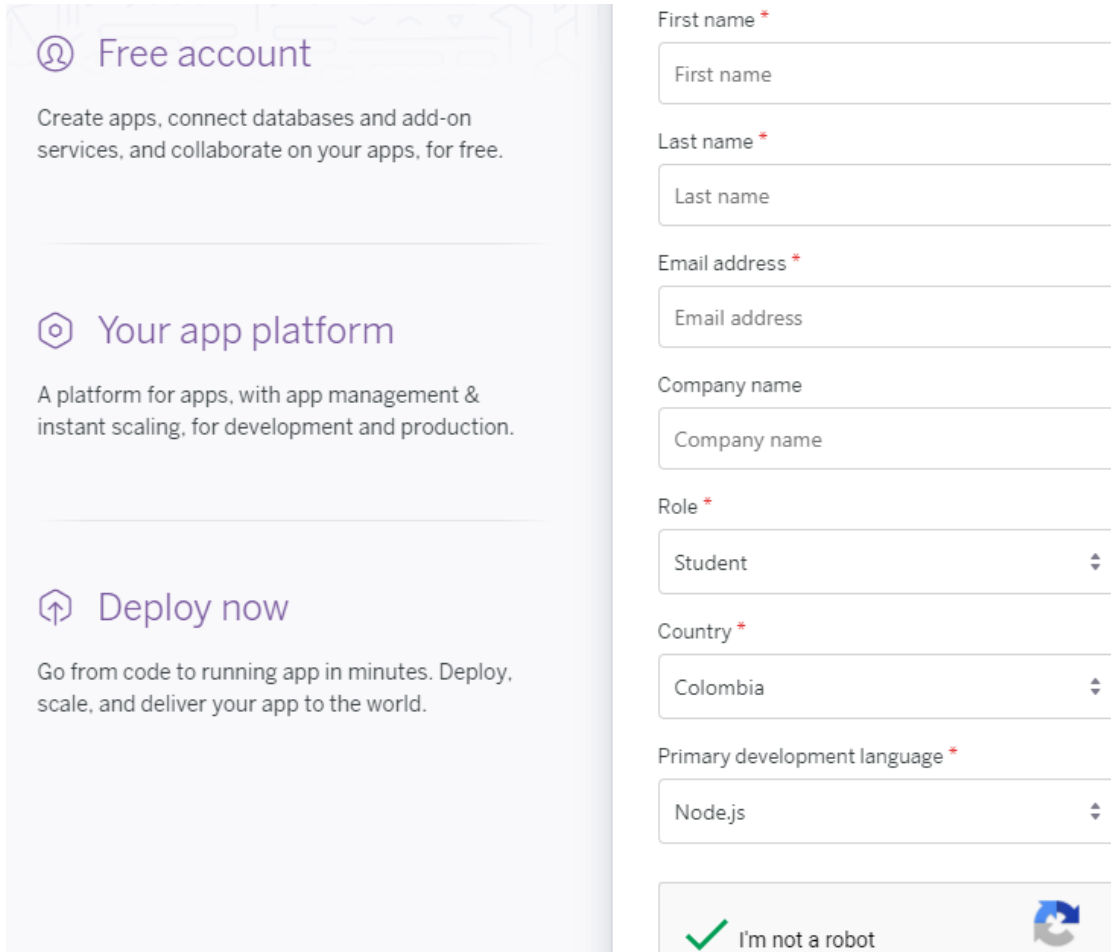


Instalación Heroku

1. Ingresar a la página <https://signup.heroku.com/login> y llenar el formulario



The image shows a screenshot of the Heroku signup page. On the left, there are three sections: 'Free account' with a description 'Create apps, connect databases and add-on services, and collaborate on your apps, for free.', 'Your app platform' with a description 'A platform for apps, with app management & instant scaling, for development and production.', and 'Deploy now' with a description 'Go from code to running app in minutes. Deploy, scale, and deliver your app to the world.' On the right, there is a form with the following fields: 'First name *' (text input), 'Last name *' (text input), 'Email address *' (text input), 'Company name' (text input), 'Role *' (dropdown menu with 'Student' selected), 'Country *' (dropdown menu with 'Colombia' selected), 'Primary development language *' (dropdown menu with 'Node.js' selected), and a checkbox 'I'm not a robot' with a reCAPTCHA image.

Free account

Create apps, connect databases and add-on services, and collaborate on your apps, for free.

