



Despliegue HEROKU CRUD Server

Por Daniel Duque

Heroku es una plataforma de servicios en la nube (conocidos como PaaS o Platform as a Service), cuya popularidad ha crecido en los últimos años debido a su facilidad de uso y versatilidad para distintos proyectos.

En el siguiente ejercicio, se desplegará el ToDo App MEVN CRUD en la plataforma Heroku cuyos servicios serán consumidos desde la nube.

Prerrequisitos e instrucciones:

Inicialmente, se debe desempaquetar el CRUD con el cual se va a trabajar en el desarrollo de esta actividad.

La app ToDo MEVN CRUD, se encuentra disponible en el siguiente enlace:

https://github.com/luisguillermomolero/MisionTIC2022/tree/main/Modulo3_Web_MisionTIC2022_Main/Semana_3/02_CRUD_MEVN_1

Para instalarla, ejecute en una terminal dentro de la carpeta /server el siguiente comando:

```
npm install
```

Luego, ejecute el siguiente comando en una terminal dentro de la carpeta /client

```
npm install
```

Opcional: Para ejecutar la app en local y hacer pruebas antes del despliegue en la nube, ejecute dentro de la carpeta /server, **en dos terminales separadas**, los siguientes comandos:

```
mongod
```

Luego

```
npm install -g nodemon
```

Seguidamente,

```
nodemon
```



Para desplegar el cliente en local, ejecute en una terminal dentro de la carpeta /client el siguiente comando:

```
npm run serve
```

Instalación y configuración de git

Para descargar git:

<https://git-scm.com/downloads>

Manual de instalación de git:

https://github.com/luisguillermomolero/MisionTIC2022/blob/main/Modulo4_Web_MisionTIC2022_Main/Semana_1/02_GIT_Installation.pdf

Para verificar que git esta correctamente instalado, en una terminal ejecute:

```
git --version
```

Instalación de heroku CLI

<https://devcenter.heroku.com/articles/heroku-cli#download-and-install>

Para verificar que Heroku está correctamente instalado, en una terminal ejecute:

```
heroku -v
```

Crear una cuenta en la página de Heroku

<https://signup.heroku.com/>

Ejecute en una terminal dentro de la carpeta /server el siguiente comando:

```
heroku login
```

éste, pedirá presionar una tecla para continuar y luego abrirá el navegador para autenticarse. También puede ejecutar el comando “heroku login -i” para utilizar la terminal.

Editar el código del CRUD

Sustituya en el objeto “script” del archivo /server/package.json la siguiente línea de código:

```
"scripts": {  
  "test": "echo \"Error: no test specified\" && exit 1",  
  "start": "node index.js"  
},
```



Luego, en el archivo /server/index.js inserte el siguiente código luego de la sentencia "app.use(cors());". A saber:

```
app.use((req, res, next)=>{
  res.header("Access-Control-Allow-Origin", "*");
  res.header("Access-Control-Allow-Headers", "Origin, X-Requested-With, Content-Type, Accept");
  res.header('Access-Control-Allow-Methods', 'GET, POST, OPTIONS, PUT, DELETE');
  res.header('Allow', 'GET, POST, OPTIONS, PUT, DELETE');
  next();
});
```

Seguidamente, para la conexión con su colección mongoose, cambie la dirección actual por la dirección de su colección en mongo atlas, de la forma:

```
mongoose
  .connect(
    "mongodb+srv://lmolero:1126254560@cluster0.bpms5.mongodb.net/mevn-crud-deploy?retryWrites=true&w=majority",
    {
      useNewUrlParser: true,
      useUnifiedTopology: true,
    }
  )
  .then(() => console.log("Database connected!"))
  .catch((err) => console.log(err));

var db = mongoose.connection;
db.once("open", () => {
  console.log("Connected to mongoDB");
});
```

A continuación, sustituya el código en la API "// add todo" por el siguiente código:

```
// add todo
app.post("/todo/add", (req, res) => {
  let newTodo = new todoModel();
  newTodo.title = req.body.todo;
  newTodo.completed = false;
  newTodo.save(err => {
    if (err) {
      console.log(err);
      // res.send("Error while adding Todo", err);
      res.send(err);
    } else {
      console.log(newTodo);
      res.send("Todo added");
    }
  });
});
```



