

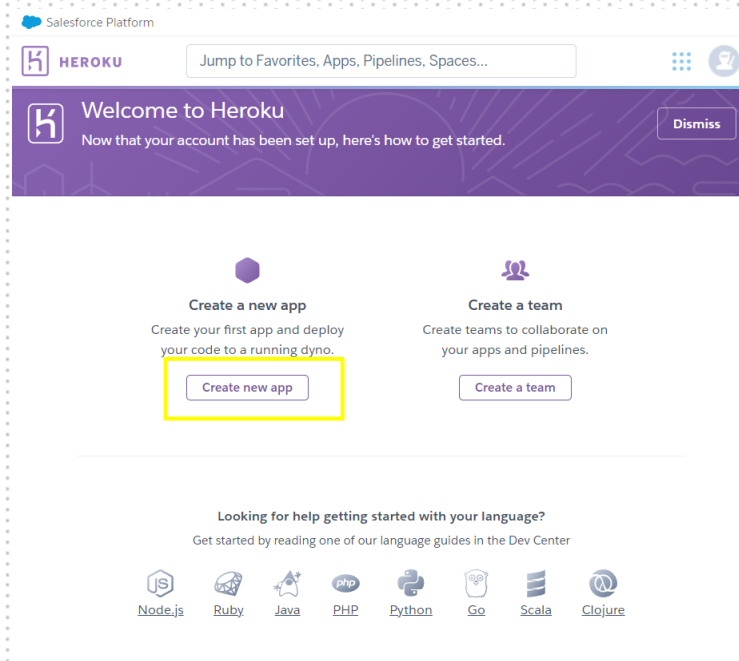
DOCUMENTACIÓN HEROKU

INDICE

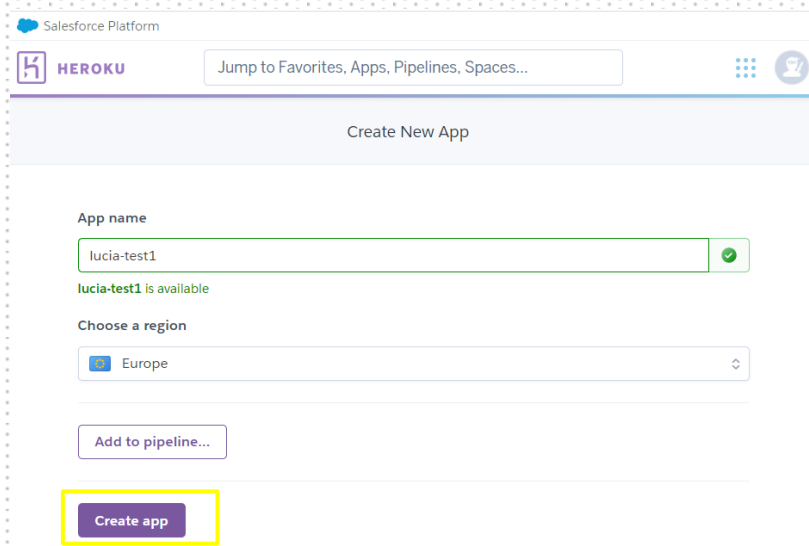
1. PRIMEROS PASOS HEROKU
2. INSTALACIÓN HEROKU CLI
3. RUTAS

1. PRIMEROS PASOS HEROKU

Creamos una cuenta en Heroku, una vez hecha accedemos a esta ventana.
Hacemos click en **create new app**.



Le ponemos nombre a nuestra aplicación y elegimos la región, en mi caso Europa.



A continuación, una vez que tenemos nuestra aplicación creada vamos a desplegar desde GitHub.

Hacemos click en Deploy y seleccionamos GitHub.

Salesforce Platform

HEROKU Jump to Favorites, Apps, Pipelines, Spaces...

lucia-test1 Open app More

Overview Resources **Deploy** Metrics Activity Access Settings

Add this app to a pipeline

Create a new pipeline or choose an existing one and add this app to a stage in it.
Add this app to a stage in a pipeline to enable additional features

Pipelines let you connect multiple apps together and **promote code** between them. [Learn more.](#)

Pipelines connected to GitHub can enable **review apps**, and create apps for new pull requests. [Learn more.](#)

Choose a pipeline

Deployment method

Heroku Git Use Heroku CLI

GitHub Connect to GitHub

Container Registry Use Heroku CLI

Connect to GitHub

Connect this app to GitHub to enable code diffs and deploys.
View your code diffs on GitHub

Connect your app to a GitHub repository to see commit diffs in the activity log.

