**Preparar el entorno de desarrollo**

**Objetivo:**

Para preparar el entorno de desarrollo para desarrollar y/o mantener una aplicación desarrollada con el framework Angular.

**Recursos necesarios:**

Es necesario instalar las siguientes aplicaciones o programas:

* Node.js + npm
* Angular CLI
* GIT
* Visual Studio Code

**Node.js** es un intérprete que ejecuta código JavaScript fuera del navegador web, es decir, en el lado del servidor. Node.js utiliza el motor de JavaScript V8 de Google Chrome para interpretar el código JavaScript y proporciona una amplia gama de funciones y bibliotecas para la creación de aplicaciones de servidor y de línea de comandos.

**Angular CLI** es una interfaz de línea de comandos (CLI) para Angular, que es un marco de trabajo (framework) de JavaScript utilizado para crear aplicaciones web del lado del cliente. Angular CLI proporciona un conjunto de herramientas de línea de comandos para crear, compilar, probar y desplegar aplicaciones Angular de manera más eficiente. Con Angular CLI, puedes crear fácilmente componentes, módulos, servicios y otros artefactos de Angular, y también pueden utilizar bibliotecas externas para mejorar la funcionalidad de sus aplicaciones. En resumen, es una herramienta que facilita la creación y mantenimiento de proyectos de Angular.

**GIT** es un sistema de control de versiones distribuido de código abierto, que se utiliza para el desarrollo de software y la gestión de código fuente. Permite un seguimiento de los cambios realizados en el código de la aplicación a lo largo del tiempo, así como colaborar con otros desarrolladores en un proyecto. Con GIT, se pueden crear ramas (branches) para trabajar en diferentes versiones del código, realizar cambios y combinaciones (merges) de ramas, y mantener un registro detallado de los cambios realizados en cada archivo.

**Visual Studio Code** es un editor de código fuente desarrollado por Microsoft, que se utiliza para programar en diversos lenguajes de programación como JavaScript, TypeScript, Python, PHP, C#, entre otros. Visual Studio Code es un editor de código gratuito y de código abierto, que se ejecuta en Windows, macOS y Linux. El editor cuenta con una amplia variedad de características, como la finalización de código (code completion), el resaltado de sintaxis, la depuración de código, la integración con control de versiones como Git, la extensibilidad a través de una gran cantidad de complementos y la posibilidad de personalizar la interfaz de usuario a través de temas y configuraciones. Visual Studio Code también cuenta con una gran comunidad de desarrolladores que contribuyen con extensiones y herramientas para el editor.

**Pasos:**

**1. Instalación de nodejs con npm**

Recomendamos instalar la versión v16.19.0

Ingresar a <https://nodejs.org/en/download/releases>

Descargar la versión v16.19.0 y seguir las instrucciones del instalador

**2. Instalación de Angular CLI**

Ejecuta el siguiente comando en una terminal:

npm install -g @angular/cli

**3. Instalación de GIT**

Ingrese a <https://git-scm.com/> descargue el instalador correspondiente a su sistema operativo, descomprima el archivo, ejecute el instalador y siga las instrucciones.

**4. Visual Studio Code**

Ingrese a <https://code.visualstudio.com/> descargue el instalador, ejecútelo y siga las instrucciones.

Una vez instalemos estos programas tenemos preparado el entorno de desarrollo y podremos comenzar a adicionar nuevas funcionalidades y hacer el mantenimiento de Dory web app.

