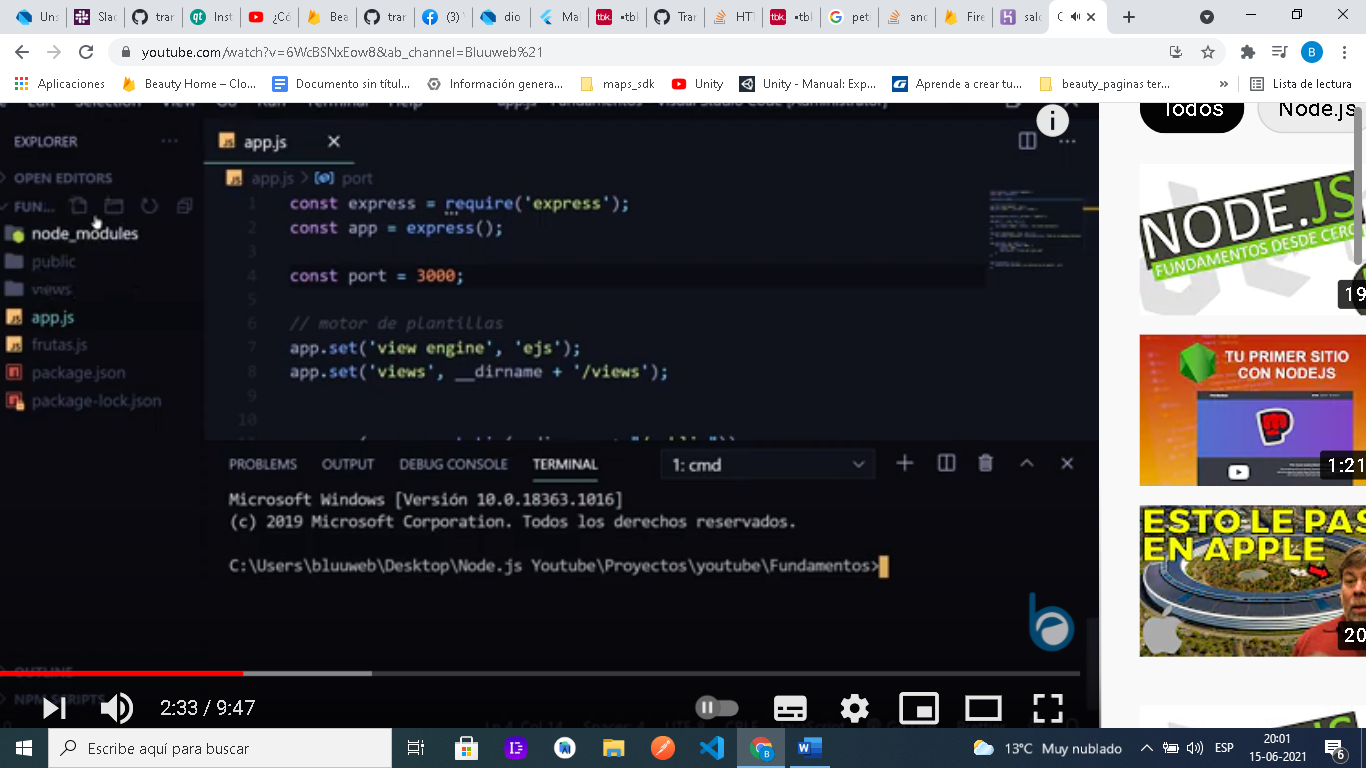
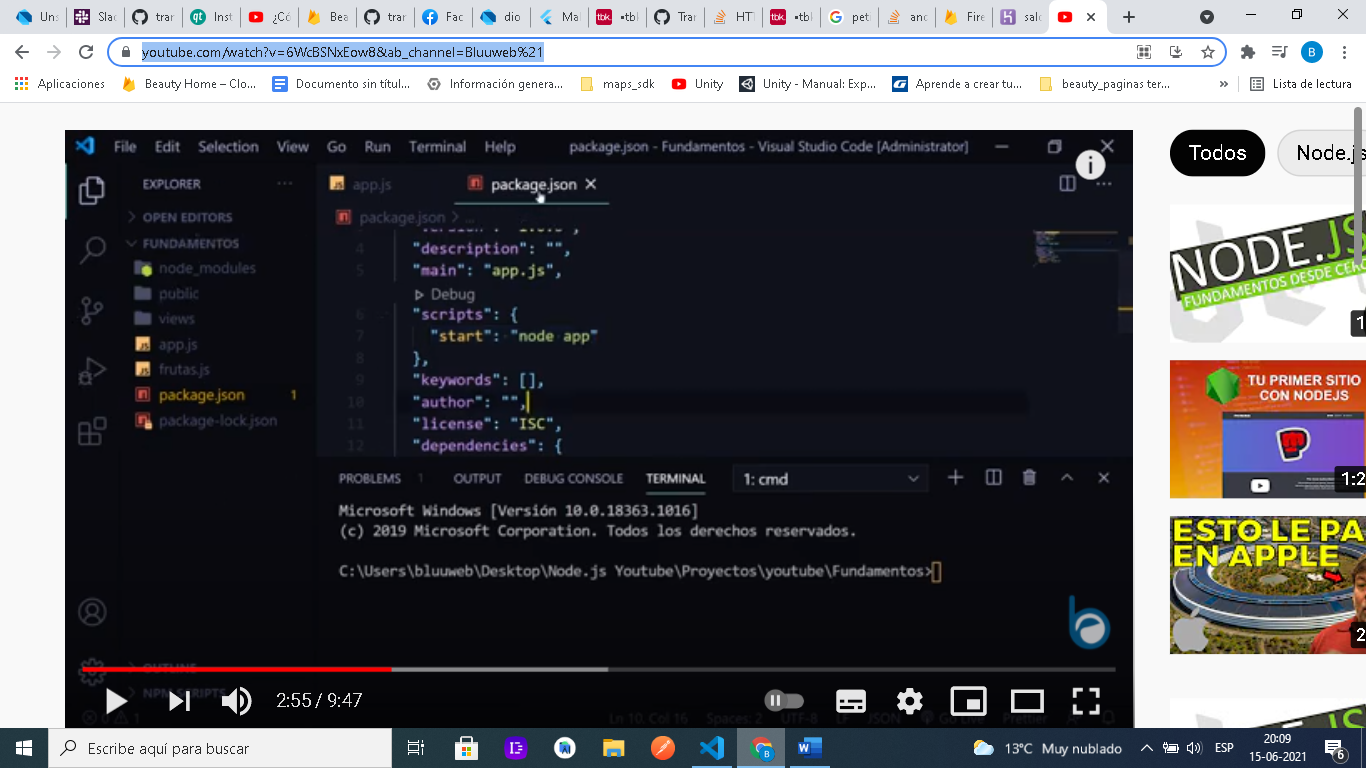
Nodejs a servidor web heroku