De igual forma, sustituya el código en la API “// completed” por el siguiente código:

```
// completed
app.get("/todo/completed", (req, res) => {
  todoModel.find({ completed: true }, (err, todos) => {
    if (err) {
      // res.send("Error while fetching Todos", err);
      res.send(err);
    } else {
      res.json(todos);
    }
  });
});
```

A continuación, sustituya el código en la API “// uncompleted” por el siguiente código:

```
// uncompleted
app.get("/todo/uncompleted", (req, res) => {
  todoModel.find({ completed: false }, (err, todos) => {
    if (err) {
      // res.send("Error while fetching Todos", err);
      res.send(err);
    } else {
      res.json(todos);
    }
  });
});
```

Seguidamente, corresponde dejar a la app de heroku escoger un puerto libre para trabajar, normalmente se usa el “puerto 3000”.

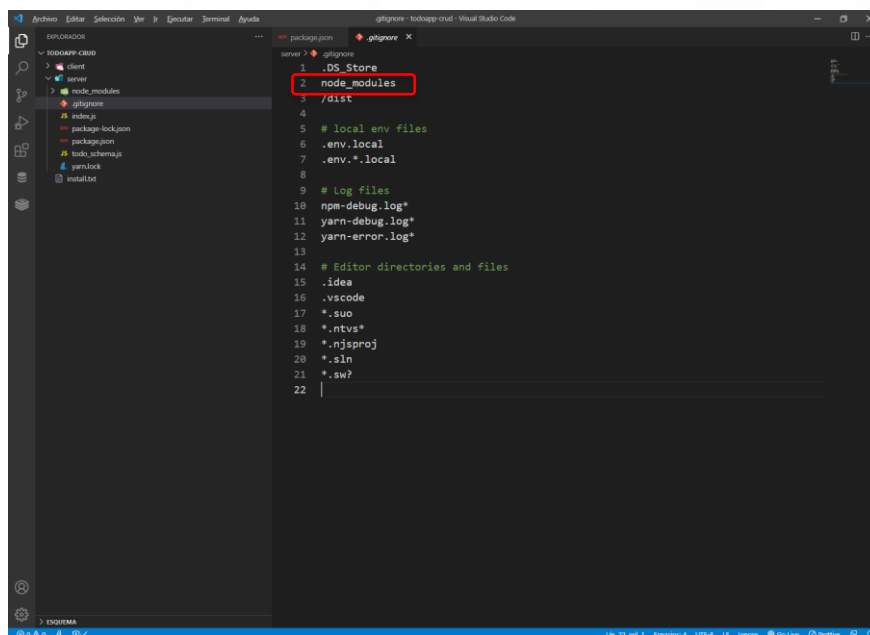
Para lograr este punto, se debe agregar/modificar la línea “app.listen” del archivo /server/index.js por el siguiente código:

```
app.set("PORT", process.env.PORT || 3000);

app.listen(app.get("PORT"), () => {
  console.log(`Server started on port: ${app.get("PORT")}`);
});
```



Luego, se agrega dentro del archivo `/server/.gitignore` el directorio `/node_modules`, de la forma:



IMPORTANTE: Si el proyecto no cuenta con un repositorio GIT inicializado, se debe INICIALIZAR antes de continuar, para ello, dentro de la carpeta `/server` se ejecuta en una terminal el siguiente comando:

```
git init
```

A continuación, se debe confirmar los cambios utilizando los siguientes comandos en una terminal dentro de la carpeta `/server`, de la forma:

```
git add .
```

Luego, remueva el archivo `/server/yarn.lock` con el siguiente comando

```
git rm -f yarn.lock
```

posteriormente

```
git commit -m "deploy backend to heroku server"
```

En este punto, está todo listo para realizar el despliegue del backend; **recuerde tener cuidado con las conexiones a bases de datos.**



Crear una aplicación Heroku

Primero, debe ubicarse dentro de la carpeta /server y ejecutar en una terminal el siguiente comando:

```
heroku create
```

En este punto, la terminal responderá con la URL de la app creada; ahora solo hace falta hacer un push a heroku e iniciara la instalación del recurso remoto. Para ello, ejecute en una terminal dentro de la carpeta /server el siguiente comando:

```
git push heroku master
```

Finalmente, con el siguiente comando se puede visualizar el proyecto en ejecución.

```
heroku open
```

<https://guarded-ocean-09625.herokuapp.com>