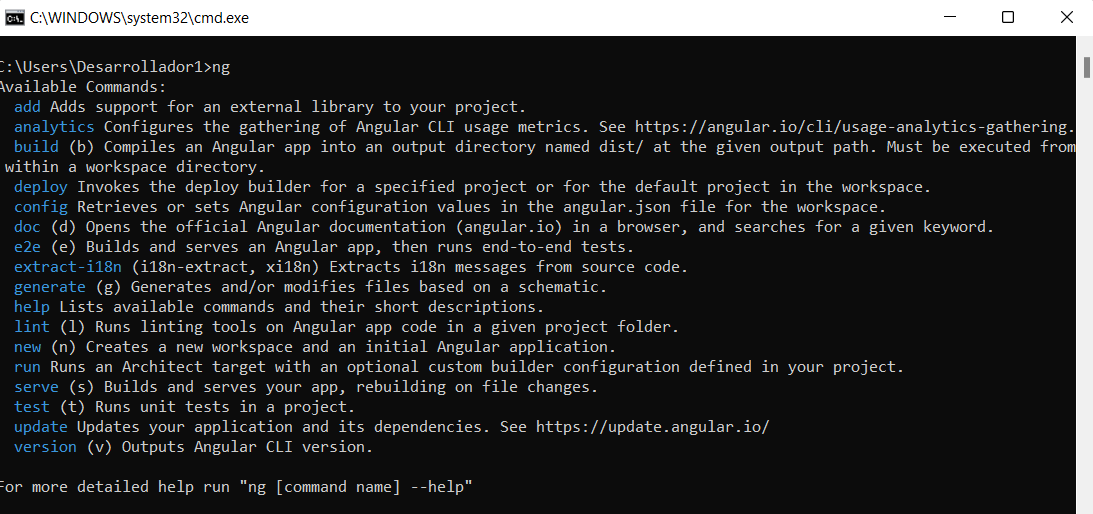
**Verificación:**

Ejecutamos los siguientes comandos

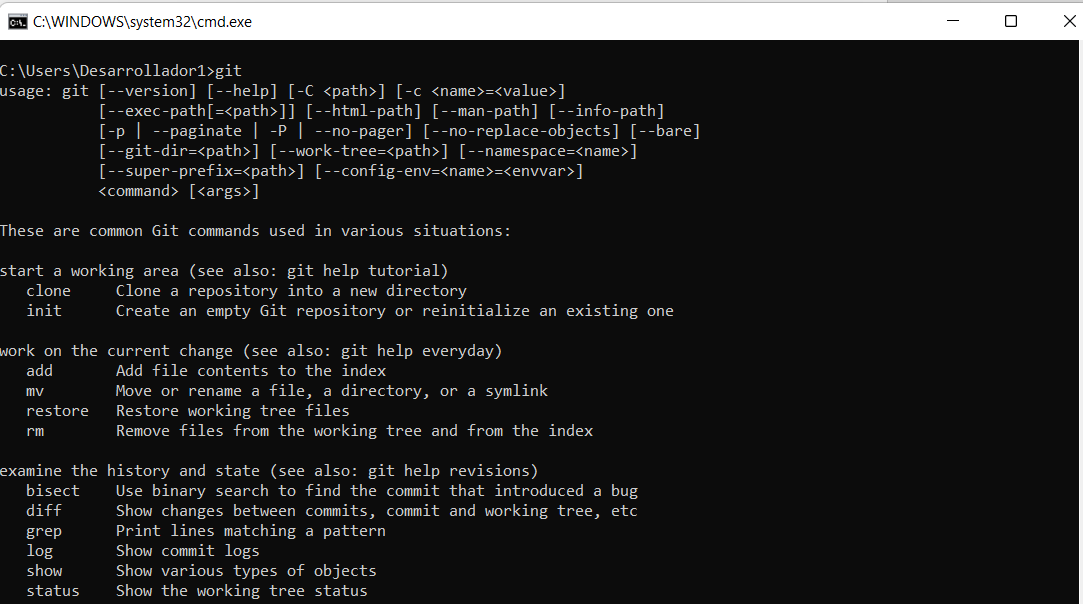
Verificación de la instalación de nodejs



Verificación de la instalación de Angular CLI



Verificación de la instalación de GIT



Verificación de la instalación de Visual Studio Code



**Como clonar el código fuente del Dory web app**

**Objetivo:**

Descargar el código fuente de la aplicación Dory web app para trabajar con este.

**Recursos necesarios:**

* Repositorio donde se almacena el código fuente:

<https://github.com/doryteam1/dory-web-app.git>

* GIT

**Pasos:**

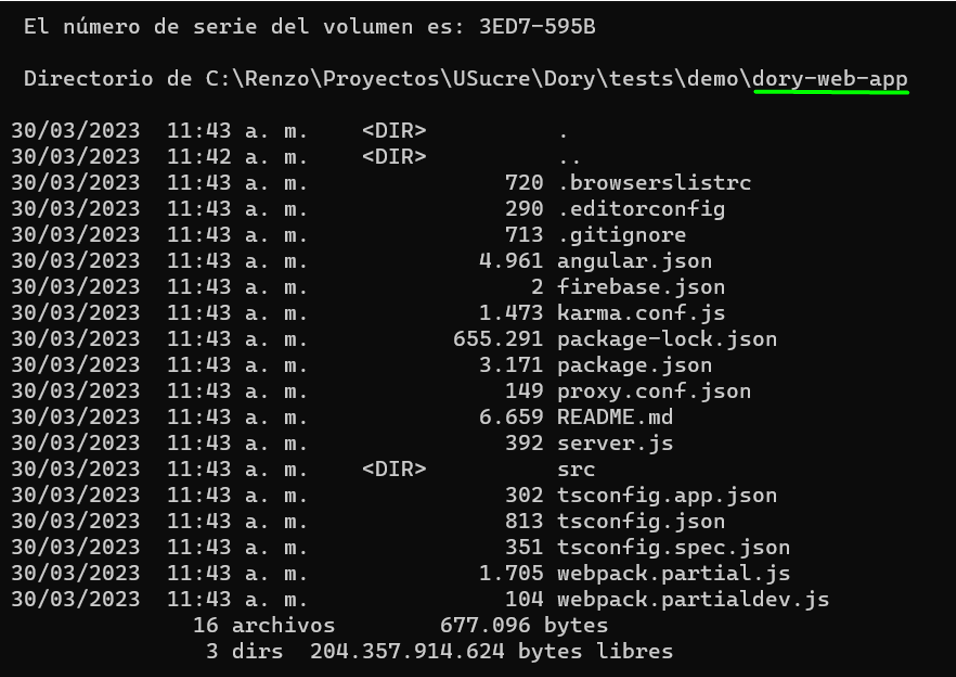
**1. Ejecutamos el comando para clonarlo:**

git clone <https://github.com/doryteam1/dory-web-app.git> –b dev

El comando anterior descarga el código fuente de la rama **dev** del proyecto en una carpeta o directorio en el sistema de archivos.

**Verificación:**

Así se ve la estructura del proyecto clonado



**Como ejecutar Dory web app en un servidor de desarrollo**

**Objetivo:**

Levantar un servidor web de desarrollo para visualizar en local las modificaciones que iremos haciendo al código fuente.

