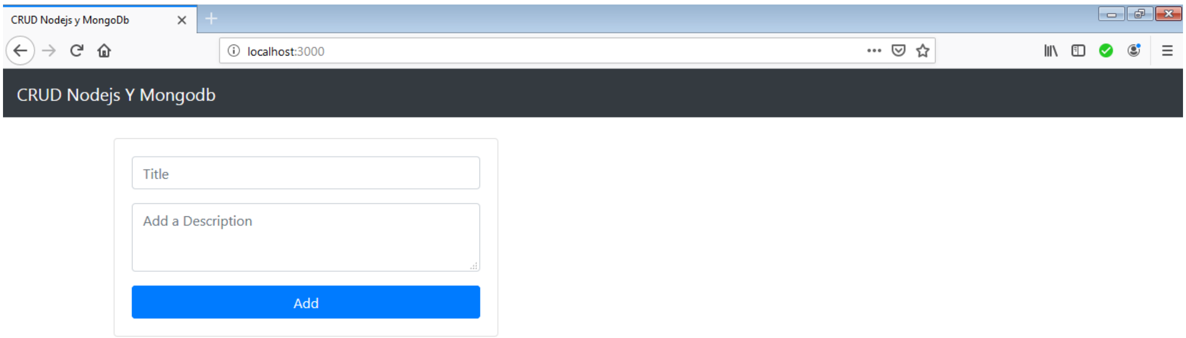
src/views/partials/_footer.js

```
</body>
</html>
```



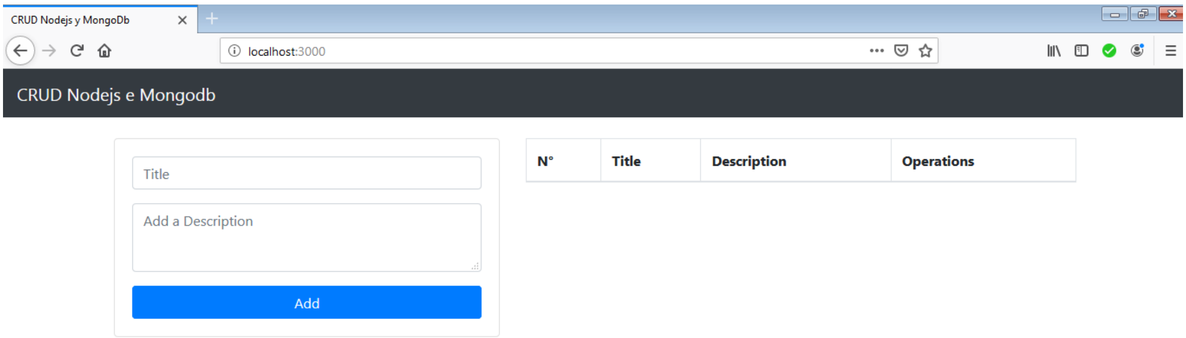
src/views/index.ejs

```
<% include partials/_header %>

  <div class="container">
    <div class="row">
      <!--FORM-->
      <div class="col-md-5">
       <div class="card">
        <div class="card-body">
         <form action="/add" method="post">
          <div class="form-group">
           <input class="form-control" type="text" name="title" placeholder="Title">
          </div>
          <div class="form-group">
           <textarea class="form-control" name="description" cols="80"
            placeholder="Add a Description"
            ></textarea>
          </div>
          <button class="btn btn-primary btn-block" type="submit">
           Add
          </button>
         </form>
        </div>
       </div>
      </div>
  </div>

<% include partials/_footer %>
```



src/views/index.ejs

```
<% include partials/_header %>

  <div class="container">
    <div class="row">
      <!--FORM-->
      <div class="col-md-5">
       <div class="card">
        <div class="card-body">
         <form action="/add" method="post">
          <div class="form-group">
           <input class="form-control" type="text" name="title"
placeholder="Title">
          </div>
          <div class="form-group">
           <textarea class="form-control" name="description" cols="80"
            placeholder="Add a Description"
            ></textarea>
          </div>
          <button class="btn btn-primary btn-block" type="submit">
           Add
          </button>
         </form>
        </div>
       </div>
      </div>

      <!-- LIST -->
      <div class="col-md-7">
        <table class="table table-bordered table-hover">
        <thead>
          <tr>
          <th>N°</th>
          <th>Title</th>
          <th>Description</th>
          <th>Operations</th>
          </tr>
        </thead>
        <tbody>
        </tbody>
        </table>
      </div>
    </div>

  </div>

<% include partials/_footer %>
```

