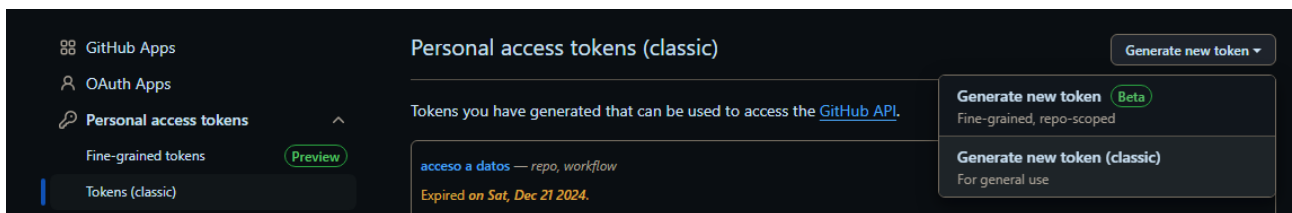


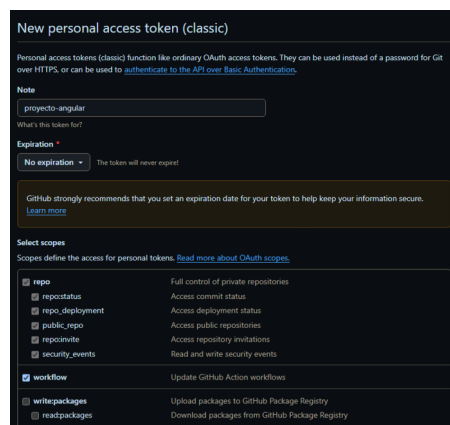
## Subida a GitHub Pages y otros

### Credenciales de acceso a GitHub

El primer paso será configurar el acceso a GitHub desde el terminal, para ello necesitaremos unos credenciales de acceso. Para ello, en la página de GitHub nos dirigiremos al apartado Settings/Developer Settings/Personal Access Tokens y seleccionamos Tokens (classic). Aquí le damos a Generate New Token (classic):



En mi caso le voy a poner que no caduque y le doy acceso a repo y workflow:



Y una vez generado el token lo dejamos guardado por ahora en un bloc de notas o en un lugar seguro y ponemos el siguiente comando para que cuando nos pida el token y lo pongamos, se quede guardado permanentemente:

```
git config --global credential.helper store
```

Nos pedirán el Token más adelante.

## Crear y subir el repositorio

El siguiente paso será dirigirnos a la carpeta de nuestro proyecto y transformarla en un repositorio. Para ello nos dirigimos a GitHub y creamos un repositorio:

**Create a new repository**

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (\*).

Owner \* OscarM0ntero / Repository name \* proyecto-angular  
proyecto-angular is available.

Great repository names are short and memorable. Need inspiration? How about [fuzzy-octo-potato](#)?

Description (optional)

☒ Public  
 Anyone on the internet can see this repository. You choose who can commit.

☐ Private  
 You choose who can see and commit to this repository.

Initialize this repository with:

☒ Add a README file  
 This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore  
 .gitignore template: None

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license  
 License: None

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

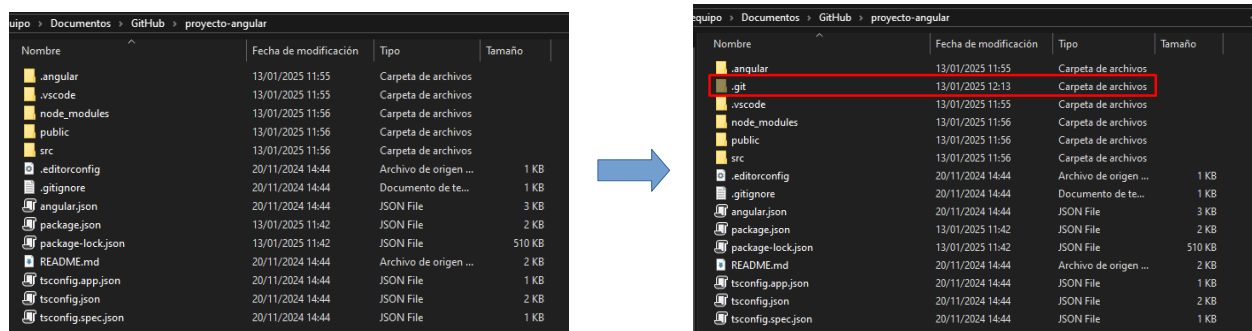
☐ You are creating a public repository in your personal account.