Deploy changes with GitHub

Connecting to a repository will allow you to deploy a branch to your app.

Automatic deploys from GitHub

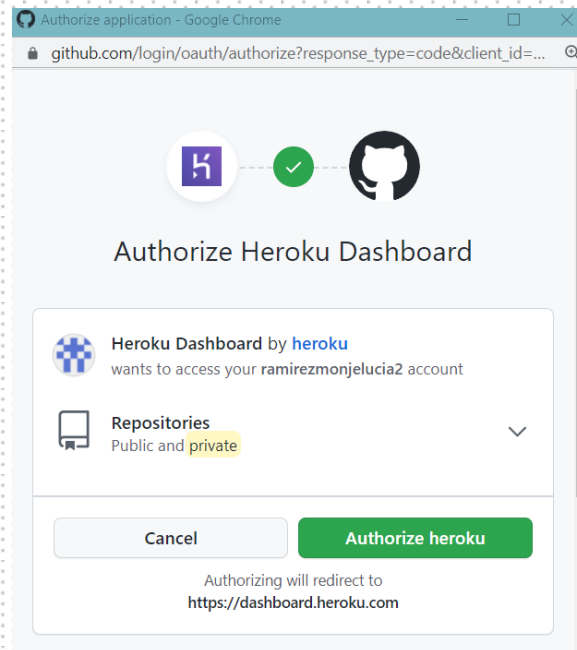
Select a branch to deploy automatically whenever it is pushed to.

Create review apps in pipelines

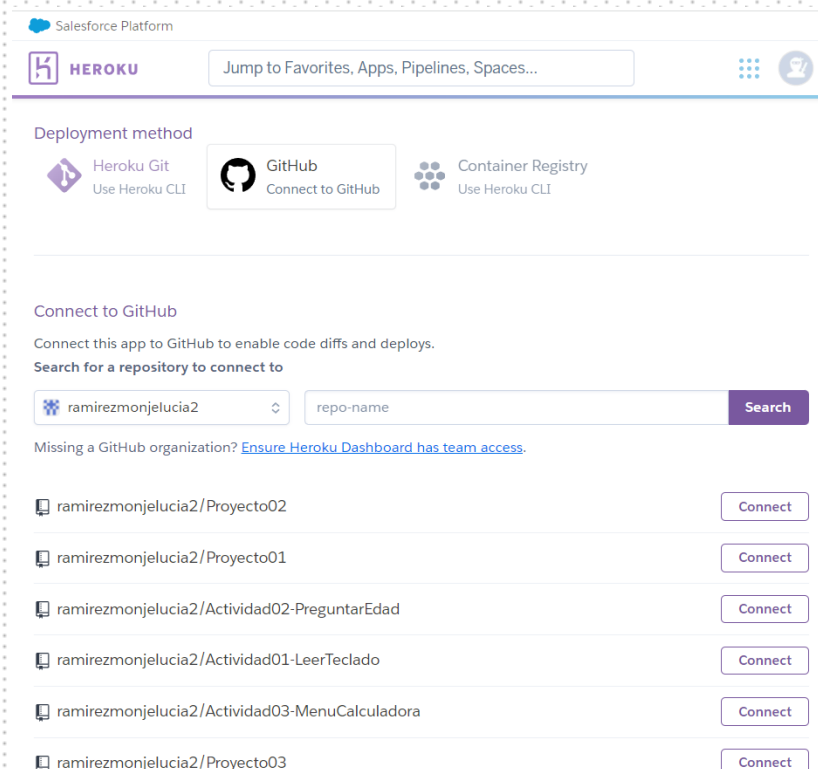
Pipelines connected to GitHub can enable **review apps**, and create apps for new pull requests. [Learn more.](#)