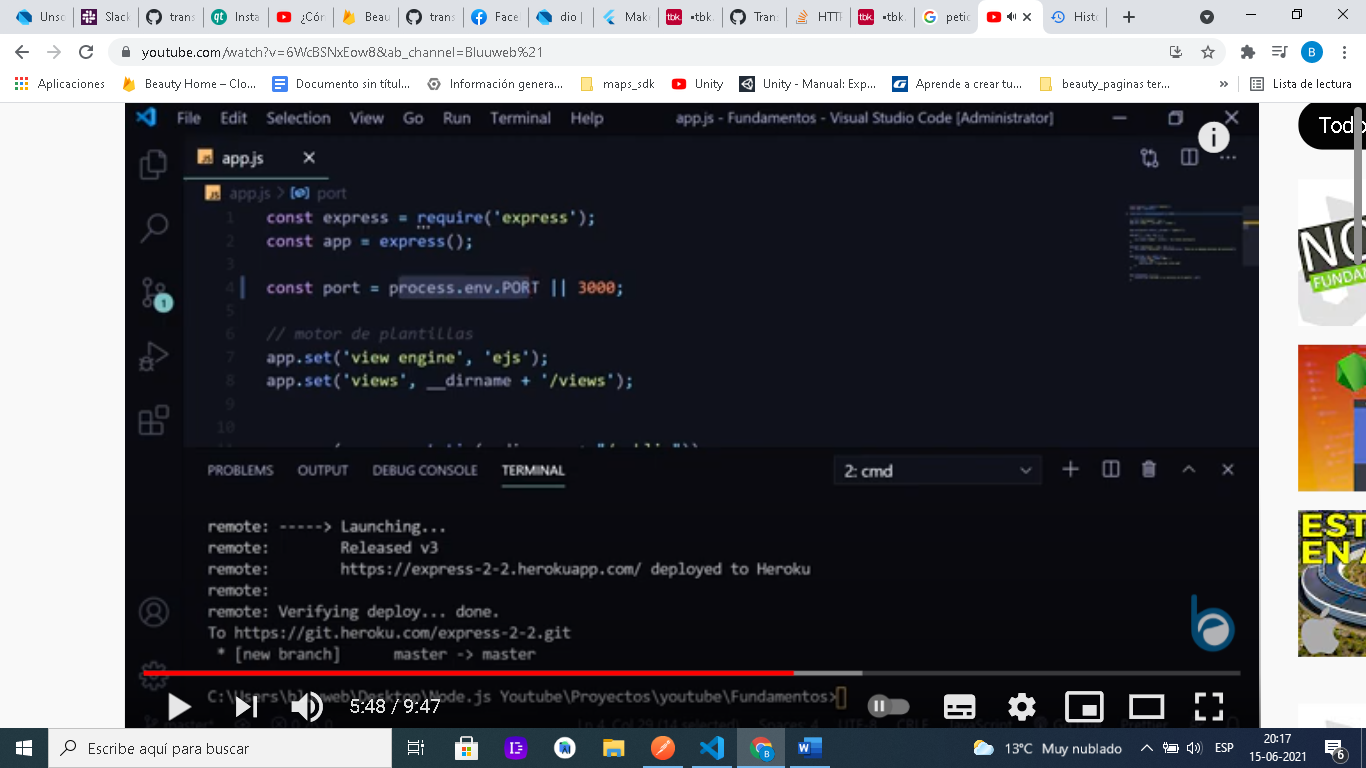
Heroku es el nombre de un hosting gratuito por un gran %

Primero nos dice no agregar el module



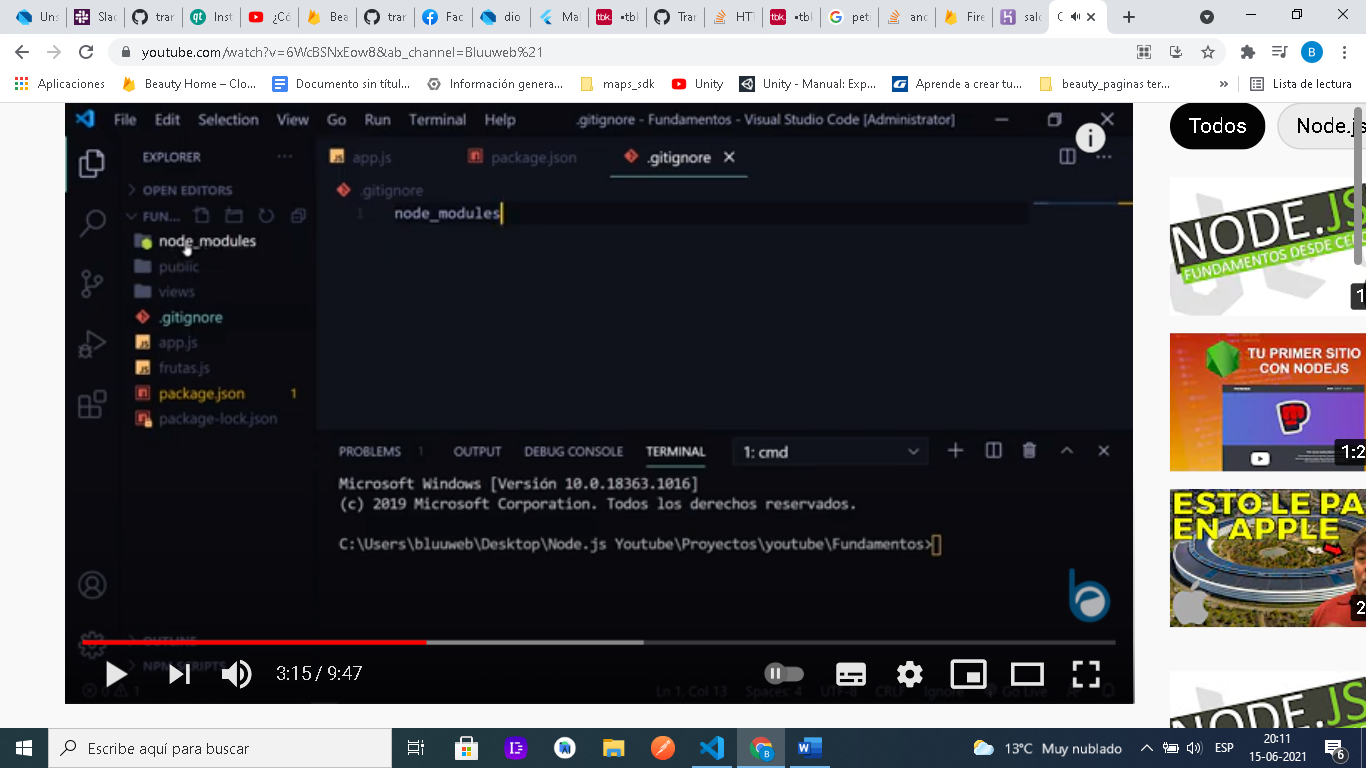
<https://www.youtube.com/watch?v=6WcBSNxEow8&ab_channel=Bluuweb%21>