[Create repository](#)

Una vez lo tengamos, cogemos su enlace (<https://github.com/OscarM0ntero/proyecto-angular.git>) y por consola nos dirigimos a la carpeta del proyecto, donde iniciaremos y enlazaremos el repositorio con los siguientes comandos:

```
git init
git remote add origin https://github.com/OscarM0ntero/proyecto-angular.git
```

Una vez iniciado y enlazado el repositorio, en la carpeta del proyecto se creará una carpeta oculta `.git`:



13/01/25

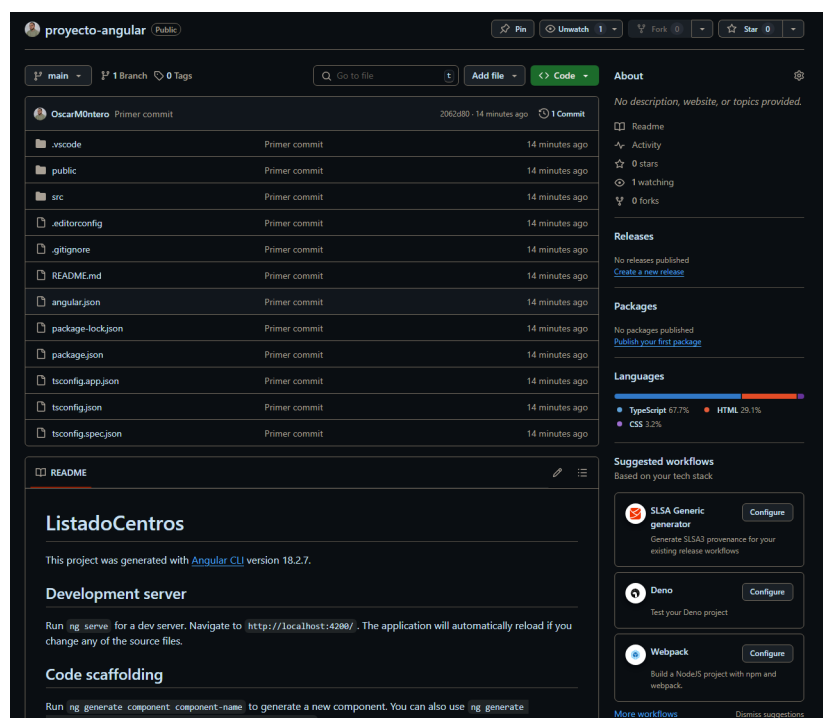
Ahora añadimos el contenido al commit y pusheamos el commit a la línea principal (main):

```
git add .
git commit -m "Primer commit"
git branch -M main
git push -u origin main
```

A la hora de hacer el push, nos pedirá un usuario y una contraseña, pondremos el usuario que queramos, pero la contraseña será el Token previamente creado:

```
PS C:\Users\oscar\Documents\GitHub\proyecto-angular> git push -u origin main
fatal: helper error (-1): User cancelled dialog.
Username for 'https://github.com': OscarM0ntero
Password for 'https://OscarM0ntero@github.com':
Enumerating objects: 68, done.
Counting objects: 100% (68/68), done.
Delta compression using up to 24 threads
Compressing objects: 100% (65/65), done.
Writing objects: 100% (68/68), 211.14 KiB | 11.11 MiB/s, done.
Total 68 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), done.
To https://github.com/OscarM0ntero/proyecto-angular.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
PS C:\Users\oscar\Documents\GitHub\proyecto-angular>
```

Y ya estaría nuestro proyecto subido en GitHub:



Oscar Montero Hinojosa  
Desarrollo de Interfaces

## Configurar GitHub Pages

Vamos a comenzar creando una rama o branch de el repositorio llamada gh-pages, que será la que utilizaremos para el GitHub Pages. Seguimos los siguientes pasos:

1. Nos aseguramos que estamos en la rama main

```
git checkout main
```

2. Creo la rama gh-pages

```
git branch gh-pages
```

3. Cambio a la rama gh-pages

```
git checkout gh-pages
```

4. Añado el contenido del proyecto compilado

```
ng build --configuration=production --output-path=dist/my-app
--base-href=/proyecto-angular/
```

```
cp -r dist/my-app/* .
```