**Recursos necesarios:**

* Angular CLI

**Pasos:**

**1. Para ejecutar el servidor de desarrollo entramos en la carpeta del proyecto recientemente clonado:**

cd dory-web-app

**2. instalamos las dependencias:**

npm install --force

**3. Ejecutamos el servidor de desarrollo:**

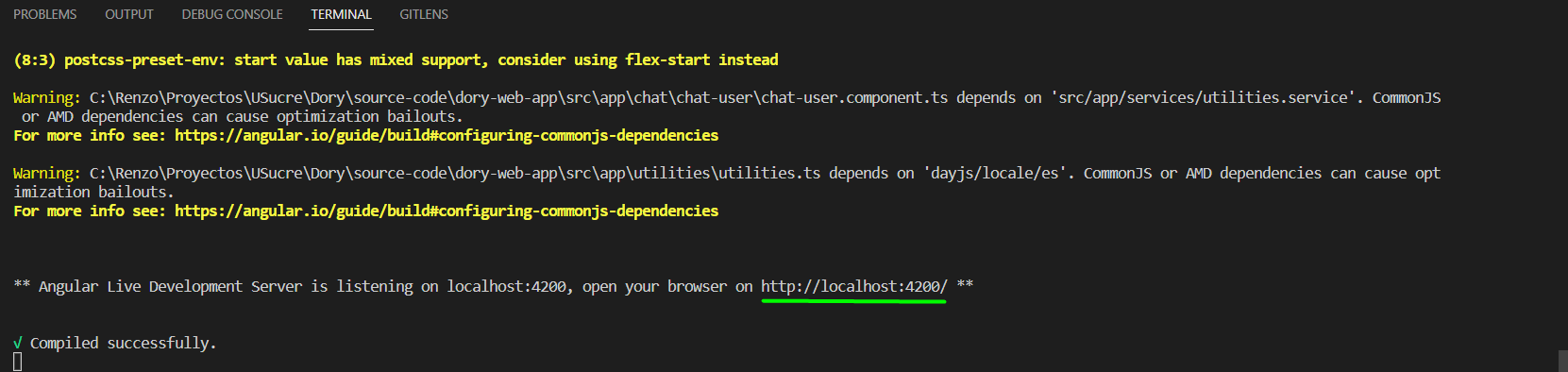
npm run ini

4.

<!------Añadir el archivo .env----------🡪

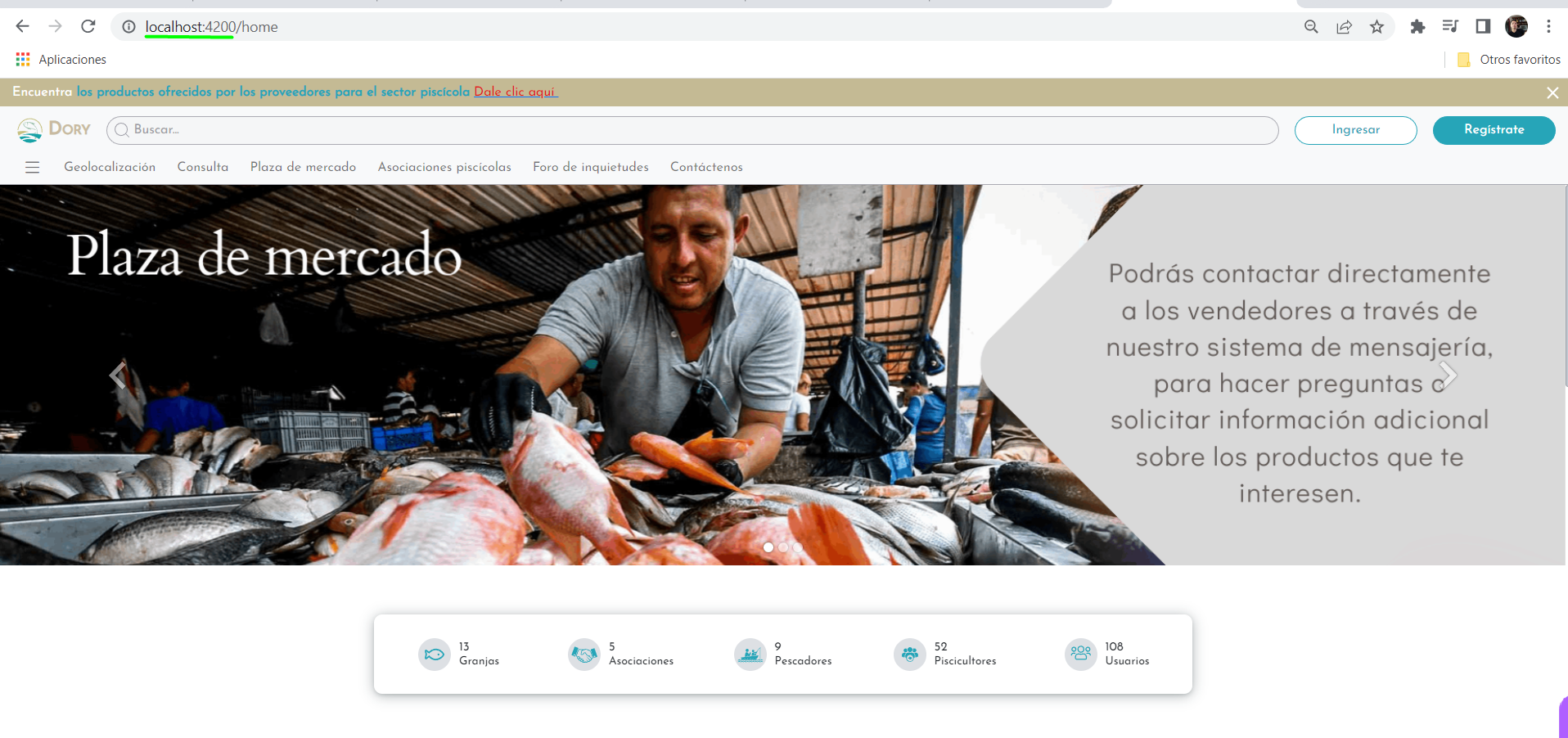
**Verificación**

El resultado de ejecutar el comando del paso 3 es:



Abrimos el navegador(Chrome, Firefox, Edge) e ingresamos la siguiente url:

<http://localhost:4200/>



**Como desplegar Dory web app en Heroku**

**Objetivo:** Colocar en línea la aplicación web Dory para que pueda ser accedida desde internet. Las siguientes instrucciones sirven tanto para un ambiente de pruebas como para un ambiente productivo.