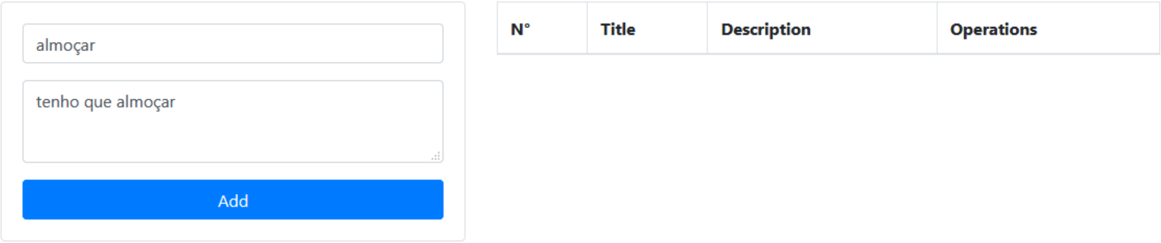


N°	Title	Description	Operations

Rota para adicionar uma tarefa

src/routes/index.js

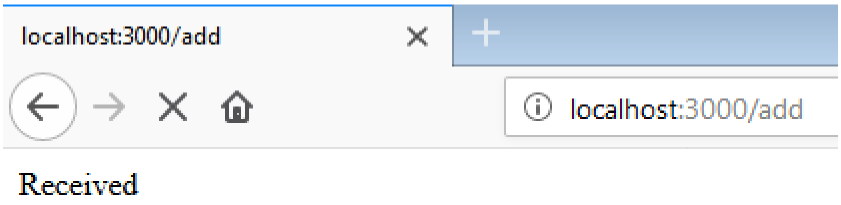
```
const express = require('express');
const router = express.Router();

const Task = require('../models/task');

router.get('/', (req, res) => {
  res.render('index');
});

router.post('/add', (req, res, next) => {
    console.log(req.body);
    res.send('Received');
  });

module.exports = router;
```



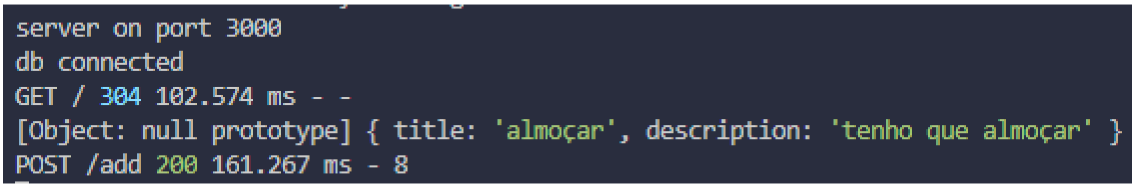
almoçar

tenho que almoçar

N°	Title	Description	Operations



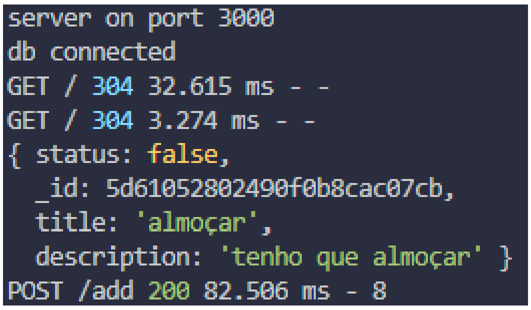
Received



```
server on port 3000
db connected
GET / 304 102.574 ms - -
[Object: null prototype] { title: 'almoçar', description: 'tenho que almoçar' }
POST /add 200 161.267 ms - 8
```

src/routes/index.js

```
const express = require('express');
const router = express.Router();

const Task = require('../models/task');

router.get('/', (req, res) => {
  res.render('index');
});

router.post('/add', (req, res, next) => {
    console.log(new Task(req.body));
    res.send('Received');
  });

module.exports = router;
```