Connect to GitHub

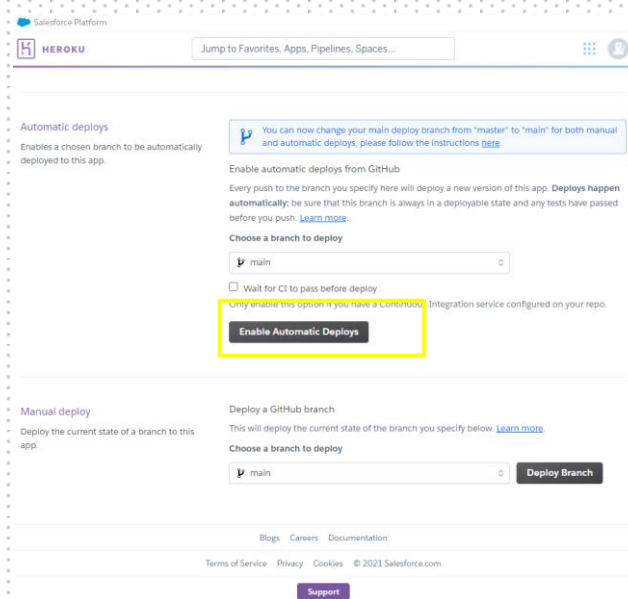
Se nos pedirá que iniciemos sesión, una vez autorizado volvemos al menú principal en Heroku.



Aquí podemos ver que ya nos aparecen todos nuestros repositorios que tenemos subidos a GitHub. Buscaremos el que deseamos desplegar en Heroku.

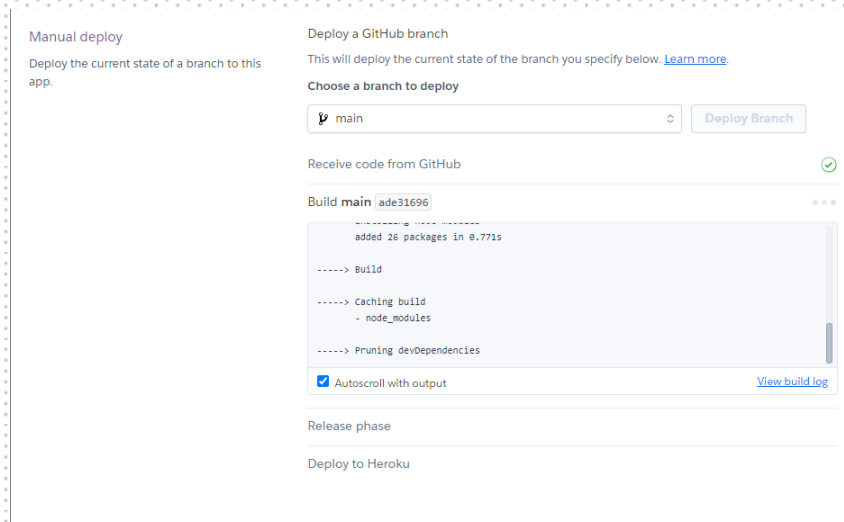


Una vez seleccionado, yo en mi caso habilitaré la opción de despliegue automático, para que si modificamos el documento de GitHub también lo haga en Heroku.