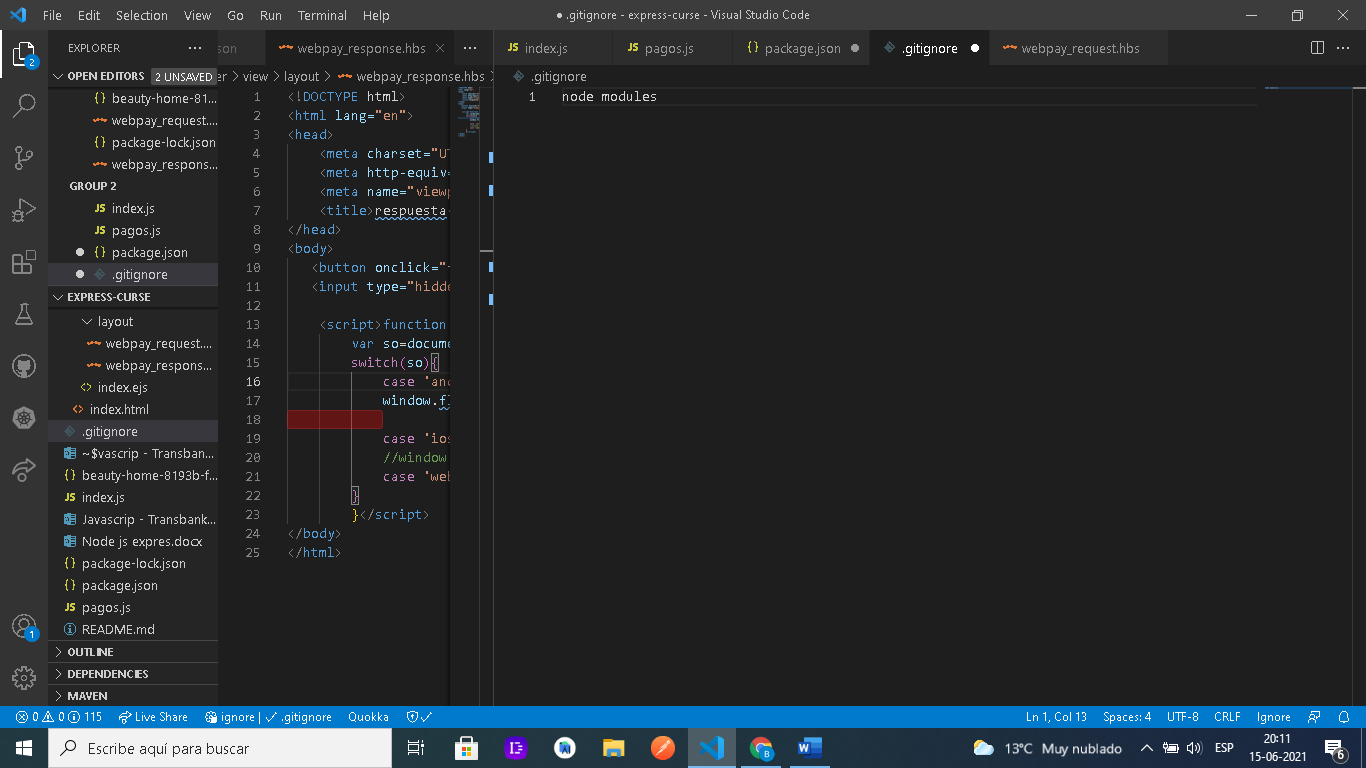




Ahora evitamos que suba la carpeta creando un gitignore

Para hacer la versión del proyecto





Create a new Git repository

Initialize a git repository in a new or existing directory

$ cd my-project/

$ git init

$ heroku git:remote -a salonhousev1

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

#### Existing Git repository

For existing repositories, simply add the heroku remote

$ heroku git:remote -a salonhousev1

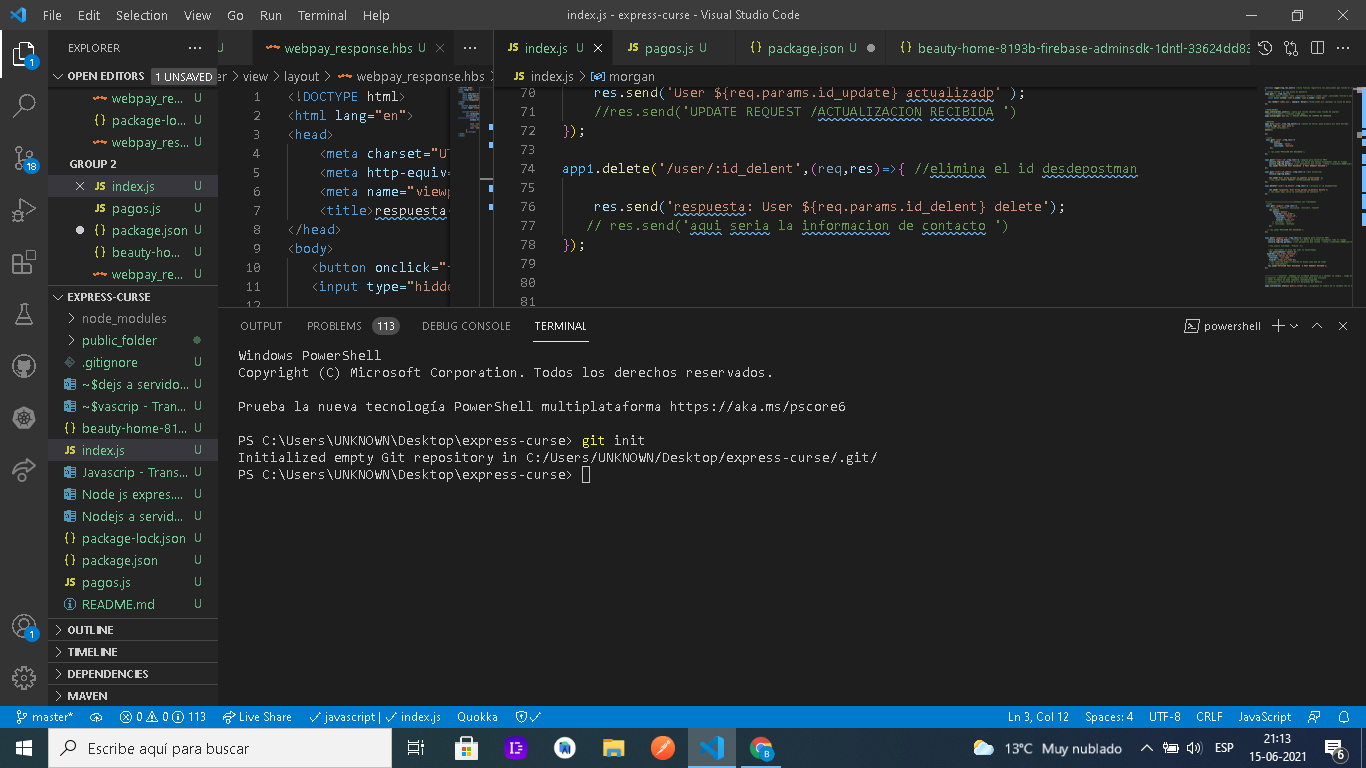
# empezamos

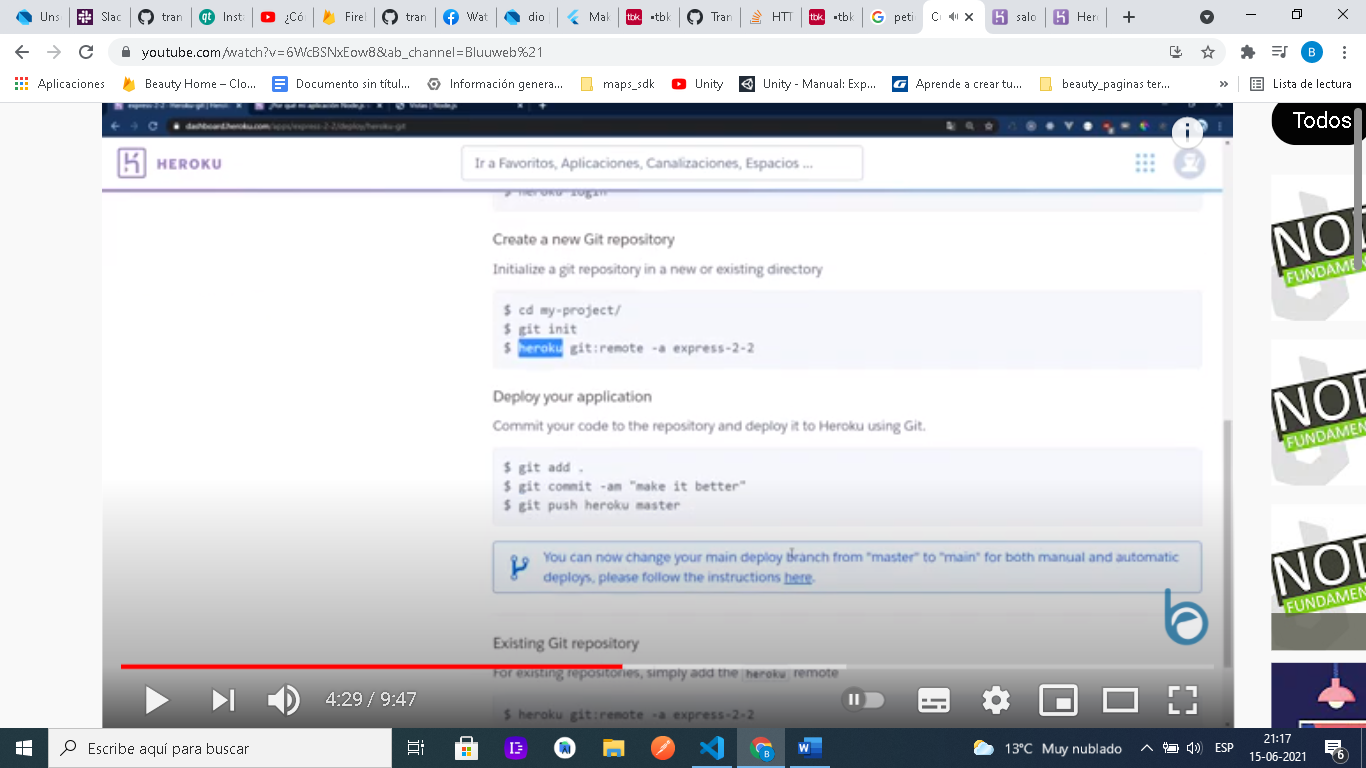
Una vez creato el git que es llamar casi todos los anteriores scrips

Configurar variable de entorno

Se refiere a cambiar el puerto ej 3000 por un código respectivoa la dirección url que nos dará el dominio de nestro servidor web