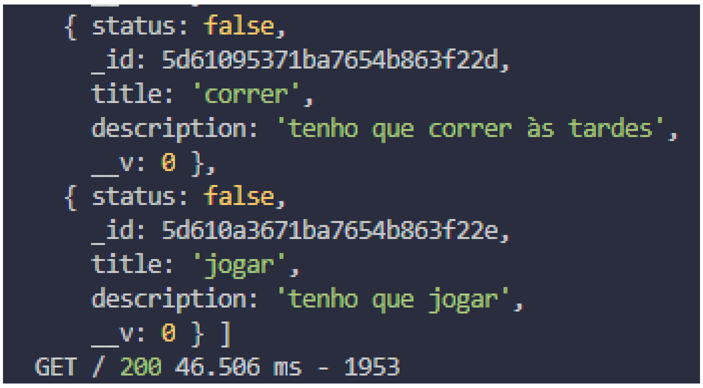


```
server on port 3000
db connected
GET / 304 32.615 ms - -
GET / 304 3.274 ms - -
{ status: false,
  _id: 5d61052802490f0b8cac07cb,
  title: 'almoçar',
  description: 'tenho que almoçar' }
POST /add 200 82.506 ms - 8
```

Criando e listando tarefas

src/routes/index.js

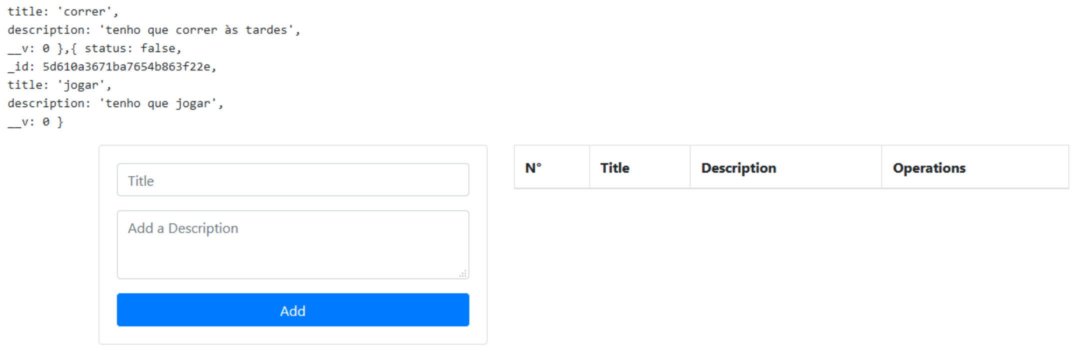
```
const express = require('express');
const router = express.Router();

const Task = require('../models/task');

router.get('/', async (req, res) => {
  const tasks = await Task.find();
  res.render('index', {
    tasks
  });
});

router.post('/add', async (req, res, next) => {
    const task = new Task(req.body);
    await task.save();
    res.redirect('/');
  });

module.exports = router;
```



```
    { status: false,
      _id: 5d61095371ba7654b863f22d,
      title: 'correr',
      description: 'tenho que correr às tardes',
      __v: 0 },
    { status: false,
      _id: 5d610a3671ba7654b863f22e,
      title: 'jogar',
      description: 'tenho que jogar',
      __v: 0 } ]
  GET / 200 46.506 ms - 1953
```

src/views/index.ejs

```
<% include partials/_header %>

    <pre><%= tasks %></pre>

    <div class="container">
        <div class="row">
            <!--FORM-->
            <div class="col-md-5">
              <div class="card">
                <div class="card-body">
                  <form action="/add" method="post">
                    <div class="form-group">
                      <input class="form-control" type="text" name="title"
placeholder="Title">
                    </div>
                    <div class="form-group">
                      <textarea class="form-control" name="description" cols="80"
                        placeholder="Add a Description"
                        ></textarea>
                    </div>
                    <button class="btn btn-primary btn-block" type="submit">
                      Add
                    </button>
                  </form>
                </div>
              </div>
            </div>

            <!-- LIST -->
            <div class="col-md-7">
                <table class="table table-bordered table-hover">
                <thead>
                    <tr>
                    <th>N°</th>
                    <th>Title</th>
                    <th>Description</th>
                    <th>Operations</th>
                    </tr>
                </thead>
                <tbody>
                </tbody>
                </table>

            </div>
            </div>

</div>

<% include partials/_footer %>
```