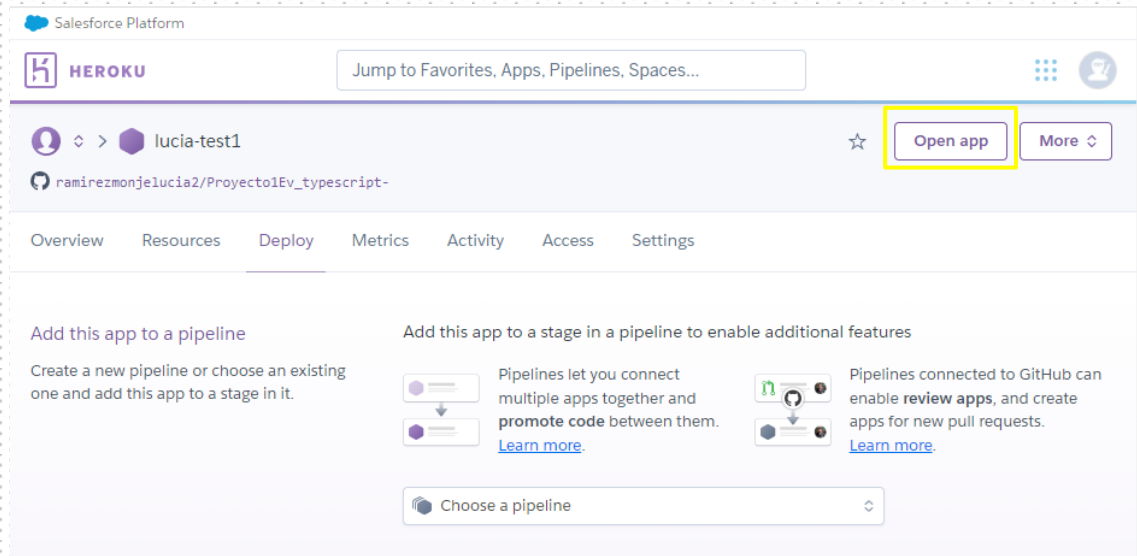


Para ser la primera vez lo desplegaremos manualmente. Seleccionando la opción **Deploy Branch**.

Aquí vemos que se está construyendo la aplicación, cuando esto finalice nos dirigiremos a la parte superior de la ventana.

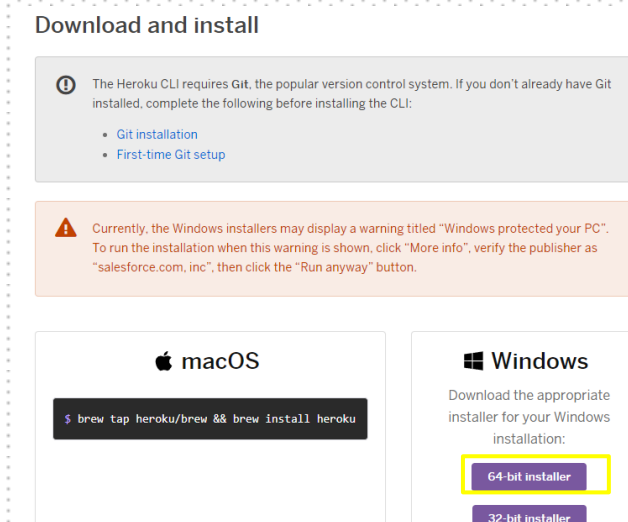


Una vez desplegado para abrir la aplicación haremos clic en **Open app**.

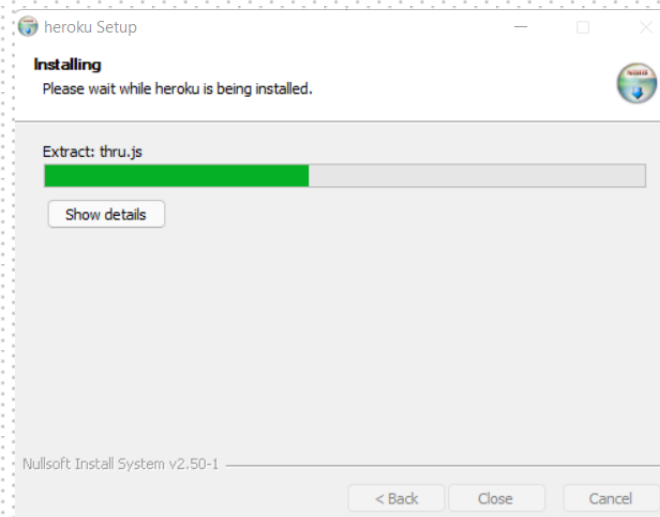


2. INSTALACIÓN HEROKU CLI

En el siguiente [enlace](#) haremos clic en instalar para Windows.