$ git init

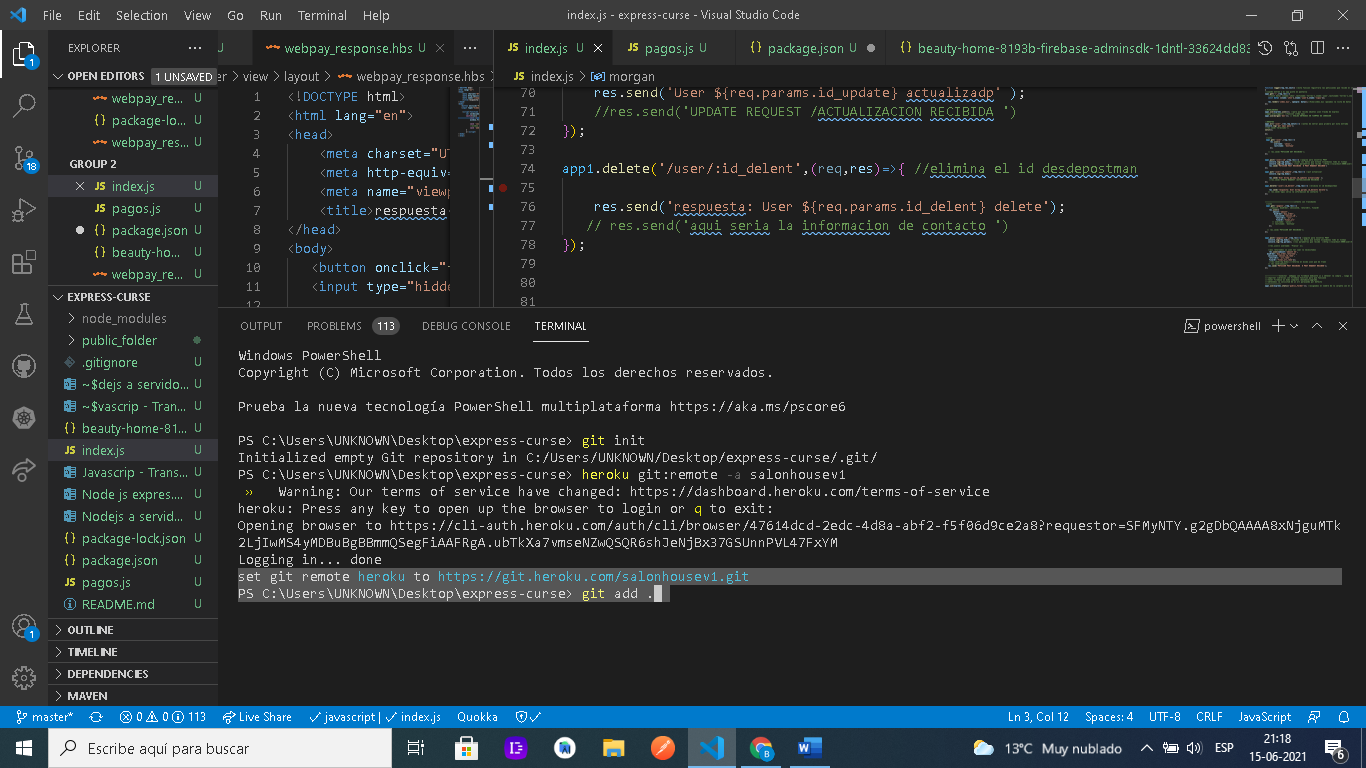


Inicializamso heroku 

set git remote heroku to https://git.heroku.com/salonhousev1.git

PS C:\Users\UNKNOWN\Desktop\express-curse> git add .

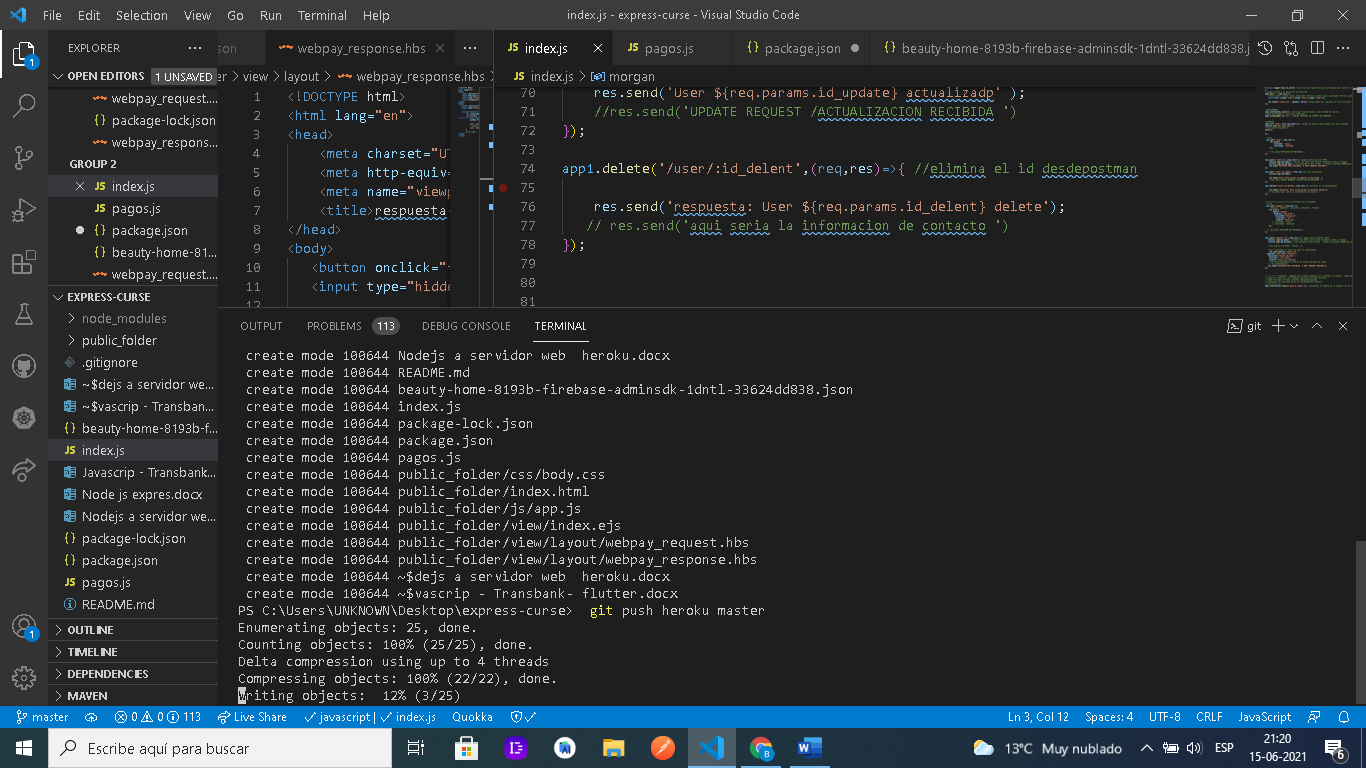
Ahora vemso lo que vamos a subir antes de enviarlo con el comando git add .



$ git add .

$ git commit -am "make it better"

$ git push heroku master



# Nos da un error para obtener el url de nuestra app web

Este nos da un mensaje con el url personal

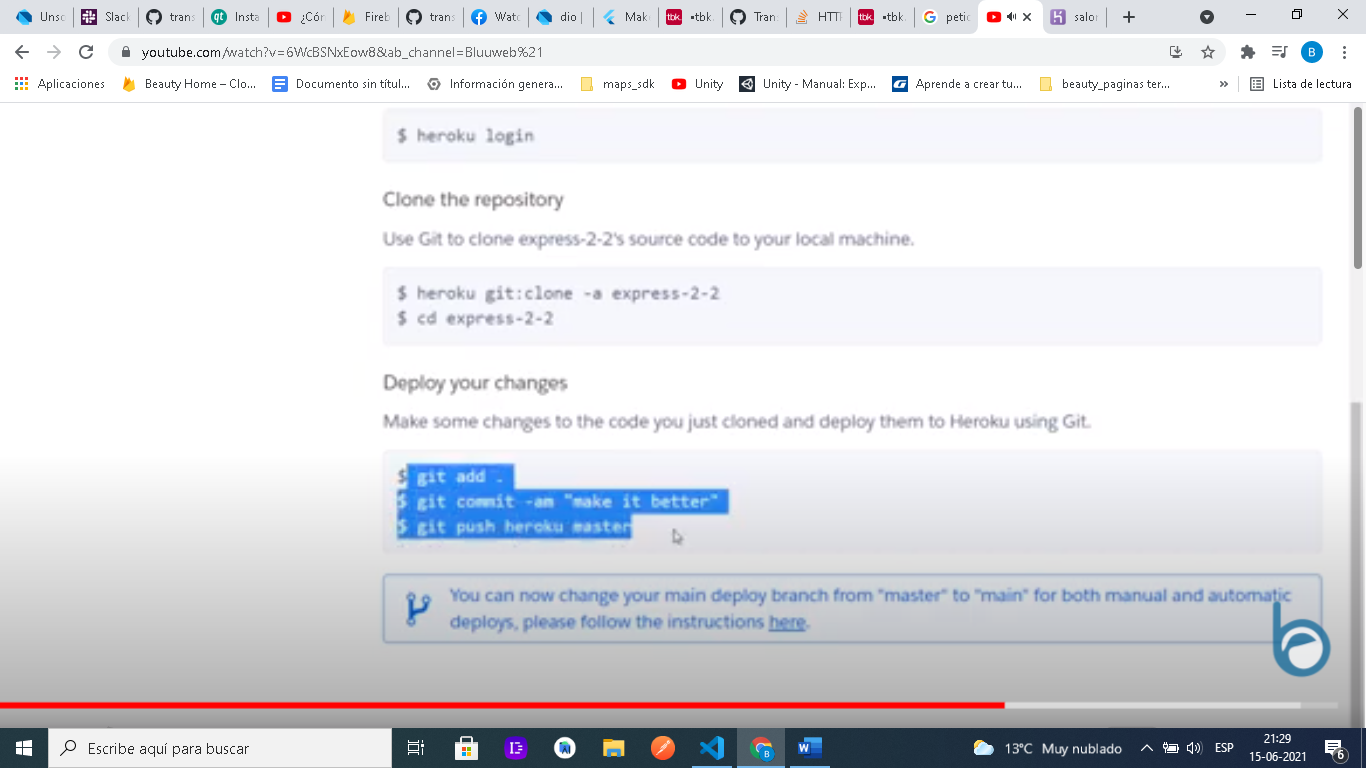
Aquí debemos reemplazar una palabra para el puerto y obtener nuestro autentico url