**Recursos necesarios:**

* GIT
* Cuenta en github
* Cuenta en Heroku con mínimo un dyno ECO

**Pasos:**

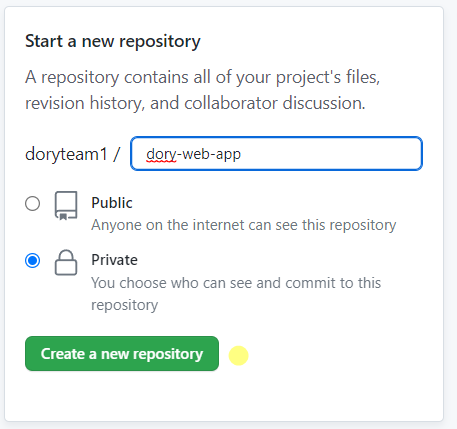
**1. Instale GIT**

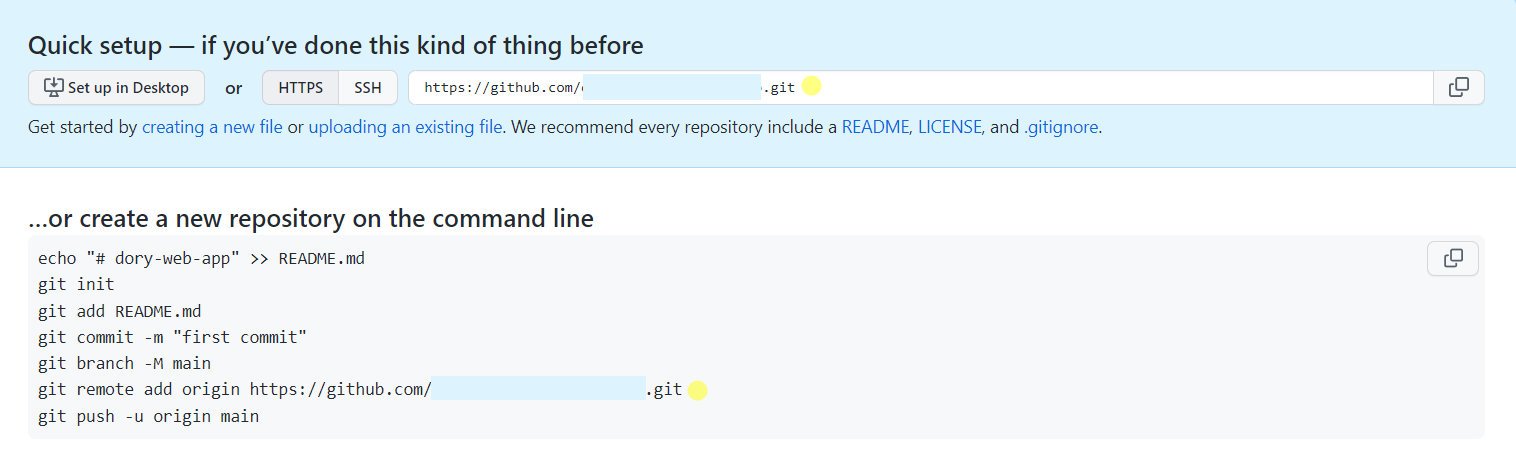
Ingrese al enlace [https://git-scm.com](https://git-scm.com/) y siga la instrucciones para llevar a cabo la instalación.

**2. Ingrese a github**

Cree una cuenta en GitHub en [https://github.com](https://github.com/) o ingrese si ya tiene una.

**3. Cree un nuevo repositorio privado**





Guarde la dirección del nuevo repositorio será usada más adelante.

**4. Acceda al código fuente**

git clone <https://github.com/doryteam1/dory-web-app.git>

cd dory-web-app

Ahora usted se encuentra en el directorio que contiene el código fuente de la aplicación.

**5. Adicione un nuevo remoto**

Estando en el directorio **dory-web-app** ejecute el siguiente comando:

git remote add dory https://github.com/\*\*\*.git

Remplace la url usada en el comando por la url del nuevo repositorio guardada en el punto 3.

Esto adiciona un nuevo remoto en el repositorio **dory-web-app**

**6. Cree la rama “master”**

git checkout -b master

**7. Suba el código al nuevo repositorio**

git push dory master

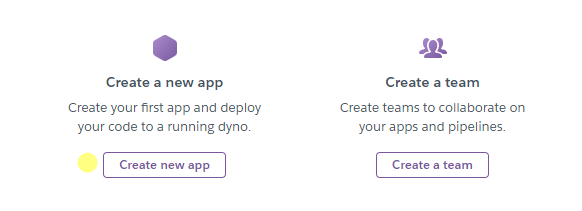
Ahora el código fuente de la aplicación se encuentra en el nuevo repositorio.

Nota: Asegúrese de tener configurada la cuenta de git con un usuario que tenga los permisos necesarios para realizar este procedimiento.