Ejecutamos el archivo de instalación



Una vez instalado nos iremos a un terminal de Visual Studio Code, y escribimos lo siguiente:

`npm install -g heroku`

```
PS C:\Users\lucia\OneDrive\Escritorio\ASIR 2ºA\GBD-Gestión de bases de datos\1EV\API_TEST1> heroku --version
heroku : El término 'heroku' no se reconoce como nombre de un cmdlet, función, archivo de script o programa ejecutable.
Compruebe si escribió correctamente el nombre o, si incluyó una ruta de acceso, compruebe que dicha ruta es correcta e
inténtelo de nuevo.
En línea: 1 Carácter: 1
+ heroku --version
+ ~~~~~
+ CategoryInfo          : ObjectNotFound: (heroku:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException

PS C:\Users\lucia\OneDrive\Escritorio\ASIR 2ºA\GBD-Gestión de bases de datos\1EV\API_TEST1> npm install -g heroku
[.....] \ fetchMetadata: sill resolveWithNewModule heroku@7.59.2 checking installable status
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
gyp ERR! stack at C:\Program Files\nodejs\node_modules\npm\node_modules\node-gyp\lib\util.js:54:7
gyp ERR! stack at C:\Program Files\nodejs\node_modules\npm\node_modules\node-gyp\lib\util.js:33:16
gyp ERR! stack at ChildProcess.exithandler (child_process.js:397:5)
gyp ERR! stack at ChildProcess.emit (events.js:400:28)
gyp ERR! stack at maybeClose (internal/child_process.js:1055:16)
gyp ERR! System Windows_NT 10.0.22000
gyp ERR! command "C:\\Program Files\\nodejs\\node.exe" "C:\\Program Files\\nodejs\\node_modules\\npm\\node_modules\\node-gyp\\
n\\node-gyp.js" "--target=v14.17.6" "rebuild"
gyp ERR! cwd C:\Users\lucia\AppData\Roaming\npm\node_modules\heroku\node_modules\ssh2\lib\protocol\crypto
gyp ERR! node -v v14.17.6
gyp ERR! node-gyp -v v5.1.0
gyp ERR! not ok
Failed to build optional crypto binding
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: cpu-features@0.0.2 (node_modules\heroku\node_modules\cpu-features):
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: cpu-features@0.0.2 install: `node-gyp rebuild`
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: Exit status 1

+ heroku@7.59.2
added 707 packages from 326 contributors in 253.009s
PS C:\Users\lucia\OneDrive\Escritorio\ASIR 2ºA\GBD-Gestión de bases de datos\1EV\API_TEST1> heroku --version
» Warning: Our terms of service have changed: https://dashboard.heroku.com/terms-of-service
heroku/7.59.2 win32-x64 node-v14.17.6
PS C:\Users\lucia\OneDrive\Escritorio\ASIR 2ºA\GBD-Gestión de bases de datos\1EV\API_TEST1> []
```

Como podemos ver después de la instalación al ejecutar el comando

heroku --version nos devuelve los datos de la versión instalada.

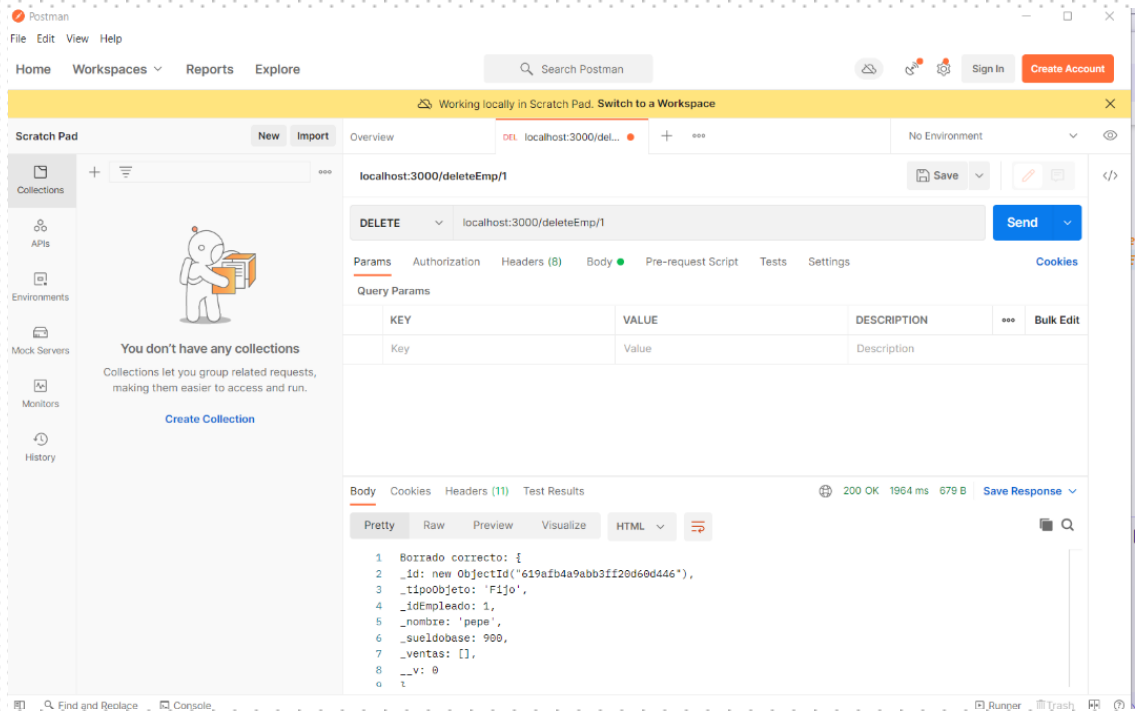
Para finalizar el manual volvemos a la ventana donde tenemos abierta la aplicación, y escribiendo una ruta cualquiera de nuestro proyecto de metodo GET podemos ver que nos devuelve la información de la base de datos de Mongo Atlas.