const port=process.env.port|| 4000;

app1.set('port',process.env.port|| 4000);

aquí hacemos esto para que que heroku nos designe un puerto

con esto indicamos que ya debe funcionar el puerto y cada vez que queramos actualizar el puerto simplemente usamos estos comandos



<https://dashboard.heroku.com/apps/salonhousev1/deploy/heroku-git>

$ git init

heroku git:remote -a salonhousev1

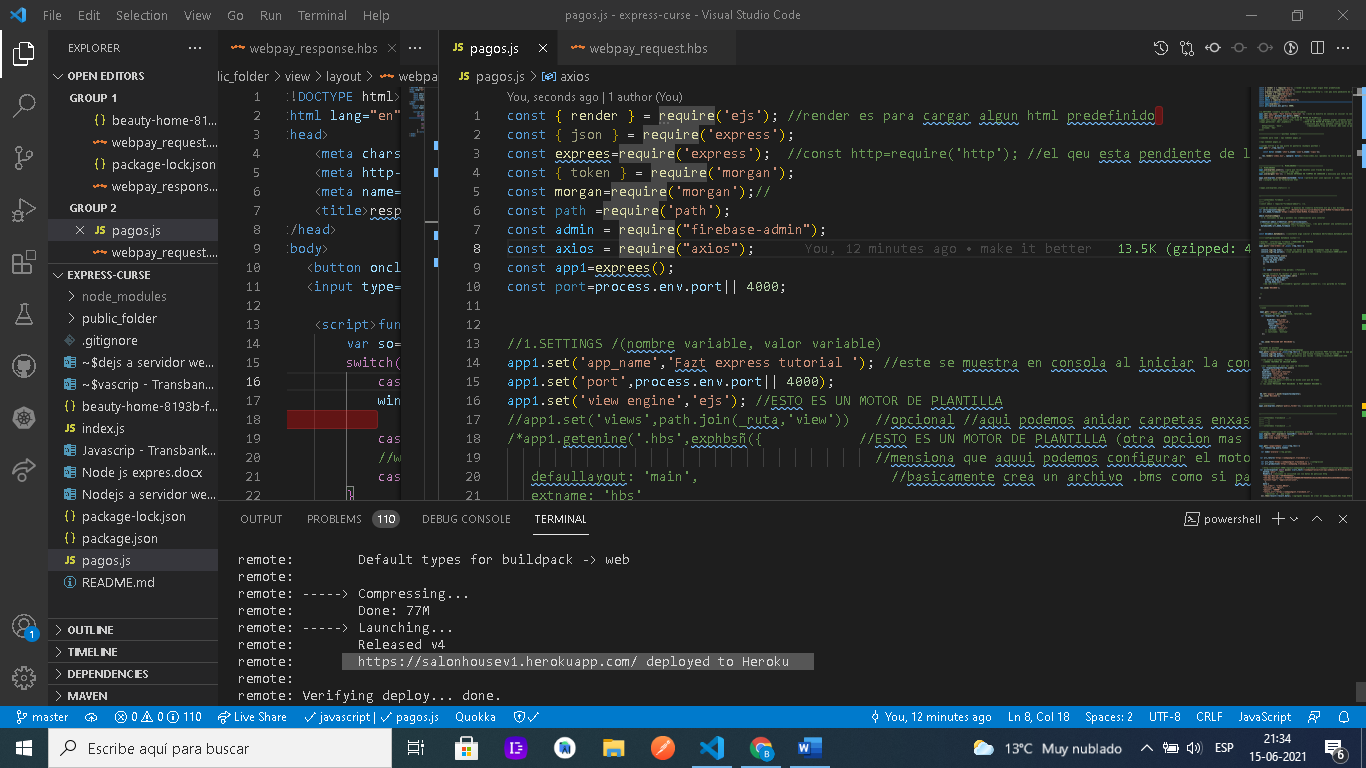
git remote add origin <https://git.heroku.com/salonhousev1.git> //ya esixte

$ git add . //guardar los cambios

$ git commit -am "cambiamos a 3000" //agregar un mensaje

$ git push heroku master //enviamos al server y nos da el url de neustro servidor