5. Subo la rama al repositorio remoto

```
git add .
git commit -m "Deploy a GitHub Pages"
git push origin gh-pages
```

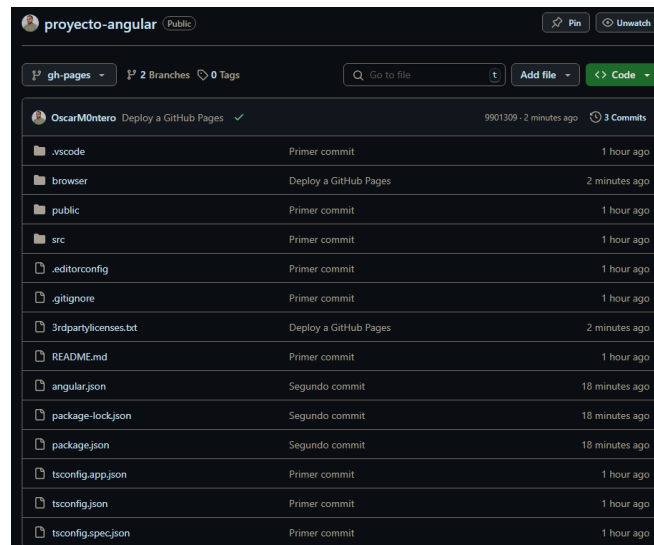
```
PS C:\Users\oscar\Documents\GitHub\proyecto-angular> git branch gh-pages
Switched to branch 'gh-pages'
PS C:\Users\oscar\Documents\GitHub\proyecto-angular> ng build --configuration=production --output-path=dist/my-app --base-href=/proyecto-angular/
Initial chunk files      Names      Raw size | Estimated transfer size
styles-D2G8BGD.css      styles      231.61 kB | 22.68 kB
main-TDK2NDAP.js        main       161.55 kB | 44.63 kB
polyfills-FFW0D2TL.js    polyfills   34.52 kB | 11.28 kB
| Initial total | 427.68 kB | 78.59 kB

Application bundle generation complete. [3.985 seconds]

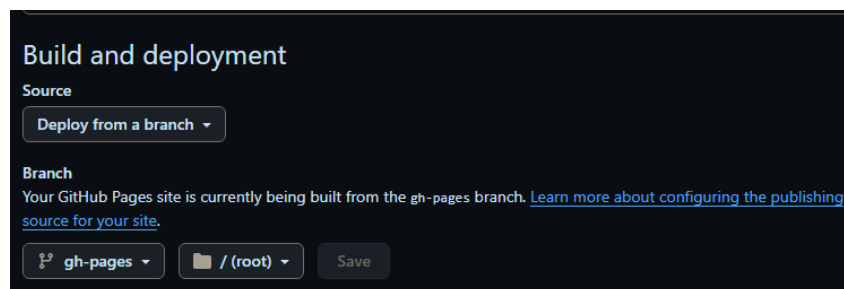
2 rules skipped due to selector errors:
.form-floating~label -> Did not expect successive traversals.
.form-floating~label -> Did not expect successive traversals.

Output location: C:\Users\oscar\Documents\GitHub\proyecto-angular\dist\my-app
PS C:\Users\oscar\Documents\GitHub\proyecto-angular> cp -r dist/my-app/* .
PS C:\Users\oscar\Documents\GitHub\proyecto-angular> git add .
warning: in the working copy of '3rdpartylicenses.txt', it will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'browser/main-TDK2NDAP.js', it will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'browser/polyfills-FFW0D2TL.js', it will be replaced by CRLF the next time Git touches it
PS C:\Users\oscar\Documents\GitHub\proyecto-angular> git commit -m "Deploy a GitHub Pages"
[gh-pages 9981389] Deploy a GitHub Pages
10 files changed, 418 insertions(+)
create mode 100644 3rdpartylicenses.txt
create mode 100644 browser/favicon.ico
create mode 100644 browser/index.html
create mode 100644 browser/logo-jacaranda.jpeg
create mode 100644 browser/logo-sanantiales.jpg
create mode 100644 browser/logo-playmer.png
create mode 100644 browser/logo-rosaleda.jpg
create mode 100644 browser/main-TDK2NDAP.js
create mode 100644 browser/polyfills-FFW0D2TL.js
create mode 100644 browser/styles-D2G8BGD.css
PS C:\Users\oscar\Documents\GitHub\proyecto-angular> git push origin gh-pages
Enumerating objects: 14, done.
Counting objects: 100% (14/14), done.
Delta compression using up to 24 threads
Compressing objects: 100% (13/13), done.
Writing objects: 100% (13/13), 182.81 KiB | 15.23 MiB/s, done.
Total 13 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'gh-pages' on GitHub by visiting:
remote:   https://github.com/OscarMontero/proyecto-angular/pull/new/gh-pages
remote:
To https://github.com/OscarMontero/proyecto-angular.git
 * [new branch] gh-pages -> gh-pages
PS C:\Users\oscar\Documents\GitHub\proyecto-angular>
```

Ya tenemos la rama gh-pages en nuestro repositorio remoto.