Your app platform

A platform for apps, with app management & instant scaling, for development and production.

Deploy now

Go from code to running app in minutes. Deploy, scale, and deliver your app to the world.

First name *

First name

Last name *

Last name

Email address *

Email address

Company name

Company name

Role *

Student

Country *

Colombia

Primary development language *

Node.js

☒ I'm not a robot

2. Verificar correo para activar cuenta



Thanks for signing up with Heroku! You must follow this link within 30 days of registration to activate your account:

<https://id.heroku.com/account/accept/10539856/25734c557d7f39417d631484ab26d6ba>

Have fun, and don't hesitate to contact us with your feedback.

The Heroku Team
<https://heroku.com>

3. Crear contraseña

A screenshot of the Heroku password creation form. It has a dark blue border. Inside, there are two input fields for password and confirmation, both with red asterisks. Below them are three green checkmarks indicating password requirements. At the bottom is a blue button with white text.

Create a new password *

Password confirmation *

Password requirements

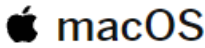
- ✓ Must be a minimum of 8 characters.
- ✓ Must contain letters, numbers, and symbols.
- ✓ Passwords must match.

☐ I would like to receive occasional updates about new product releases, technical content, tips, and more to help optimize my Heroku experience. I can unsubscribe at any time.


SET PASSWORD AND LOG IN

4. Iniciar sesión y aceptar términos y servicios

5. Ingresar a la página de Heroku CLI <https://devcenter.heroku.com/articles/heroku-cli>, descargar e instalar el software según el sistema operativo.



```
$ brew tap heroku/brew && brew install heroku
```




Windows

Download the appropriate installer for your Windows installation:

64-bit installer

32-bit installer




Ubuntu 16+

Run the following from your terminal:

```
$ sudo snap install --classic heroku
```

Configuración y despliegue del Backend

1. Crear una nueva app en Heroku




Create a new app

Create your first app and deploy your code to a running dyno.

Create new app



2. Ingresar nombre con el cuál se identificará la aplicación y será parte de la url

App name



tienda-56-api is available

Choose a region

 United States 

Add to pipeline...

Create app

En el código fuente del backend...

3. Si no existe, crear archivo .gitignore en la raíz del proyecto. Agregar línea:

```
node_modules/
```

4. En el archivo package.json, sección "scripts" agregar la siguiente línea:

```
"start": "babel-node app.js --exec",
```

5. También en el archivo package.json, sección "dependencies" agregar la siguiente línea:

```
"babel-cli": "^6.26.0",
```

Es decir, el archivo quedará similar al siguiente:

```
{
  "name": "api_tienda",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "start": "babel-node app.js --exec",
    "dev": "nodemon app.js",
```

```

    "devbabel": "nodemon app.js --exec babel-node"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "dependencies": {
    "babel-cli": "^6.26.0",
    "connect-history-api-fallback": "^1.6.0",
    "cors": "^2.8.5",
    "express": "^4.17.1",
    "mongoose": "^6.0.10",
    "morgan": "^1.10.0"
  },
  "devDependencies": {
    "@babel/cli": "^7.15.7",
    "@babel/core": "^7.15.5",
    "@babel/node": "^7.15.4",
    "@babel/preset-env": "^7.15.6"
  }
}

```

Volver a la página de Heroku para ejecutar algunos de los comandos propuestos para el despliegue...

If you haven't already, log in to your Heroku account and follow the prompts to create a new SSH public key.

```
$ heroku login
```

Create a new Git repository

Initialize a git repository in a new or existing directory

```

$ cd my-project/
$ git init
$ heroku git:remote -a tienda-56-api

```

Deploy your application

Commit your code to the repository and deploy it to Heroku using Git.

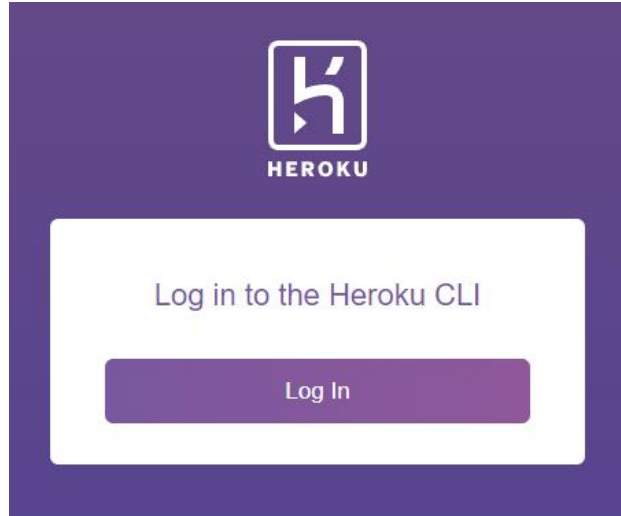
```

$ git add .
$ git commit -am "make it better"
$ git push heroku master