git pull <https://git.heroku.com/salonhousev1.git>



https://salonhousev1.herokuapp.com/ deployed to Heroku

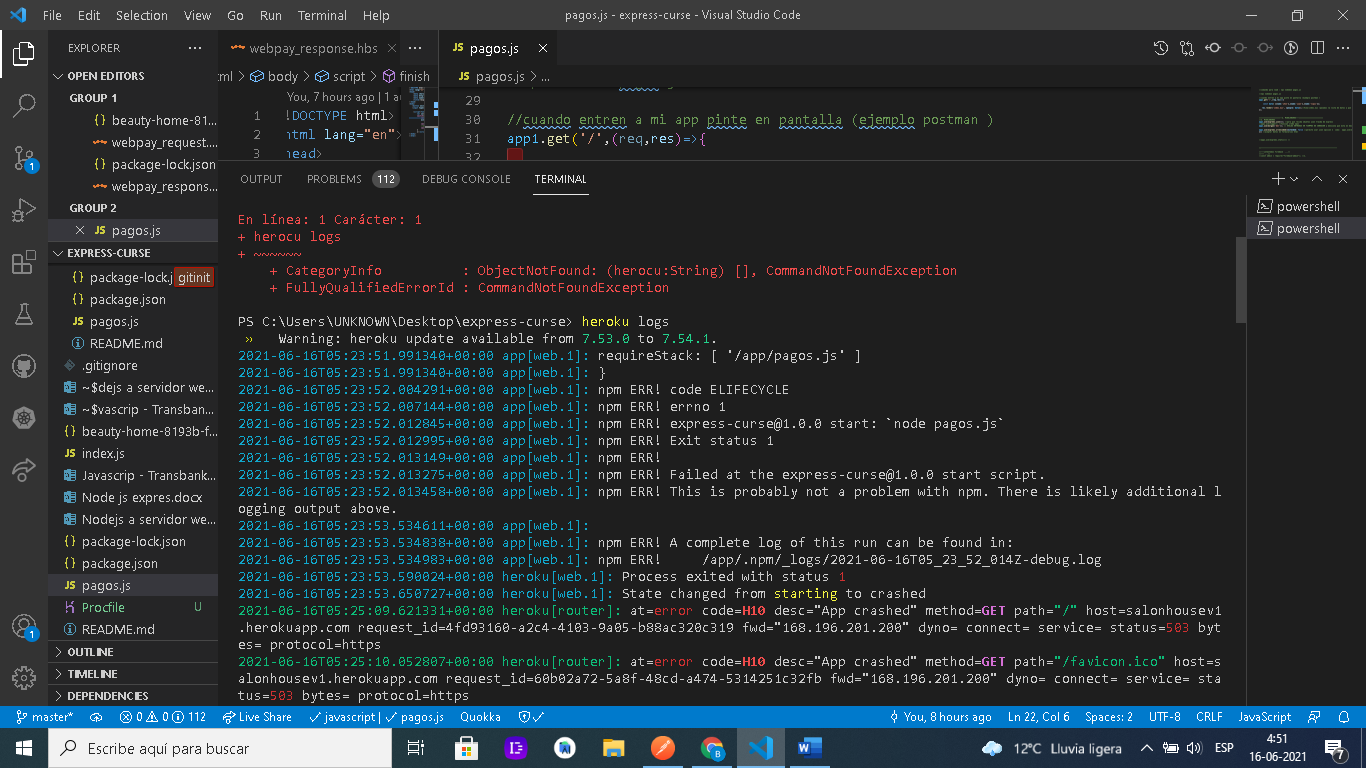
obtenemos el url

## ver herrroes de consola con heroku

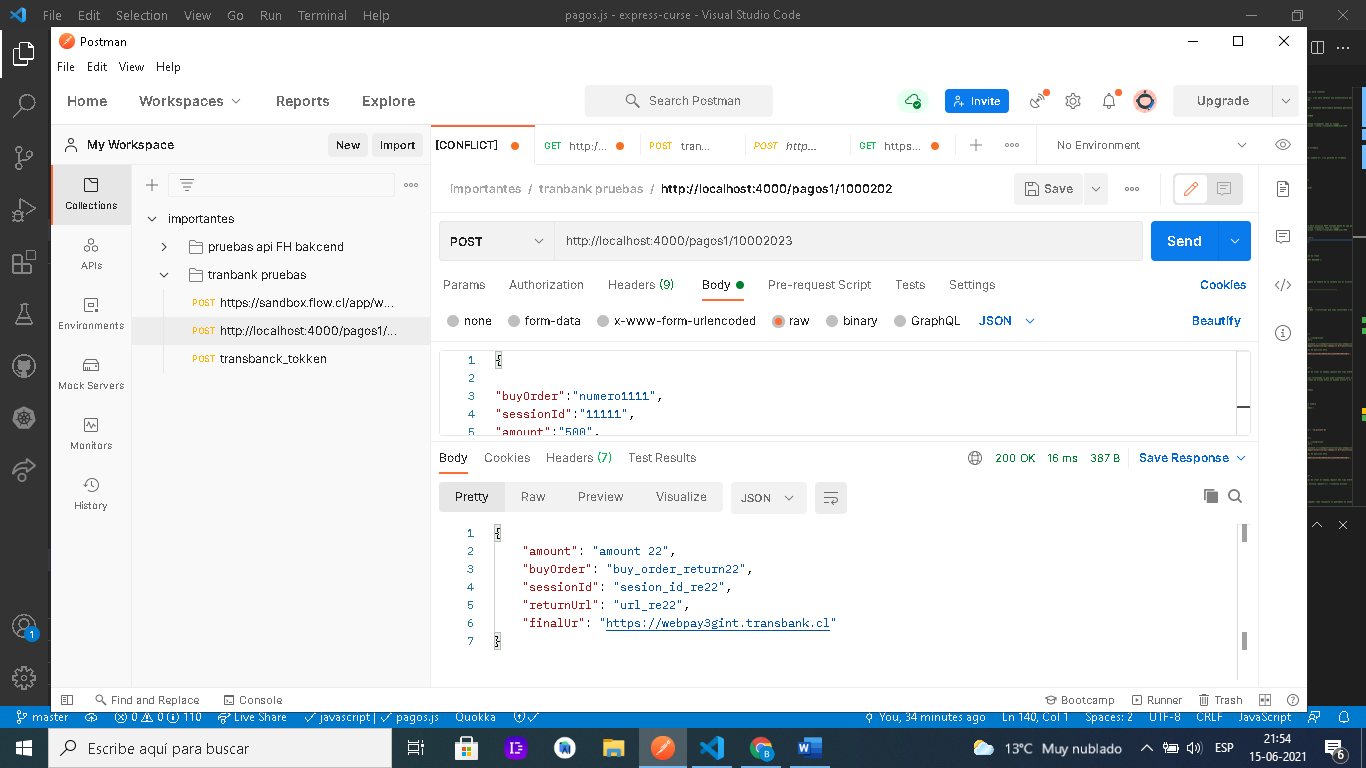
usa simplemente heroku logs

<https://www.youtube.com/watch?v=mkn5n8Ws-HE&ab_channel=AprendeJavaScriptconCarlosAzaustre>

aquí explican que debemos eliminar de las dependencias el nodemon para ir a produccion