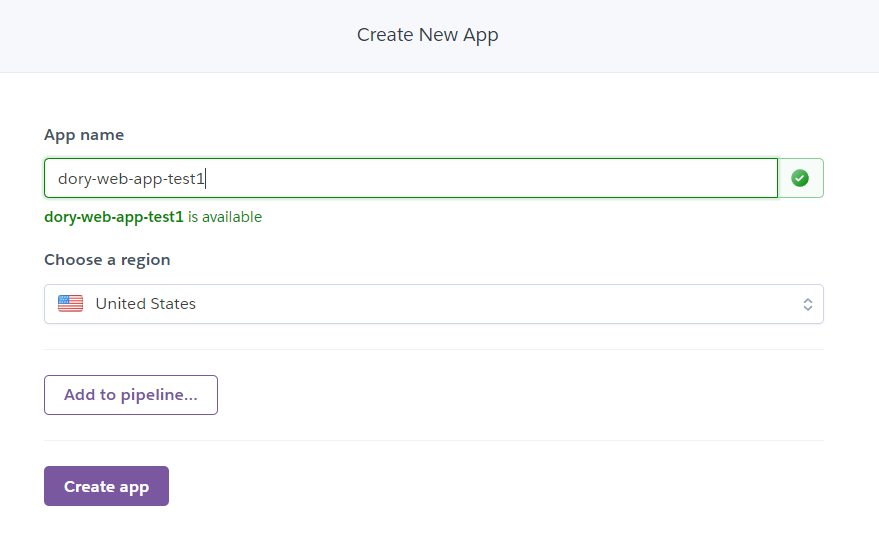
**8. Ingrese a Heroku**

Cree una cuenta en Heroku en <https://signup.heroku.com/login> o ingrese si ya tiene una.

**9. Cree una app en Heroku**



Heroku usa contenedores para ejecutar y escalar todas las aplicaciones. Estos contenedores se denominan Dynos.



Remplace el nombre de aplicación(App name) por uno que prefiera.

Diligencie el formulario y haga clic en el botón **Create app**

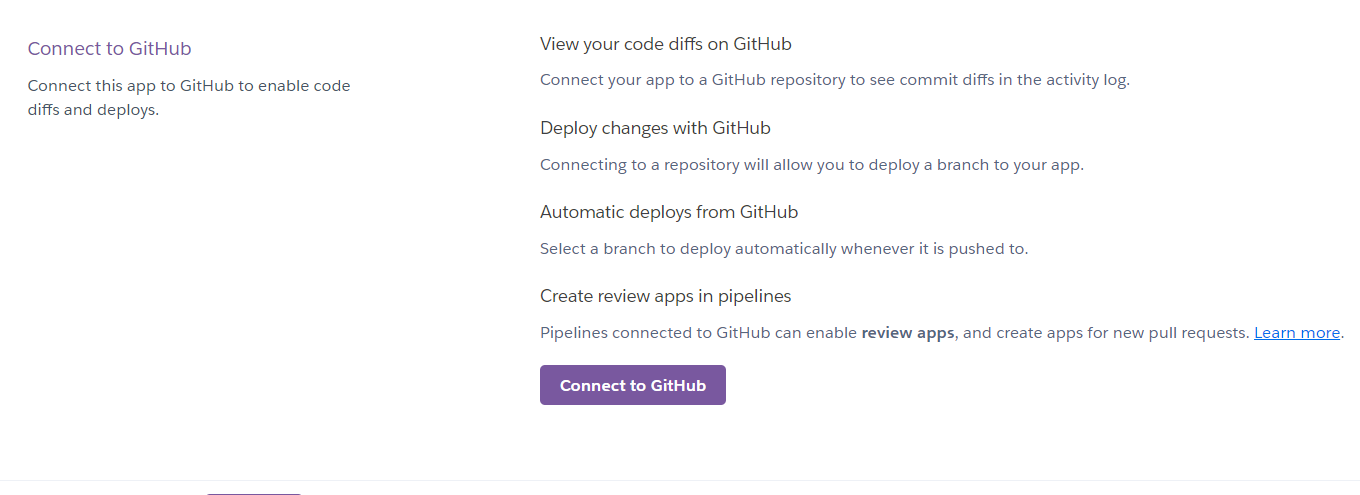
**10. Configurar App**

Para configurar la app en Heroku realice los siguientes pasos:

a. Seleccione GitHub como método de despliegue.