```

6. En la consola o terminal Iniciar sesión de Heroku con la siguiente instrucción, para luego oprimir la tecla enter e iniciar sesión desde el navegador web.

```
heroku login
```



7. Crear base git en el proyecto si no existe

```
git init
```

8. Desde el terminal o consola (No administrador) ubicado en la carpeta del proyecto backend ejecutar la siguiente línea:

```
heroku config:set NPM_CONFIG_PRODUCTION=false
```

9. Desde el mismo terminal o consola (No administrador) ejecutar la siguiente línea para reinstalar todas las dependencias estipuladas en el package.json:

```
npm install
```

10. Enlazar la carpeta local con el repositorio de Heroku mediante la instrucción de git remote:

```
heroku git:remote -a tienda-56-api
```

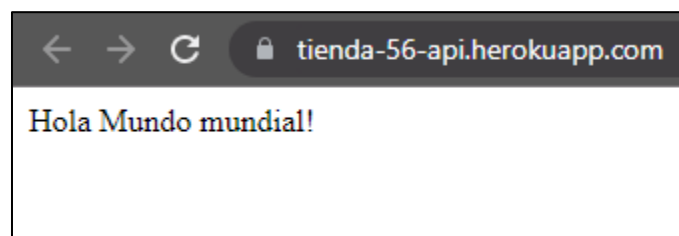
(tienda-56-api será el nombre dado a la aplicación)

11. Y a continuación se realiza en la consola el proceso de despliegue proporcionado por Heroku:

```
git add .  
git commit -am "backend funcional"  
git push heroku master
```

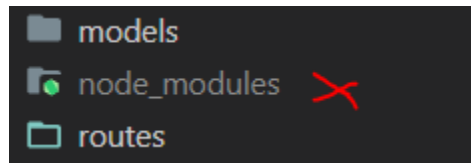
```
remote: -----> Build  
remote:  
remote: -----> Caching build  
remote:      - node_modules  
remote:  
remote: -----> Pruning devDependencies  
remote:      Skipping because NPM_CONFIG_PRODUCTION is 'false'  
remote:  
remote: -----> Build succeeded!  
remote: -----> Discovering process types  
remote:      Procfile declares types      -> (none)  
remote:      Default types for buildpack -> web  
remote:  
remote: -----> Compressing...  
remote:      Done: 41.8M  
remote: -----> Launching...  
remote:      Released v4  
remote:      https://tienda-56-api.herokuapp.com/ deployed to Heroku  
remote:  
remote: Verifying deploy... done.  
To https://git.heroku.com/tienda-56-api.git  
* [new branch]      master -> master
```

12. Probar en el navegador, utilizando la url generada.



NOTA:

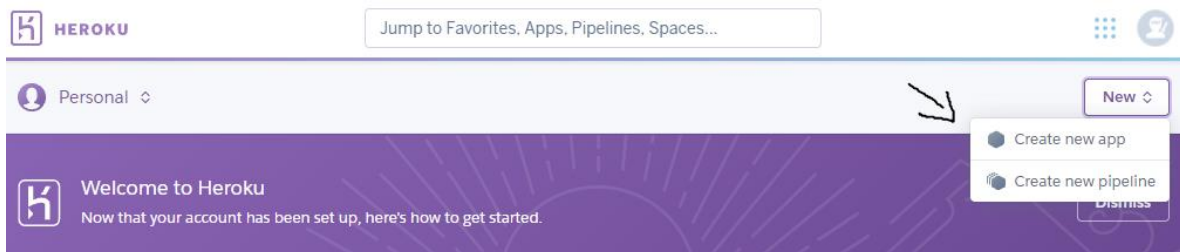
Si el despliegue es erróneo, Eliminar en su totalidad la carpeta `node_modules` del proyecto (Con permisos de administrador si lo requiere).



Y repetir pasos del 8 al 12.