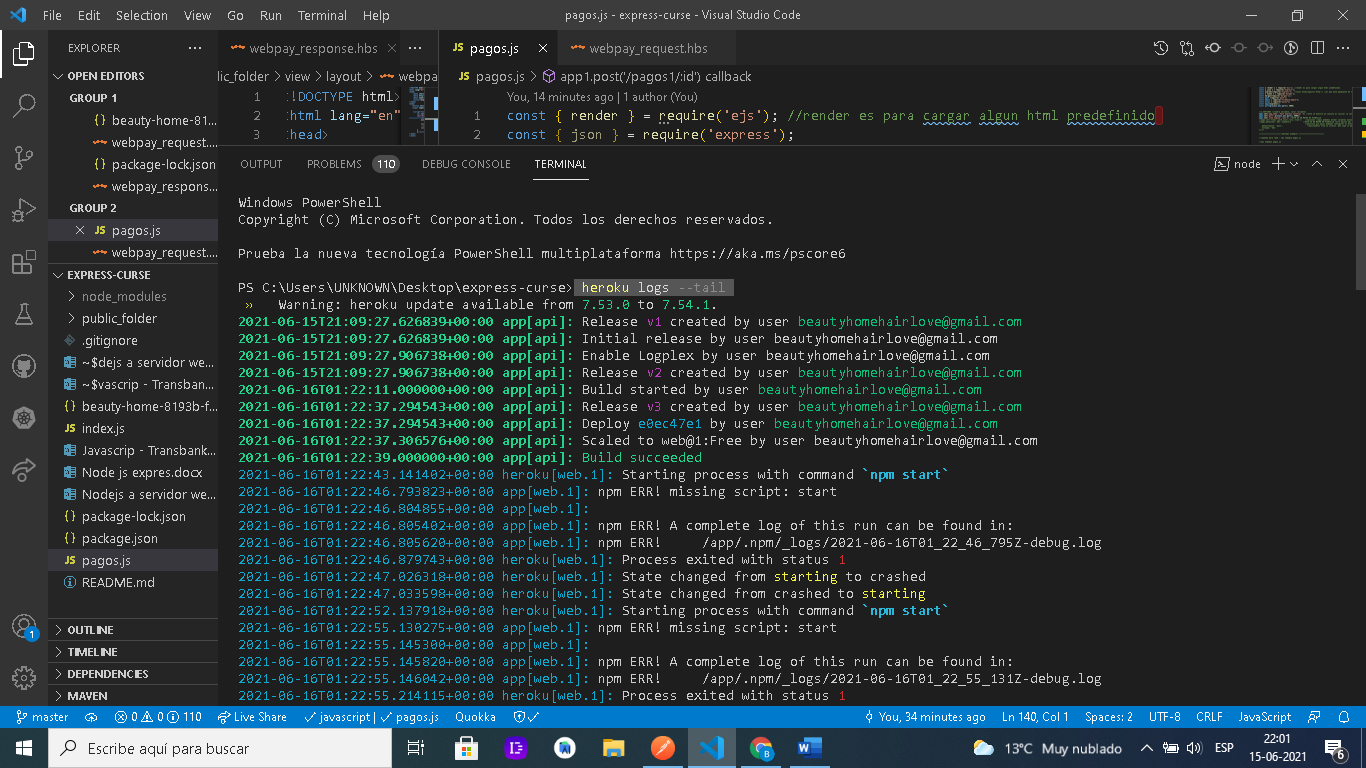


/// en el siguiente ejemplo compruebo como usar postman



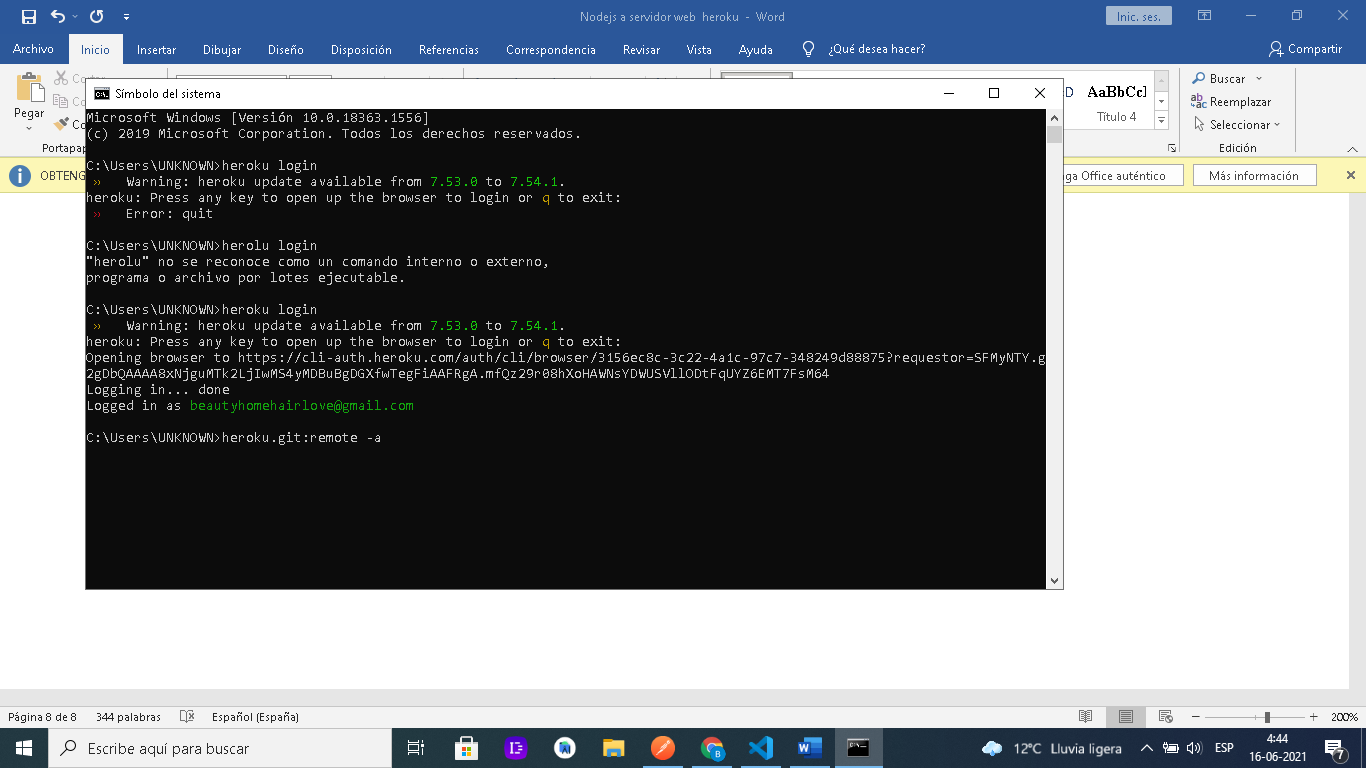
heroku logs –tail

nos da toda la información del servidro



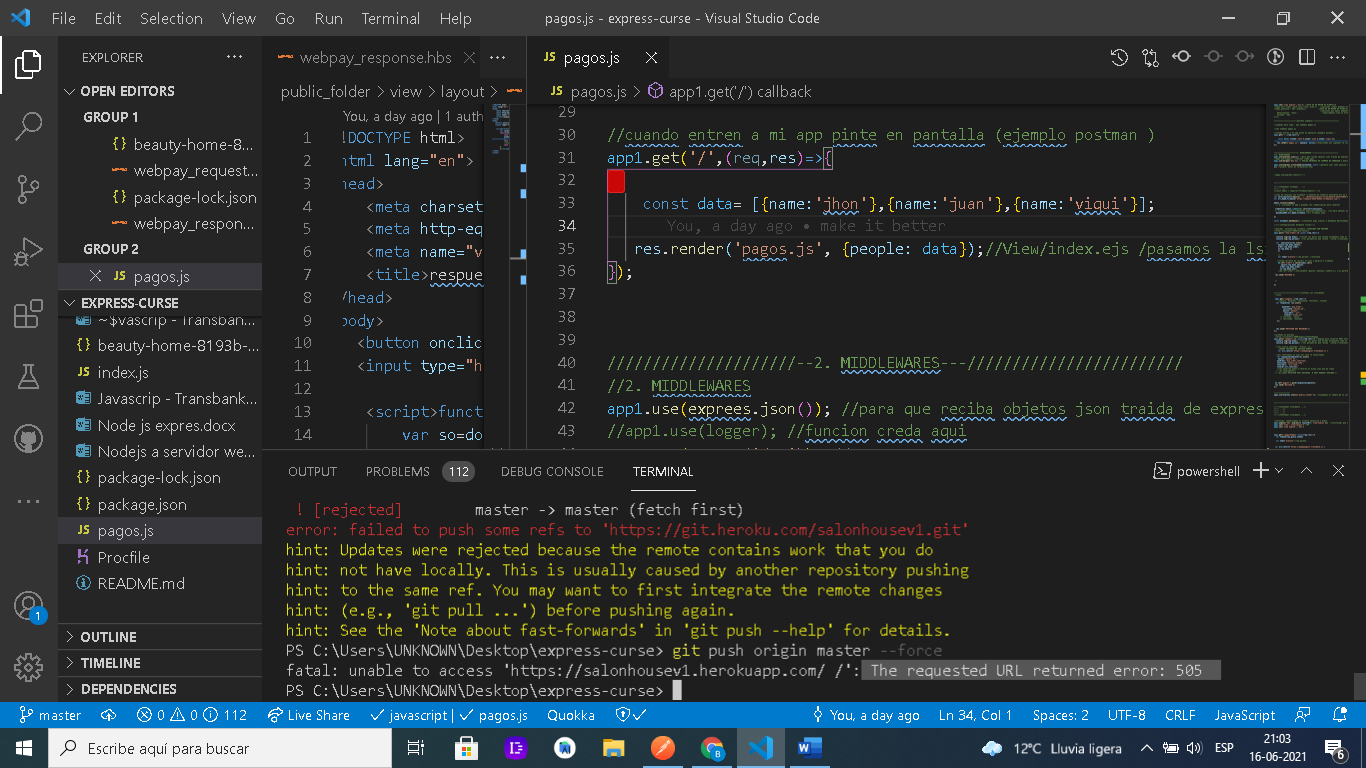
## heroku run rails console

>heroku login



Intento 3

The requested URL returned error: 505



GIA DEFINITIVA GITHUB PARA SUBIR GIT A HEROKU O MI DIRECCION DE GITHUB

$ git init

heroku git:remote -a salonhousev1

git remote add origin <https://git.heroku.com/salonhousev1.git> //ya esixte

$ git add . //guardar los cambios

$ git commit -am "cambiamos a 3000" //agregar un mensaje

$ git push heroku master //enviamos al server y nos da el url de neustro servidor

git pull <https://git.heroku.com/salonhousev1.git>

//este me funciono una vez para heroku directamente

$ git init

heroku git:remote -a salonhousev1

//git remote add origin <https://git.heroku.com/salonhousev1.git>

git add .

git commit -am "cambiamos a 3000"

git pull <https://git.heroku.com/salonhousev1.git> //esto actualiza el repositorio local desde otro..

git push heroku master

git push heroku master – forcé //lo sube pero me da error aun en la consola

## versión kevin nodejs:

git init

git add -A

git commit -am "cambiamos a 3000"

///git pull <https://git.heroku.com/salonhousev1.git> //pull obtiene los datos de la direccion

git remote add origin https://github.com/Jesus-tasker/transbank1-nodejs-express.git

//

git init

git remote add master https://github.com/Jesus-tasker/transbank1-nodejs-express.git

git remote add origin https://github.com/Jesus-tasker/transbank1-nodejs-express.git

git pull origin master

git push origin master  
  
Explicacion git desde 0 a github:

-git init : Lo que hace cada comando es, git init, inicializa el git

-git add . crea un git

-git commit -am "cambiamos a 3000"

- git remote add origin [https://github.com/Jesus-tasker/transbank1-nodejs-express.git]

//hasta a qui va bien

//-git push -u git push origin master

-git push <https://github.com/Jesus-tasker/transbank1-nodejs-express.git>

-Git remote: origin master //(origin master)agregas un remote que lo une con github en la nube que se llama "origin", osea el remote se llama origin …..si dice que ya existe 20/06

Publicar en github el js y node js

- **git** clone <https://github.com/Jesus-tasker/transbank1-nodejs-express.git>

-git init // : Lo que hace cada comando es, git init, inicializa el git

-git add //. crea un git

-git commit -am "cambiamos a 3000"

- git remote add origin https://github.com/Jesus-tasker/transbank1-nodejs-express.git