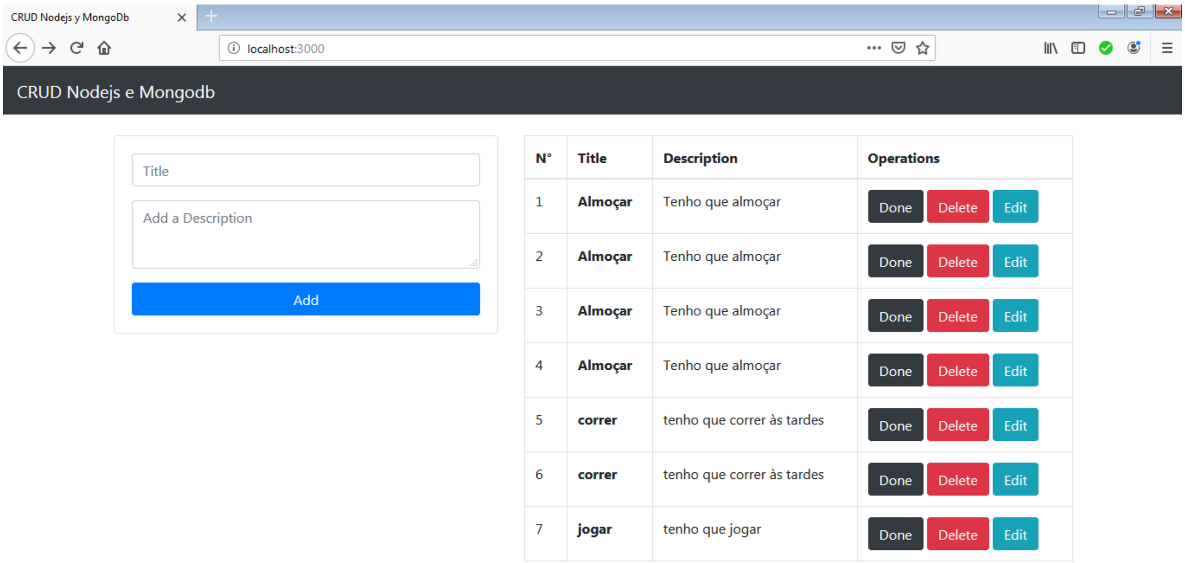


```
title: 'correr',
description: 'tenho que correr às tardes',
__v: 0 },{ status: false,
_id: 5d610a3671ba7654b863f22e,
title: 'jogar',
description: 'tenho que jogar',
__v: 0 }
```

N°	Title	Description	Operations

src/views/index.ejs

```
<% include partials/_header %>

  <div class="container">
    <div class="row">
      <!--FORM-->
      <div class="col-md-5">
       <div class="card">
        <div class="card-body">
         <form action="/add" method="post">
          <div class="form-group">
           <input class="form-control" type="text" name="title"
placeholder="Title">
          </div>
          <div class="form-group">
           <textarea class="form-control" name="description" cols="80"
            placeholder="Add a Description"
            ></textarea>
          </div>
          <button class="btn btn-primary btn-block" type="submit">
           Add
          </button>
         </form>
        </div>
       </div>
      </div>

      <!-- LIST -->
      <div class="col-md-7">
        <table class="table table-bordered table-hover">
        <thead>
          <tr>
          <th>N°</th>
          <th>Title</th>
          <th>Description</th>
          <th>Operations</th>
          </tr>
        </thead>
        <tbody>
        <% for(var i=0; i < tasks.length; i++) { %>
          <tr>
            <td><%= i + 1%></td>
            <td><strong><%= tasks[i].title %></strong></td>
            <td><%= tasks[i].description %></td>
            <td>
            <a
              class="<%= tasks[i].status ? 'btn btn-success' : 'btn btn-
dark'%>"
              href="/turn/<%= tasks[i]._id %>">Done</a>
            <a href="/delete/<%= tasks[i]._id %>"
              class="btn btn-danger">
              Delete
            </a>
            <a href="/edit/<%= tasks[i]._id %>" class="btn btn-info">
              Edit
            </a>
            </td>
          </tr>
          <% } %>
        </tbody>
        </table>

      </div>
      </div>

</div>

<% include partials/_footer %>
```