Configuración y despliegue del Frontend

1. Crear nueva app en Heroku





2. Ingresar nombre con el cuál se identificará la aplicación y será parte de la url

App name

tienda--56 

tienda--56 is available

Choose a region

 United States 

Add to pipeline...

Create app

3. Desde el terminal o consola (No administrador) ubicado en la carpeta del proyecto frontend ejecutar la siguiente línea para construir la aplicación:

```
npm run build
```

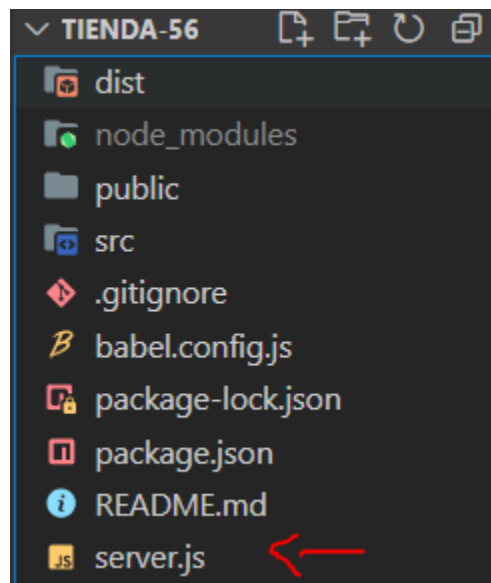
4. Instalar express para apoyar en el arranque de la aplicación desplegada, mediante la siguiente instrucción en consola o terminal con permisos de administrador:

```
npm install express serve-static --save
```

5. Ir al archivo .gitignore para comentar o eliminar la línea dist.

```
.DS_Store  
node_modules  
# /dist
```

6. Crear un nuevo archivo en la raíz del proyecto llamado server.js



7. Agregar el siguiente contenido al archivo server.js

```
const express = require('express')  
const serveStatic = require('serve-static')  
const path = require('path')  
const app = express()
```

```
app.use('/', serveStatic(path.join(__dirname, '/dist')))  
const port = process.env.PORT || 8080  
app.listen(port)
```

8. En el archivo package.json, sección “scripts” agregar la siguiente línea:

```
"start": "node server.js"
```

Es decir, el archivo quedará similar al siguiente:

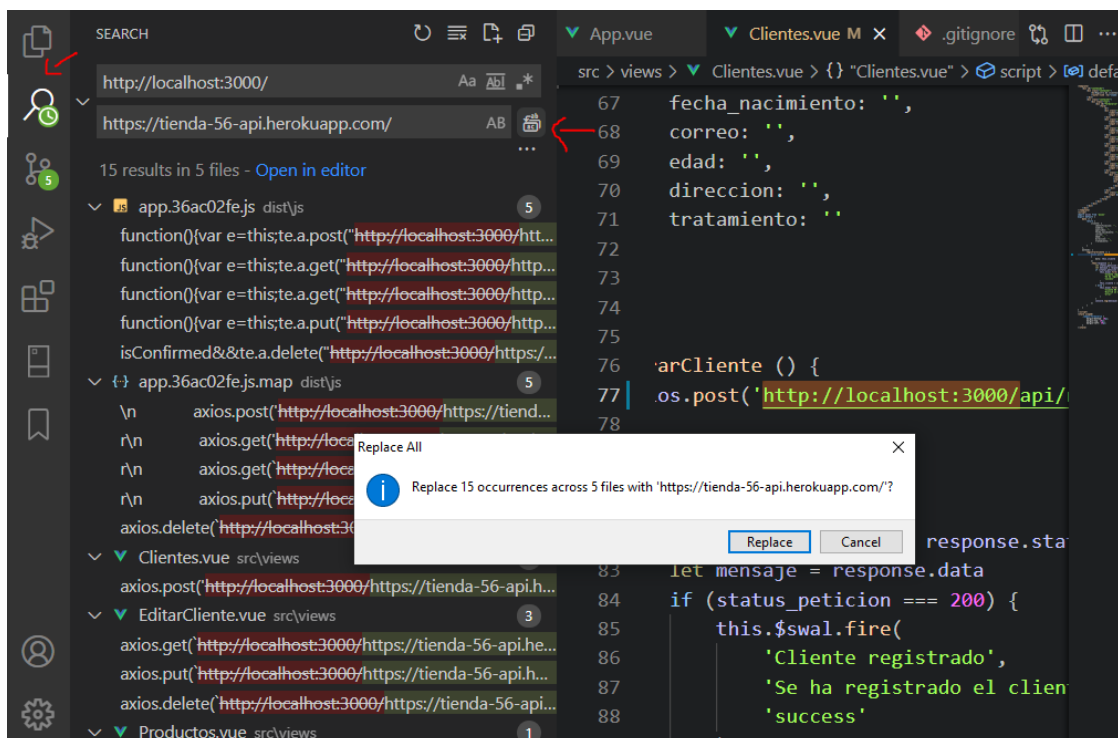
```
{  
  "name": "tienda-56",  
  "version": "0.1.0",  
  "private": true,  
  "scripts": {  
    "serve": "vue-cli-service serve",  
    "build": "vue-cli-service build",  
    "lint": "vue-cli-service lint",  
    "start": "node server.js"  
  },  
  "dependencies": {  
    "axios": "^0.21.4",  
    "core-js": "^3.6.5",  
    "express": "^4.17.1",  
    "serve-static": "^1.14.1",  
    "vue": "^3.0.0",  
    "vue-router": "^4.0.11",  
    "vue-sweetalert2": "^5.0.2"  
  },  
  "devDependencies": {  
    "@vue/cli-plugin-babel": "~4.5.0",  
    "@vue/cli-plugin-eslint": "~4.5.0",  
    "@vue/cli-service": "~4.5.0",  
    "@vue/compiler-sfc": "^3.0.0",  
    "babel-eslint": "^10.1.0",  
    "eslint": "^6.7.2",  
    "eslint-plugin-vue": "^7.0.0"  
  },  
  "eslintConfig": {
```

```

"root": true,
"env": {
  "node": true
},
"extends": [
  "plugin:vue/vue3-essential",
  "eslint:recommended"
],
"parserOptions": {
  "parser": "babel-eslint"
},
"rules": {}
},
"browserslist": [
  "> 1%",
  "last 2 versions",
  "not dead"
]
}

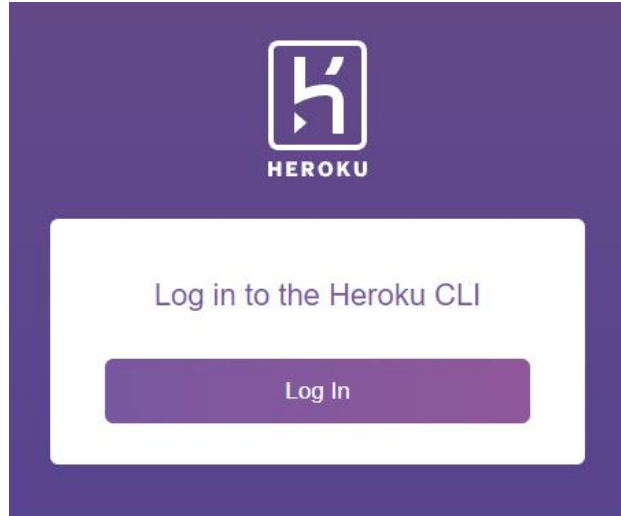
```

- Reemplazar todas las rutas de <http://localhost:3000/> por la ruta o url del backend desplegado en Heroku. Se puede realizar con la herramienta de Visual Studio Code



10. En la consola o terminal Iniciar sesión de Heroku con la siguiente instrucción, para luego oprimir la tecla enter e iniciar sesión desde el navegador web.

```
heroku login
```



11. Crear base git en el proyecto si no existe

```
git init
```

12. Enlazar la carpeta local con el repositorio de Heroku mediante la instrucción de git remote:

```
heroku git:clone -a tienda-56
```

(tienda-56 será el nombre dado a la aplicación)

13. Y a continuación se realiza en la consola el proceso de despliegue proporcionado por Heroku:

```
git add .  
git commit -am "frontend funcional"  
git push heroku master
```

```
remote: -----> Build succeeded!
remote: -----> Discovering process types
remote:   Procfile declares types      -> (none)
remote:   Default types for buildpack -> web
remote:
remote: -----> Compressing...
remote:   Done: 46.4M
remote: -----> Launching...
remote:   Released v4
remote:   https://tienda-56.herokuapp.com/ deployed to Heroku
remote:
remote: Verifying deploy... done.
To https://git.heroku.com/tienda-56.git
   faaf7c7..473c662  master -> master
```

14. Probar en el navegador, utilizando la url generada.