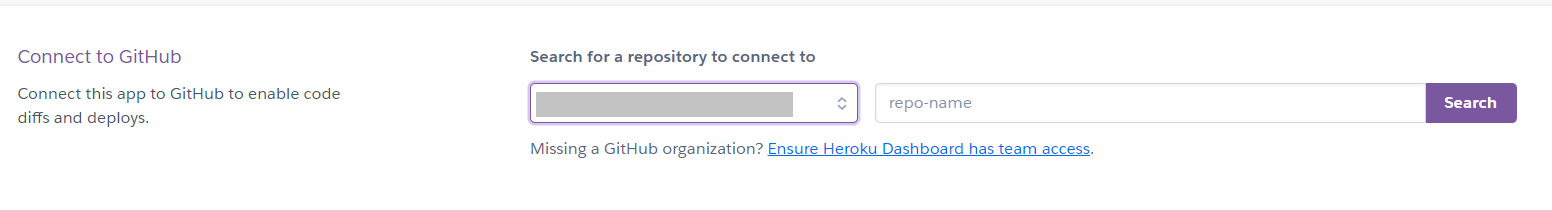


b. Presione el botón Connect to GitHub

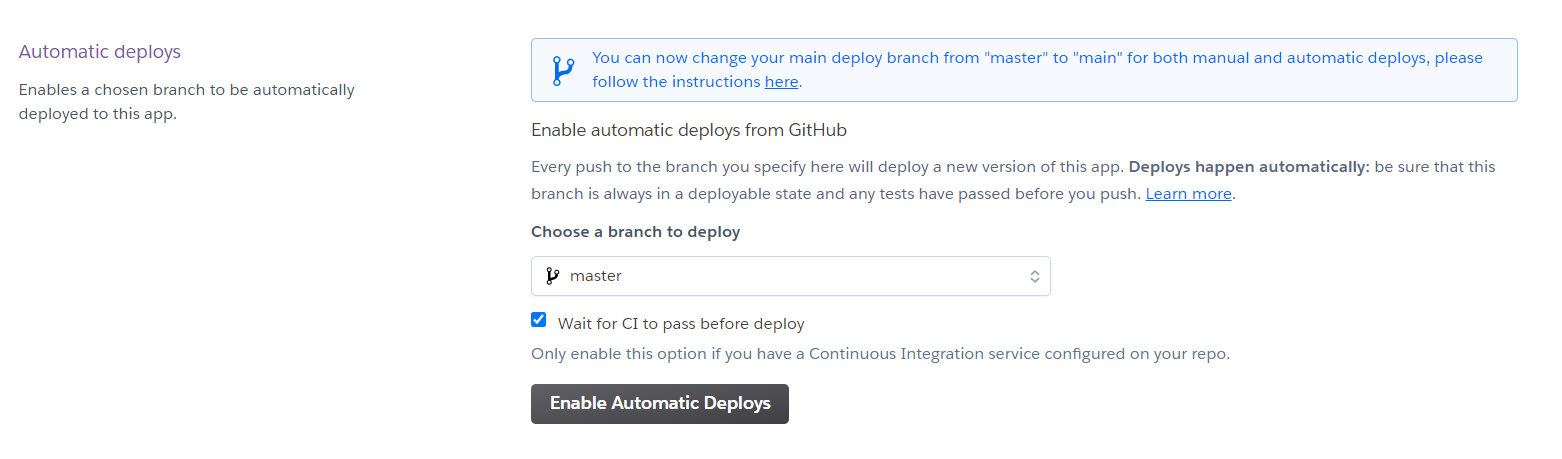
****

c. Inicie sesión con su usuario y contraseña de GitHub para conectarse a Heroku

d. Conecte la App con el nuevo repositorio

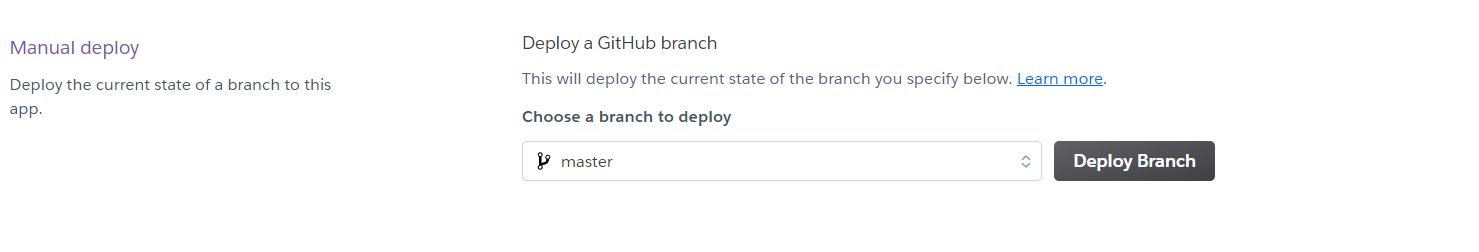
Seleccione el nuevo repositorio creado en el punto 3

e. Seleccione la rama master para despliegues automáticos.



presione el botón Enable Automatic Deploys para activar los despliegues automáticos.

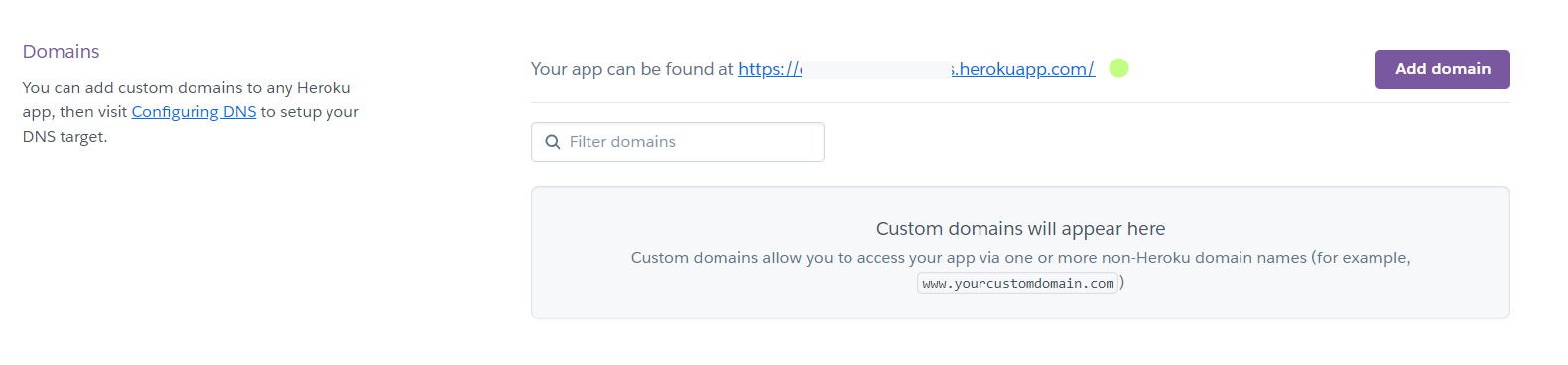
**12. Despliegue la App**

****

Presione el botón Deploy Branch para realizar un despliegue manual de la rama master.

Una vez terminado el proceso la app se encuentra online.

