



Despliegue HEROKU CRUD Front

Por Daniel Duque

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

```
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

Un punto importante es dejar las peticiones realizadas desde el front, a la URL correspondiente del Back, para eso se deben modificar las peticiones del archivo App.vue y ajustar las URLs, en este caso se cambian los localhost por la URL <https://guarded-ocean-09625.herokuapp.com>

```
<script>
import axios from "axios";
export default {
  name: "App",
  data: () => ({
    newToDo: "",
    uncompletedTodos: [],
    completedTodos: [],
  }),
  methods: {
    addToDo() {
      axios
        .post("https://guarded-ocean-09625.herokuapp.com/todo/add", {
```



```
    todo: this.newTodo,
  })
  .then((response) => {
    this.message = response.data;
    console.log(response.data);
    this.list();
  });
  this.newTodo = "";
},
completeTodo(todoID) {
  axios
    .post(
      "https://guarded-ocean-09625.herokuapp.com/todo/complete/" + todoID
    )
    .then((response) => {
      console.log(response.data);
      this.list();
    });
},
deleteTodo(todoID) {
  axios
    .delete("https://guarded-ocean-09625.herokuapp.com/todo/" + todoID)
    .then((response) => {
      console.log(response.data);
      this.list();
    });
},
list() {
  // fetch uncompleted task
  axios
    .get("https://guarded-ocean-09625.herokuapp.com/todo/uncompleted")
    .then((response) => (this.uncompletedTodos = response.data))
    .catch((error) => console.log(error));
  // fetch completed task
  axios
    .get("https://guarded-ocean-09625.herokuapp.com/todo/completed")
    .then((response) => (this.completedTodos = response.data))
    .catch((error) => console.log(error));
},
},
created() {
  this.list();
},
};
</script>
```

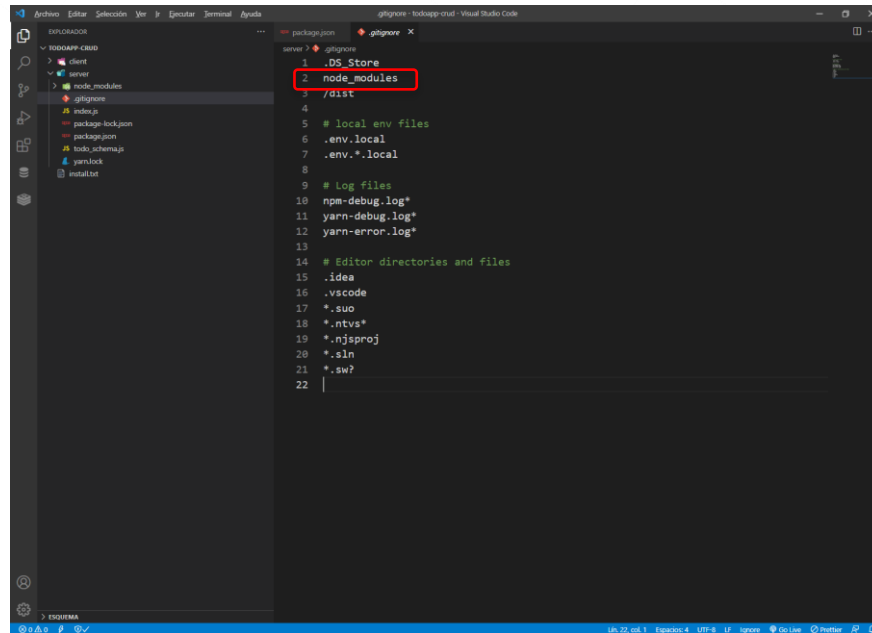
Debido a algunas características propias de VueJS, es importante definir un archivo estático de inicio, para ello es necesario crear el archivo llamado /client/static.json con el siguiente texto:

```
{
  "root": "dist",
  "clean_urls": true,
```



```
"routes": {  
  "/*": "index.html"  
}
```

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 frontend to heroku server"
```

En este punto está todo listo para realizar el despliegue del frontend, **recuerde tener cuidado con las URLs de las peticiones.**



Crear una aplicación Heroku

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

```
heroku create
```

Luego

```
heroku buildpacks:add heroku/nodejs
```

y finalmente

```
heroku buildpacks:add https://github.com/heroku/heroku-buildpack-static
```

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://warm-temple-83695.herokuapp.com/>