CRUD Nodejs e Mongodb

Title

Add a Description

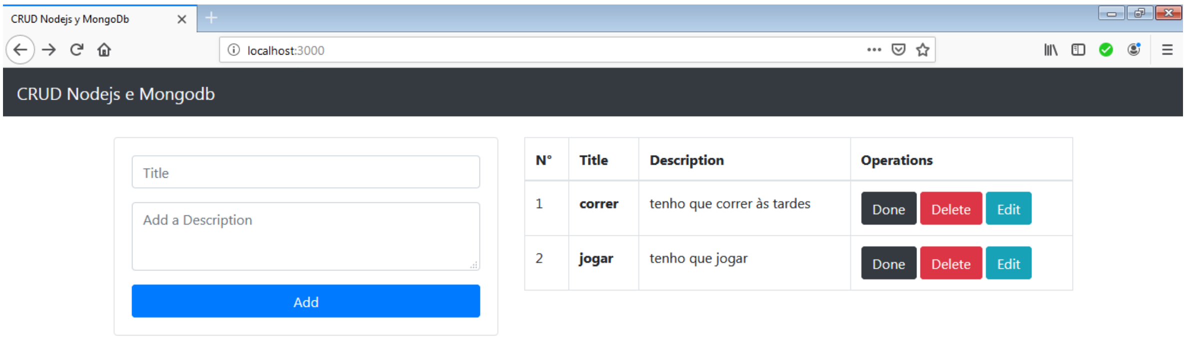
Add

N°	Title	Description	Operations
1	**Almoçar**	Tenho que almoçar	Done Delete Edit
2	**Almoçar**	Tenho que almoçar	Done Delete Edit
3	**Almoçar**	Tenho que almoçar	Done Delete Edit
4	**Almoçar**	Tenho que almoçar	Done Delete Edit
5	**correr**	tenho que correr às tardes	Done Delete Edit
6	**correr**	tenho que correr às tardes	Done Delete Edit
7	**jogar**	tenho que jogar	Done Delete Edit

Deletando tarefas

src/routes/index.js

```
const express = require('express');
const router = express.Router();

const Task = require('../models/task');

router.get('/', async (req, res) => {
  const tasks = await Task.find();
  console.log(tasks);
  res.render('index', {
    tasks
  });
});

router.post('/add', async (req, res, next) => {
    const task = new Task(req.body);
    await task.save();
    res.redirect('/');
  });

  router.get('/delete/:id', async (req, res, next) => {
    let { id } = req.params;
    await Task.remove({_id: id});
    res.redirect('/');
  });

module.exports = router;
```

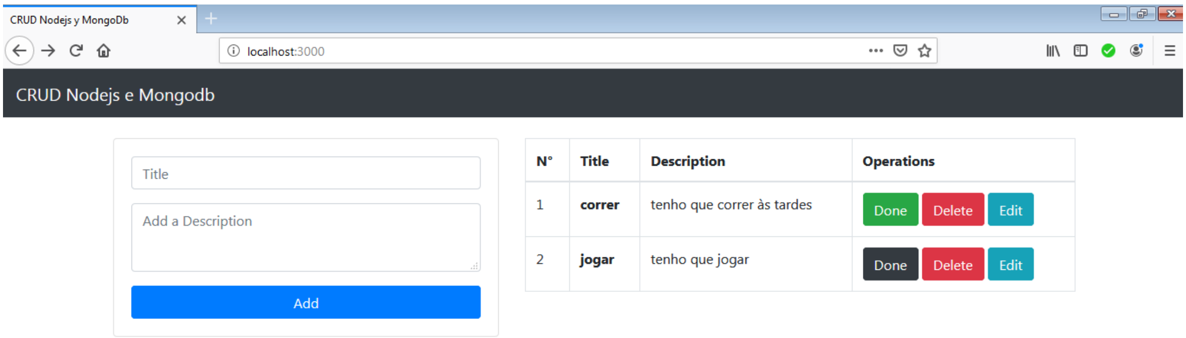


N°	Title	Description	Operations
1	**correr**	tenho que correr às tardes	Done Delete Edit
2	**jogar**	tenho que jogar	Done Delete Edit