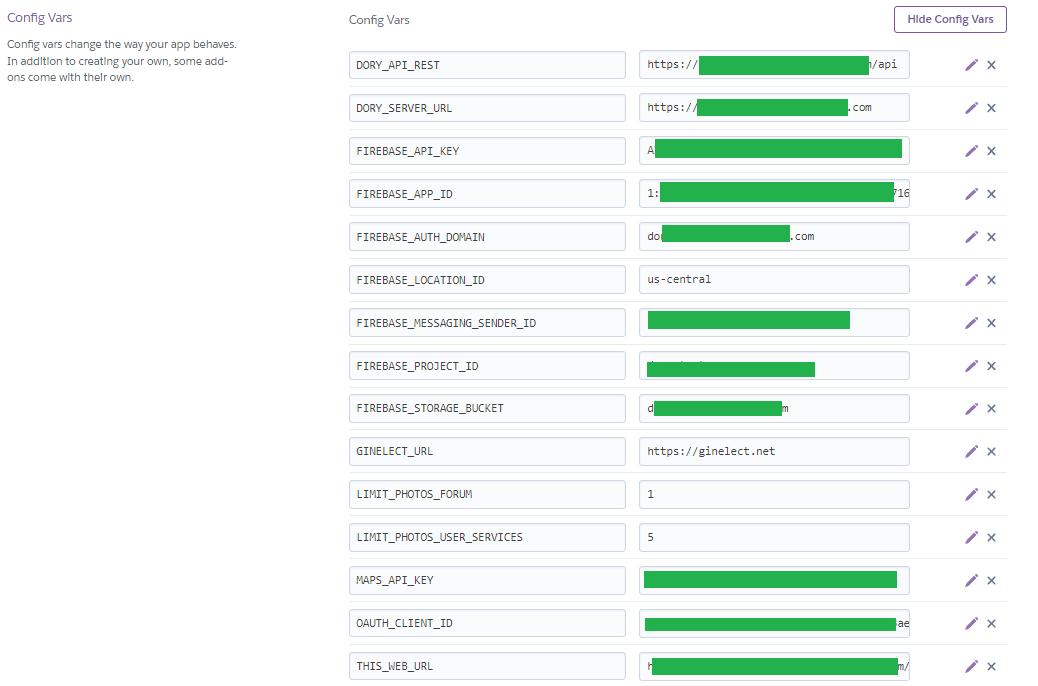
**Verificación:**



En la sección Settings de la app encontrará la sección “Domains”. En esta encontrará la url para acceder a la aplicación web. Insertela en el navegador para acceder a esta y verificar su correcto funcionamiento.

**Como configurar las variables de entorno en Heroku**

**Objetivo:** Como y que variables de entorno configurar en Heroku para conectar la aplicación con servicios externos como el Api Rest de Dory, Firebase Storage, Google Identity y el mapa de google.



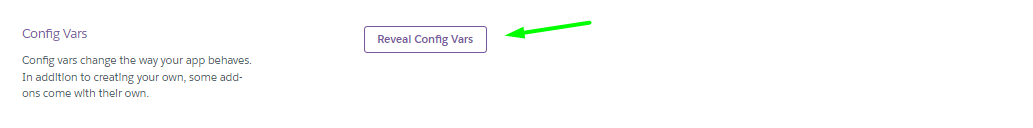
**Recursos necesarios:**

* Cuenta en Heroku con mínimo un dyno ECO

**Pasos:**

**1. En la pestaña “Settings” de la aplicación en la sección “Config Vars” en Heroku presionar “Reveal Config Vars”.**





**2. Crear las siguientes variables de entorno:**

**DORY\_API\_REST**: Url base del api rest de la plataforma.

**DORY\_SERVER\_URL:** Url del servidor donde se encuentra el api rest.

**GINELECT\_URL:** La url del sitio web del grupo de investigación Ginelect.

**LIMIT\_PHOTOS\_FORUM:** Usado para limitar el número de fotos que se pueden compartir en el foro.

**LIMIT\_PHOTOS\_USER\_SERVICES:** Limite impuesto en frontend para el numero de fotos o imágenes que se pueden subir en el storage de Firebase.

**MAPS\_API\_KEY:** Api key de google. Puedes generar una en el siguiente enlace: https://developers.google.com/maps/documentation/javascript/get-api-key

**OAUTH\_CLIENT\_ID:** ID de cliente. Siga los pasos para obtener uno en: <https://developers.google.com/identity/protocols/oauth2>

**THIS\_WEB\_URL:** Url o dominio donde se encuentra esta aplicación web. Lo puede encontrar en la pestaña “Settings” de la app en la sección “Domains”.

Para configurar las variables de entorno relacionadas con Firebase use el objeto “firebaseConfig”. Este lo puede obtener en el proyecto de Firebase creado.

**FIREBASE\_API\_KEY**

**FIREBASE\_APP\_ID**

**FIREBASE\_AUTH\_DOMAIN**

**FIREBASE\_LOCATION\_ID**

**FIREBASE\_MESSAGING\_SENDER\_ID**

**FIREBASE\_PROJECT\_ID**

**FIREBASE\_STORAGE\_BUCKET**

**Como desplegar el Api rest de Dory en Heroku**

**Objetivo:** Desplegar el api rest de la plataforma Dory en Heroku.

**Recursos necesarios:**

* Servidor de base de datos

**Pasos:**

**1. Instalar servidor de base de datos MySQL**

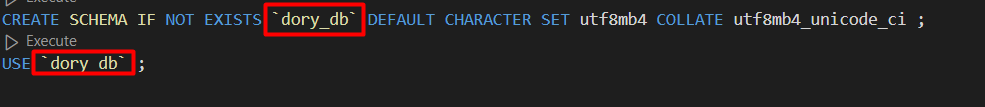
Instale el servidor de base de datos en un servicio de la nube de su preferencia.