-git push -u origin master

-git pull <https://github.com/Jesus-tasker/transbank1-nodejs-express.git>

git push <https://github.com/Jesus-tasker/transbank1-nodejs-express.git>

//verificarciones

<https://www.youtube.com/watch?v=H03E8_HImAU&ab_channel=FacultadAutodidacta>

git init

git add -A : agrega todo lo que hay ene l repositorio ya que es la primera ves en subir el git

git status

git commit -m "cambiamos a 3000" //nota es -am o -m (rescomendado )

git remote add origin <https://github.com/Jesus-tasker/transbank1-nodejs-express.git>

git pull <https://github.com/Jesus-tasker/transbank1-nodejs-express.git>

git push origin master –force

// error: hint: to the same ref. You may want to first integrate the remote

git remote -v //en caso de error ver las configuraciones remotas

git remote rm origin //asi lo haces de nuevo

//publicado

como haerlo siempre cuando ya se subio a github

git add -A //parece ser general

git status

git commit -m "versión 2"

git push origin master

# Desde 0 cree un nuevo repositorio en gitdhub y publico

git init

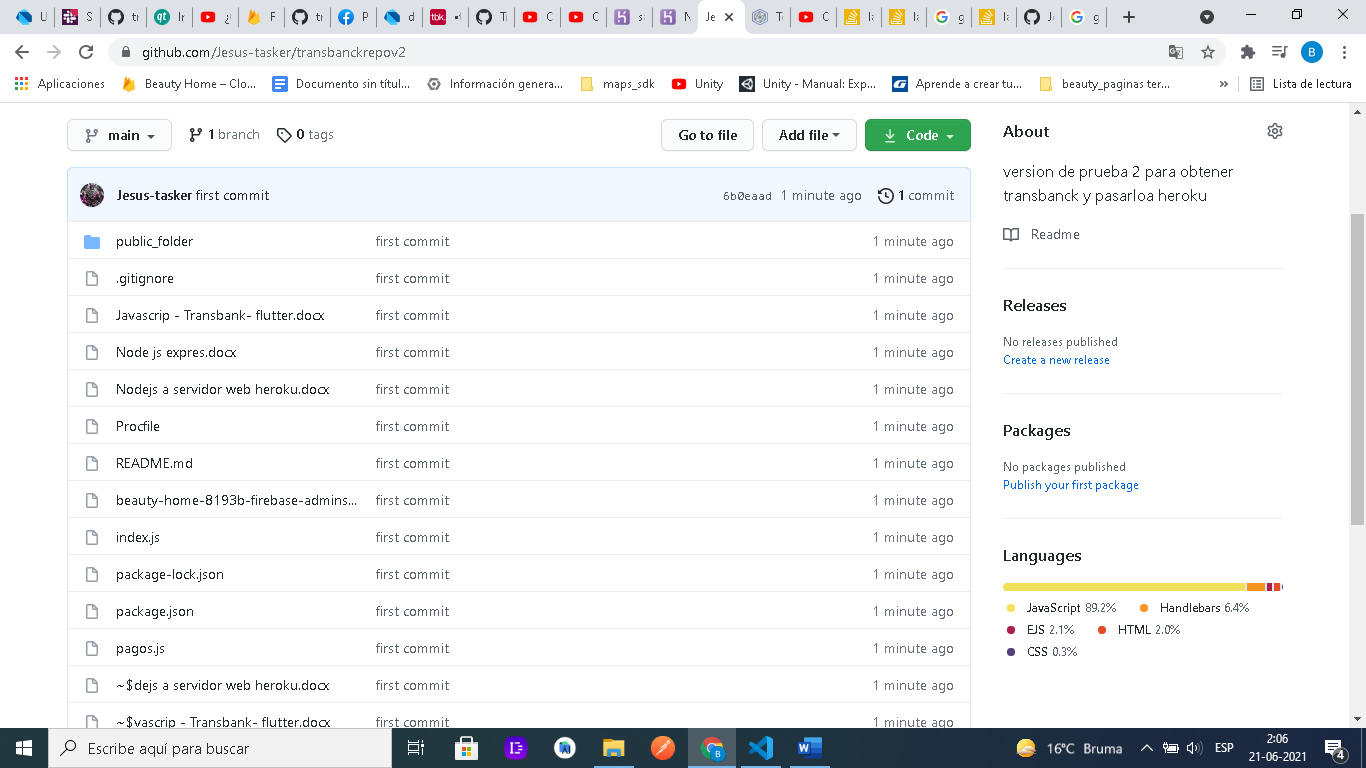
git add README.md

git commit -m "first commit"

git branch -M main

git remote add origin https://github.com/Jesus-tasker/transbanckrepov2.git

git push -u origin main



# Actualizar el git

git remote add origin <https://github.com/Jesus-tasker/transbanckrepov2.git>

git add -A

git commit -m "versión "

// git branch -M main //no lo use pero qeuizas es necesario

git push -u origin main

y asi actualizamos el github

aprender a usar variables de entorno (enviroment variable )

guardar datos importantes en el servidor

<https://medium.com/the-node-js-collection/making-your-node-js-work-everywhere-with-environment-variables-2da8cdf6e786>

http://localhost:5000/pagarv2?token=ajajajajak

1 la APK pide a api solicitar la URL

2 api solicitar a transback el link de botón de pago

3 api retorna la URL a la APK

4 APK usuario paga

5 página de pago RECIBIDO

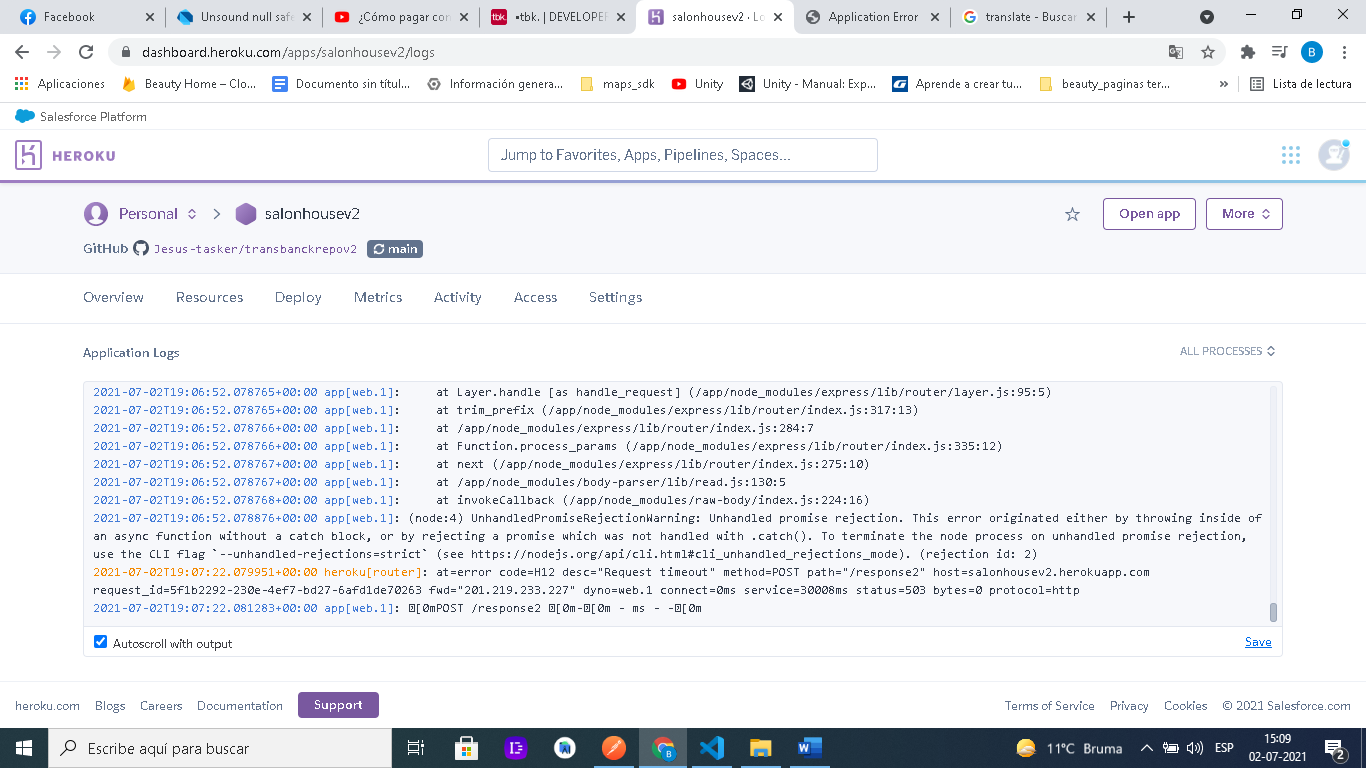
Luego comienza en paralelo

1 transback válida el pago y envía estado a api

2 api registra los datos en la DB

3 se envía por socket a la APK

4 APK recibe alerta y muestra estado final del pago



1. git stash (almacena el trabajo no comprometido temporalmente)
2. git pull (haga que su local y remoto estén sincronizados)
3. git stash pop (recupera los cambios no confirmados)
4. empujar git