```
lucia-test1.herokuapp.com/viviendas
Aplicaciones YouTube Drive Gmail Moodle Classroom PASEN Cisco Openwebinars
1 // 20211215090936
2 // https://lucia-test1.herokuapp.com/viviendas
3
4 [
5   {
6     "_ubicacion": {
7       "municipio": "utrera",
8       "ciudad": "sevilla",
9       "codpost": 4123
10    },
11    "_caracteristicas": {
12      "habitaciones": 3,
13      "baños": 2,
14      "ascensor": false,
15      "equipamiento": [
16        "calefaccion",
17        "aire acondicionado",
18        "deposito de agua"
19      ]
20    },
21    "_estado": {
22      "vendido": true,
23      "fecha": "2021-11-22T02:42:38.303Z",
24      "empleado": 1
25    },
26    "_id": "619afb4c9abb3ff20d60d44e",
27    "_tipoObjeto": "Chalet",
```


3. RUTAS

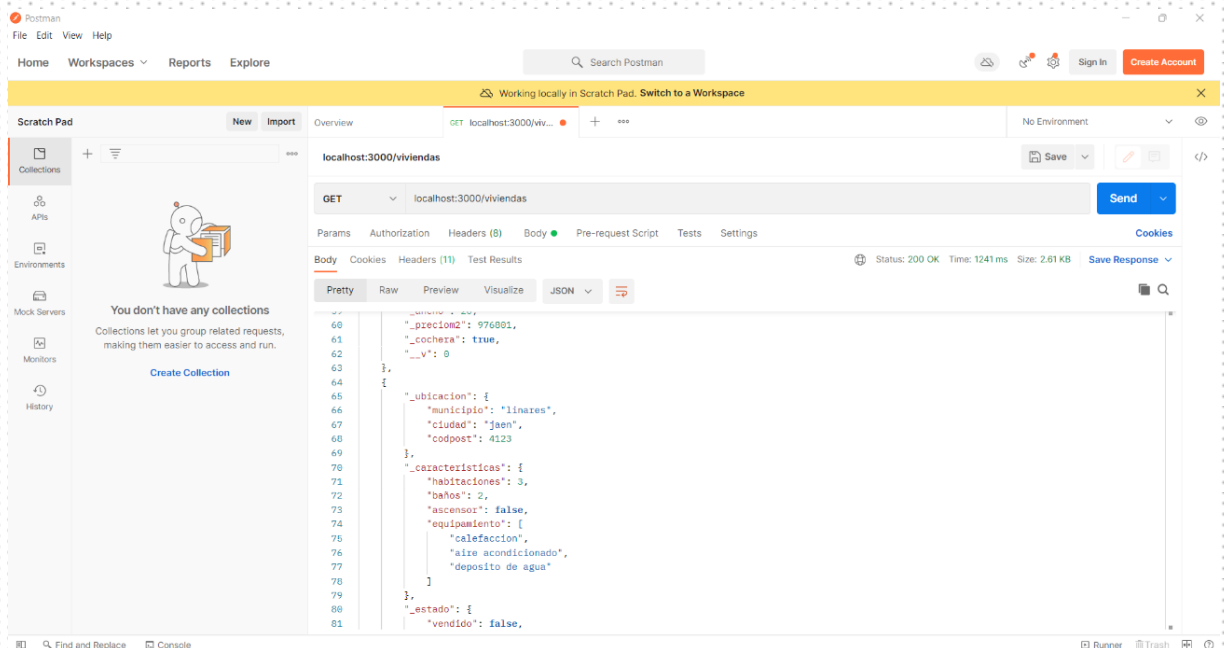
DELETE