Ahora nos dirigimos en el repositorio a la Settings/Pages y podremos comprobar que ya se ha seleccionado la rama:



Ya tenemos nuestro GitHub Pages funcionando, en mi caso en el siguiente enlace:

<https://oscarmontero.github.io/proyecto-angular/>

## Scripts para subida

Comenzamos instalando angular-cli-ghpages:

```
npm install -g angular-cli-ghpages
```

En angular.json, nos aseguramos que builder finalice en browser, no application, outputPath apunte a la carpeta dist/my-app y la opción "browser" la sustituimos por "main" :

```
"architect": {
  "build": {
    "builder": "@angular-devkit/build-angular:browser",
    "options": {
      "outputPath": "dist/my-app",
      "index": "src/index.html",
      "main": "src/main.ts",
      "polyfills": [
        "zone.js"
      ],
```

A continuación, nos dirigimos a package.json y añadimos el siguiente script:

```
"deploy": "ng build --configuration=production --base-href=/proyecto-angular/ && npx angular-cli-ghpages --dir=dist/my-app"
```

```
"scripts": {
  "ng": "ng",
  "start": "ng serve",
  "build": "ng build",
  "watch": "ng build --watch --configuration development",
  "test": "ng test",
  "deploy": "ng build --configuration=production --base-href=/proyecto-angular/ && npx angular-cli-ghpages --dir=dist/my-app"
},
```

Y al realizar el siguiente comando, conseguiremos el deploy de nuestro proyecto compilado en GitHub Pages:

*npm run deploy*

```
PS C:\Users\oscar\Documents\Github\proyecto-angular> npm run deploy
> proyecto-angular@0.0.0 deploy
> ng build --configuration=production --base-href=/proyecto-angular/ && npx angular-cli-ghpages --dir=dist/my-app
✓ Browser application bundle generation complete.
✓ Copying assets complete.
✓ Generating index.html...2 rules skipped due to selector errors:
  .form-floating>label -> Did not expect successive traversals.
  .form-floating>label -> Did not expect successive traversals.
✓ Index html generation complete.

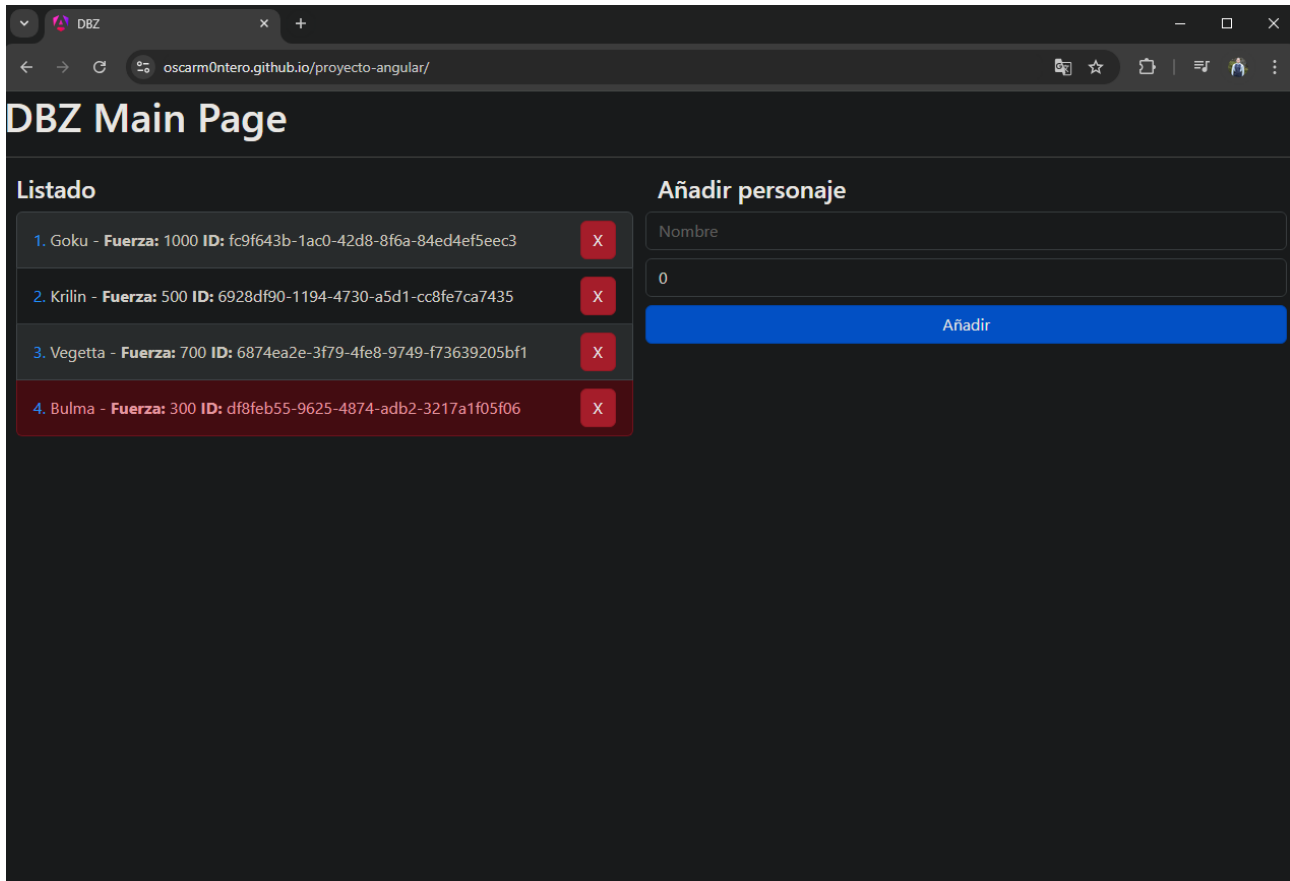
Initial chunk files | Names | Raw size | Estimated transfer size
styles.144363e0b027c0f.css | styles | 224.04 kB | 22.11 kB
main.504ce03cccf5e16d.js | main | 160.21 kB | 44.29 kB
polyfills.8884e4c769e2fcf3.js | polyfills | 34.80 kB | 11.37 kB
runtime.ff48e7a9c337b2dd.js | runtime | 890 bytes | 506 bytes
Initial total | 419.94 kB | 78.48 kB

Build at: 2025-01-13T13:14:32.369Z - Hash: cb9ae4389417c63c - Time: 6066ms
404.html file created
.nojeekyll file created
✎ Uploading via git, please wait...
Cloning https://github.com/OscarMontero/proyecto-angular.git into C:\Users\oscar\Documents\Github\proyecto-angular
Cleaning
Fetching origin
Checking out origin/gh-pages
Removing files
Copying files
Adding all
Committing
Pushing
★ Successfully published via angular-cli-ghpages! Have a nice day!
```

## Resultado final

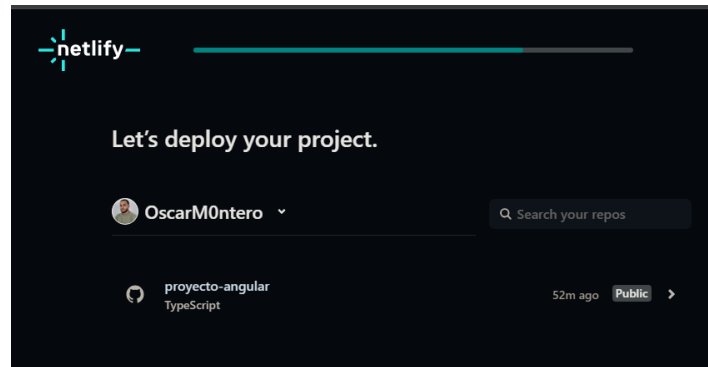
Podemos verificar el resultado en:

<https://oscarm0ntero.github.io/proyecto-angular/>

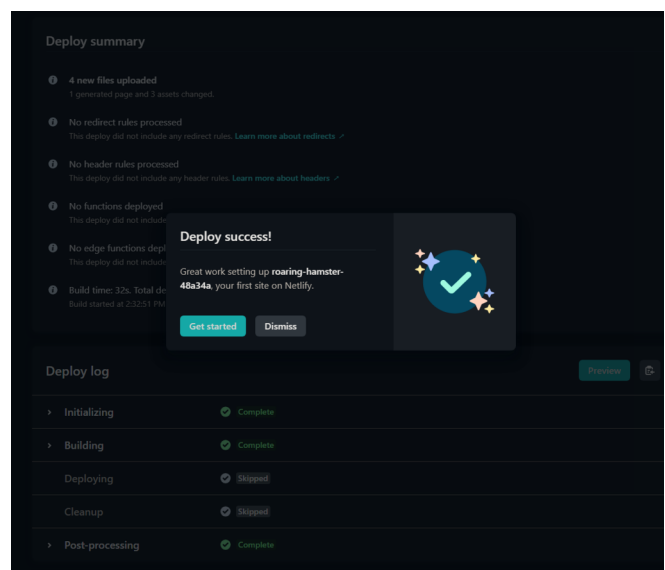
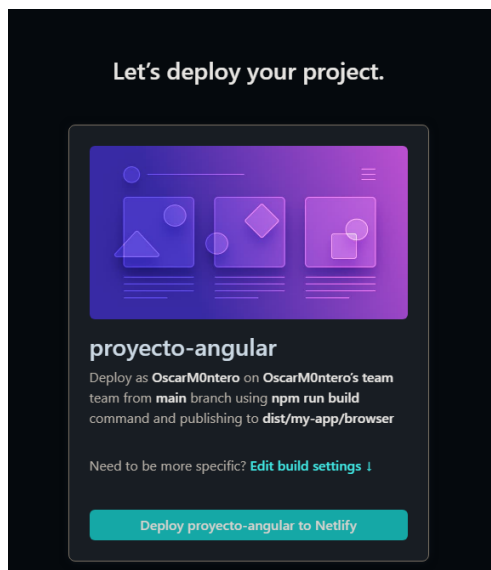


## Desplegar proyecto en Netfly

Nos registramos en Netfly con nuestra cuenta y en el proceso de registro nos permite seleccionar el repositorio que queramos desplegar.



Le damos click y nos dejara desplegarlo.



Ya lo tenemos desplegado en el siguiente enlace:

<https://roaring-hamster-48a34a.netlify.app/>



## Desplegar proyecto en http-server local

1. Instalamos http-server:

```
npm install -g http-server
```

2. Compilamos en modo producción:

```
ng build --configuration=production
```

3. Lo servimos con:

```
http-server ./dist/my-app
```

```
PS C:\Users\oscar\Documents\GitHub\projecto-angular> npm install -g http-server
added 9 packages, removed 5 packages, and changed 41 packages in 3s

15 packages are looking for funding
  run `npm fund` for details
PS C:\Users\oscar\Documents\GitHub\projecto-angular> ng build --configuration=production
✔ Browser application bundle generation complete.
✔ Copying assets complete.
✔ Generating index.html... 2 rules skipped due to selector errors:
  .form-floating~label -> Did not expect successive traversals.
  .form-floating~label -> Did not expect successive traversals.
✔ Index html generation complete.

Initial chunk files | Names | Raw size | Estimated transfer size
styles.144363acbb827cdf.css | styles | 224.04 kB | 22.31 kB
main.504ce83cca5e16d.js | main | 160.21 kB | 44.29 kB
polyfills.8084eac789e2fcf3.js | polyfills | 34.80 kB | 11.37 kB
runtime.f468e79bc337b2d6.js | runtime | 890 bytes | 584 bytes
| Initial total | 419.94 kB | 78.48 kB

Build at: 2025-01-13T13:37:06.195Z - Hash: cb9ae4383417c63c - Time: 1566ms
PS C:\Users\oscar\Documents\GitHub\projecto-angular> http-server ./dist/my-app
Starting up http-server, serving ./dist/my-app

http-server version: 14.1.1

http-server settings:
  CORS: disabled
  Cache: 3600 seconds
  Connection Timeout: 120 seconds
  Directory Listings: visible
  AutoIndex: visible
  Serve GZIP Files: false
  Serve Brotli Files: false
  Default File Extension: none

Available on:
  http://25.41.156.101:8080
  http://192.168.1.152:8080
  http://192.168.56.1:8080
  http://127.0.0.1:8080
Hit CTRL-C to stop the server
```

En mi caso lo tengo disponible en:

<http://25.41.156.101:8080>

<http://192.168.1.152:8080>

<http://192.168.56.1:8080>

<http://127.0.0.1:8080>