Marcando tarefas como realizadas ou não realizadas

src/routes/index.js

```
const express = require('express');
const router = express.Router();

const Task = require('../models/task');

router.get('/', async (req, res) => {
  const tasks = await Task.find();
  console.log(tasks);
  res.render('index', {
    tasks
  });
});

router.post('/add', async (req, res, next) => {
    const task = new Task(req.body);
    await task.save();
    res.redirect('/');
  });

  router.get('/turn/:id', async (req, res, next) => {
    let { id } = req.params;
    const task = await Task.findById(id);
    task.status = !task.status;
    await task.save();
    res.redirect('/');
  });

  router.get('/delete/:id', async (req, res, next) => {
    let { id } = req.params;
    await Task.remove({_id: id});
    res.redirect('/');
  });

module.exports = router;
```



N°	Title	Description	Operations
1	correr	tenho que correr às tardes	Done Delete Edit
2	jogar	tenho que jogar	Done Delete Edit

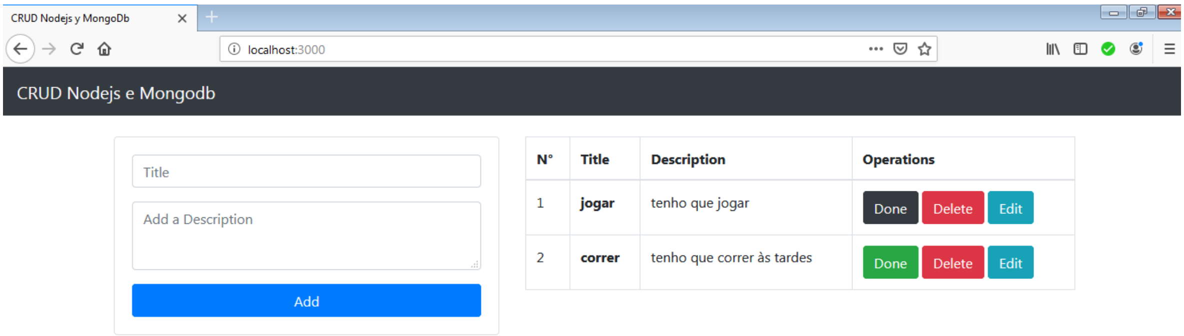
Editando uma tarefa

src/routes/index.js

```
const express = require('express');
const router = express.Router();

const Task = require('../models/task');

router.get('/', async (req, res) => {
  const tasks = await Task.find();
  console.log(tasks);
  res.render('index', {
    tasks
  });
});

router.post('/add', async (req, res, next) => {
    const task = new Task(req.body);
    await task.save();
    res.redirect('/');
  });

  router.get('/turn/:id', async (req, res, next) => {
    let { id } = req.params;
    const task = await Task.findById(id);
    task.status = !task.status;
    await task.save();
    res.redirect('/');
  });

  router.get('/edit/:id', async (req, res, next) => {
    const task = await Task.findById(req.params.id);
    console.log(task)
    res.render('edit', { task });
  });
  router.post('/edit/:id', async (req, res, next) => {
    const { id } = req.params;
    await Task.update({_id: id}, req.body);
    res.redirect('/');
  });

  router.get('/delete/:id', async (req, res, next) => {
    let { id } = req.params;
    await Task.remove({_id: id});
    res.redirect('/');
  });

module.exports = router;
```