```
this._router.delete('/deleteEmp/:idEmp', this.deleteEmpleado)
```



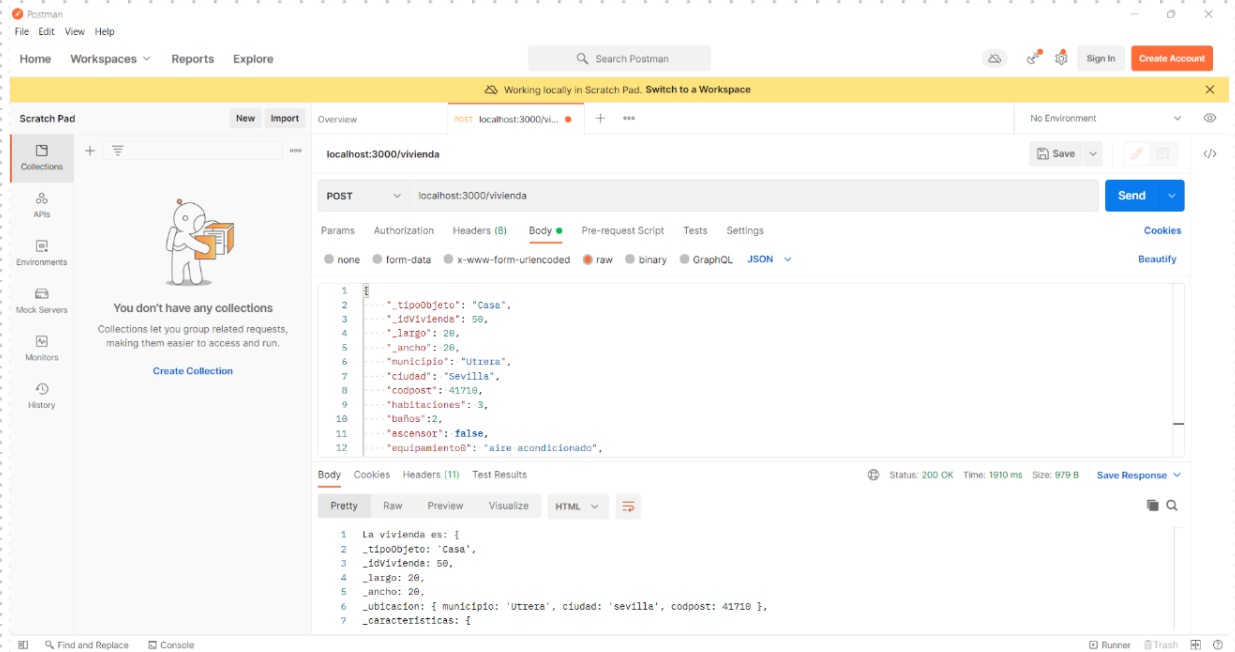
GET

```
this._router.get('/viviendas', this.getViviendas)
this._router.get('/viviendas/:type', this.getTypes)
```



POST

```
this._router.post('/vivienda', this.postVivienda)
```



PUT

```
this._router.put('/empleado/:idEmpleado', this.updateEmpleado)
```

