

# **Despliegue de API Heroku**



Alumno:Alvaro Lucio-Villegas de Cea

**Módulo:ADMINISTRACIÓN de Sistemas Gestores de Bases de  
Datos**

# Índice

---

<b>1.-Subir el Código</b>	<b>2</b>
<b>2.-Heroku</b>	<b>3</b>
<b>3.-Comprobación</b>	<b>3</b>

## 1-.Subir el Código

### 1-.Subir el código a tu repositorio de Github

Para que la aplicación se inicie en Heroku es necesario hacer una serie de pasos

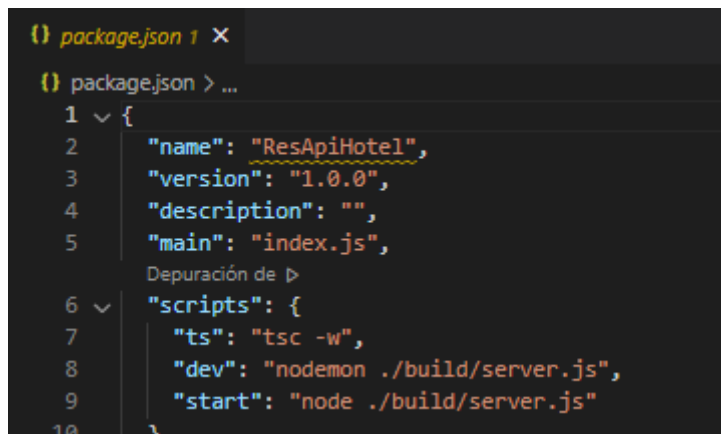
#### 1.1

Instalar los módulos express mongoose morgan cors

```
npm install @types/node @types/cors @types/express @types/morgan nodemon -D
```

#### 1.2

En el archivo package.json en el apartado de “scripts” escribimos los siguiente.



```
() package.json 1 x
() package.json > ...
1  {
2    "name": "ResApiHotel",
3    "version": "1.0.0",
4    "description": "",
5    "main": "index.js",
6    "scripts": {
7      "ts": "tsc -w",
8      "dev": "nodemon ./build/server.js",
9      "start": "node ./build/server.js"
10   }
```

## 2.Una vez terminado el programa lo subimos a nuestro repositorio de Github

```
[21:36:08] Starting compilation in watch mode...

[21:36:14] Found 0 errors. Watching for file changes.

PS C:\Users\Alvc00\Desktop\Proyecto Navidad> git add .
PS C:\Users\Alvc00\Desktop\Proyecto Navidad> git commit -m "Tercer commit"
[main 022caeb] Tercer commit
2 files changed, 654 insertions(+), 119 deletions(-)
PS C:\Users\Alvc00\Desktop\Proyecto Navidad> git push -u master2 main
Enumerating objects: 15, done.
Counting objects: 100% (15/15), done.
Delta compression using up to 4 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (8/8), 3.09 KiB | 1.03 MiB/s, done.
Total 8 (delta 5), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (5/5), completed with 5 local objects.
To https://github.com/lucioceaalvaro/Proyecto-API-Resp-Hotel.git
0dff799..022caeb main -> main
Branch 'main' set up to track remote branch 'main' from 'master2'.
PS C:\Users\Alvc00\Desktop\Proyecto Navidad> 
```

## 2.-Heroku

3. Cuando el repositorio está creado nos dirigimos a la página de [Heroku](#) y crearemos una app que enlazaremos con el repositorio de github. Y lanzamos la aplicación de forma automática.

The screenshot shows the Heroku dashboard for a user named 'Personal'. At the top, there's a navigation bar with the Heroku logo and a search bar. Below it, a 'Welcome to Heroku' banner is visible. The main content area shows the app 'proyecto-segunda-evaluacion' which is connected to the GitHub repository 'lucioceaalvaro/Proyecto-API-Resp-Hotel'. The app is currently in the 'main' branch. Below the app name, there are tabs for 'Overview', 'Resources', 'Deploy', 'Metrics', 'Activity', 'Access', and 'Settings'. The 'Deploy' tab is selected, showing options to add the app to a pipeline or connect it to GitHub. The 'Deployment method' section shows 'Heroku Git' as the selected method, with a 'Connected' status. Below this, it shows the app is connected to the GitHub repository 'lucioceaalvaro/Proyecto-API-Resp-Hotel' by 'lucioceaalvaro'. There are links to 'Releases in the activity feed' and 'Automatically deploys from main'.

## 3.-Comprobación

Comprobamos que está en funcionamiento

The screenshot shows a web browser window with the address bar displaying 'https://proyecto-segunda-evaluacion.herokuapp.com'. Below the address bar, the text 'Plataforma Hotel' is displayed in a large, bold, serif font. Below this, there are four links: 'Ver Clientes', 'Ver Empleados', 'Ver Habitaciones', and 'Ver Reservas', all in a smaller, blue, serif font.