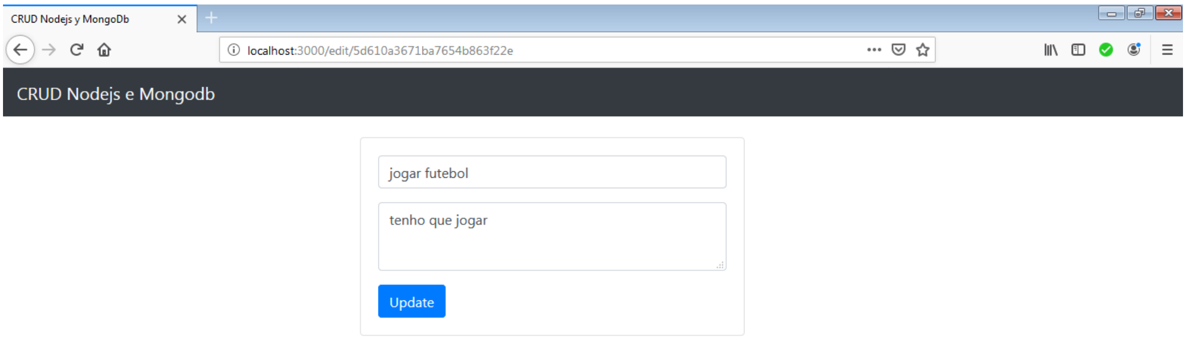
src/views/edit.ejs

```
<% include partials/_header %>

<div class="container">
  <div class="row">
    <!--FORM-->
    <div class="col-md-5 offset-md-3">
      <div class="card">
        <div class="card-body">
          <form action="/edit/<%= task._id %>" method="post">
            <div class="form-group">
              <input class="form-control" type="text" name="title"
placeholder="Title" value="<%= task.title %>">
            </div>
            <div class="form-group">
              <textarea class="form-control" name="description" cols="80"
                placeholder="Add a Description"
                ><%= task.description %></textarea>
            </div>
            <button class="btn btn-primary" type="submit">
              Update
            </button>
          </form>
        </div>
      </div>
    </div>
  </div>
</div>

<% include partials/_footer%>
```



CRUD Nodejs e Mongodb

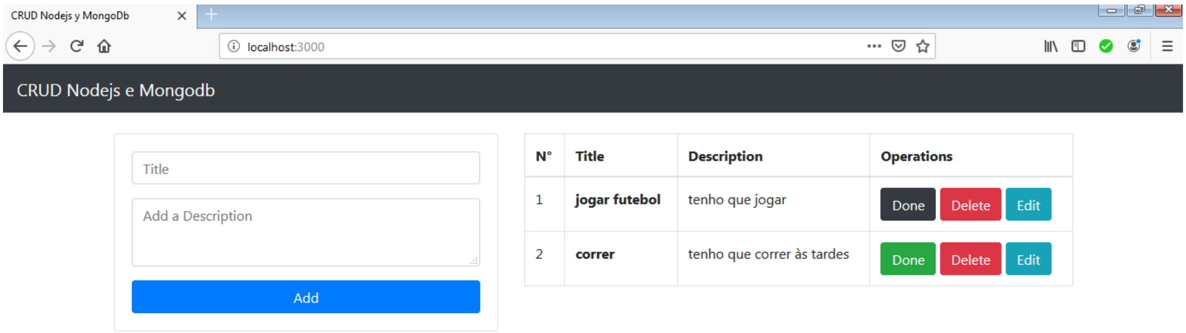
N°	Title	Description	Operations
1	**jogar**	tenho que jogar	Done Delete Edit
2	**correr**	tenho que correr às tardes	Done Delete Edit



CRUD Nodejs e Mongodb

jogar futebol

tenho que jogar

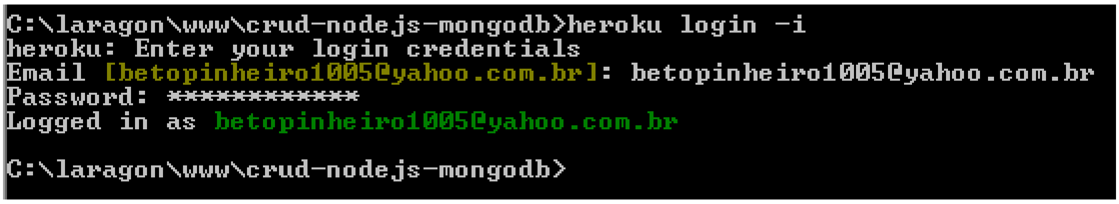


CRUD Nodejs e Mongodb

N°	Title	Description	Operations
1	**jogar futebol**	tenho que jogar	Done Delete Edit
2	**correr**	tenho que correr às tardes	Done Delete Edit

String de conexão

mongodb+srv://betopinheiro1005:<password>@cluster-nodejs-mongodb-ulvuw.mongodb.net/crud-mongo?retryWrites=true&w=majority



```
C:\laragon\www\crud-nodejs-mongodb>heroku login -i
heroku: Enter your login credentials
Email [betopinheiro1005@yahoo.com.br]: betopinheiro1005@yahoo.com.br
Password: *************
Logged in as betopinheiro1005@yahoo.com.br

C:\laragon\www\crud-nodejs-mongodb>
```