**2. Guarde el hostname, nombre de la base de datos, nombre de usuario y password de la base de datos. Estos se usarán en los siguientes pasos.**

**3. Ejecutar el script createDB.sql en el servidor mysql, para crear la base de datos. Este script se encuentra en el siguiente repositorio, en la carpeta scripts/creation:**

<https://github.com/doryteam1/dory-project-files.git>

**Nota:** Debe ingresar al script y cambiar el nombre de la base de datos por la recientemente creada.



**4. Ejecutar el script poblarBD.sql en la base de datos creada en el punto 1, para establecer los datos iniciales. Este se encuentra en el repositorio:**

<https://github.com/doryteam1/dory-project-files.git>

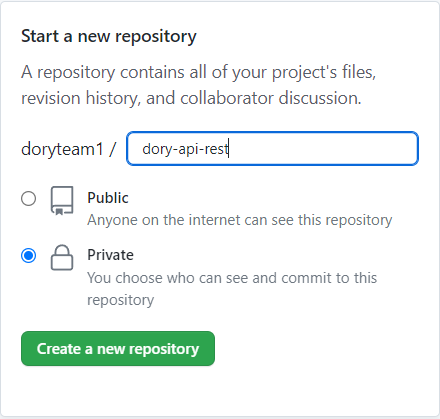
**2. Instale GIT**

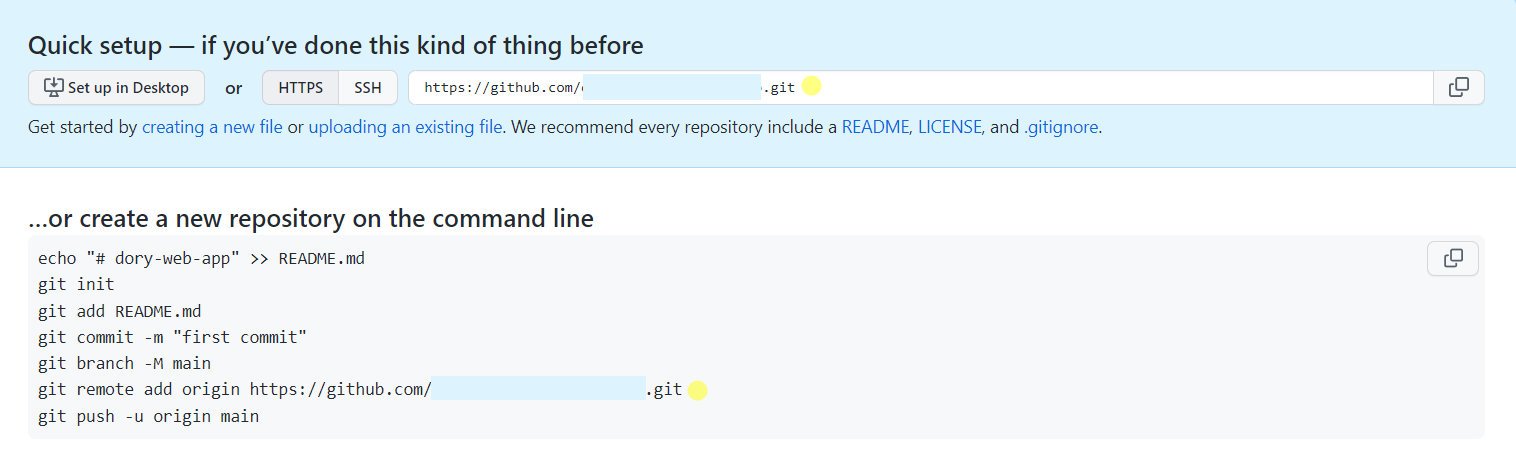
Ingrese al enlace [https://git-scm.com](https://git-scm.com/) y siga las instrucciones para llevar a cabo la instalación.

**3. Ingrese a GitHub**

Cree una cuenta en GitHub en [https://github.com](https://github.com/) o ingrese si ya tiene una.

**4. Cree un nuevo repositorio privado**





Guarde la dirección del nuevo repositorio será usada más adelante.

**5. Acceda al código fuente**

git clone https://github.com/doryteam1/dory-api-rest.git

cd dory-api-rest

Ahora usted se encuentra en el directorio que contiene el código fuente de la aplicación.

**6. Adicione un nuevo remoto**

Estando en el directorio **dory-api-rest** ejecute el siguiente comando:

git remote add dory https://github.com/\*\*\*.git

Remplace la url usada en el comando por la url del nuevo repositorio guardada en el punto 4.

Esto adiciona un nuevo remoto en el repositorio **dory-api-rest**

**7. Cree la rama “master”**

git checkout -b master

**8. Suba el código al nuevo repositorio**

git push dory master

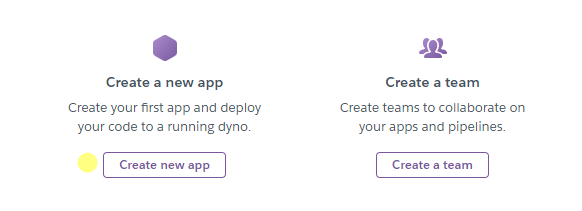
Ahora el código fuente de la aplicación se encuentra en el nuevo repositorio.

**Nota:** Asegúrese de tener configurada la cuenta de git con un usuario que tenga los permisos necesarios para realizar este procedimiento.

**9. Ingrese a Heroku**

Cree una cuenta en Heroku en <https://signup.heroku.com/login> o ingrese si ya tiene una.

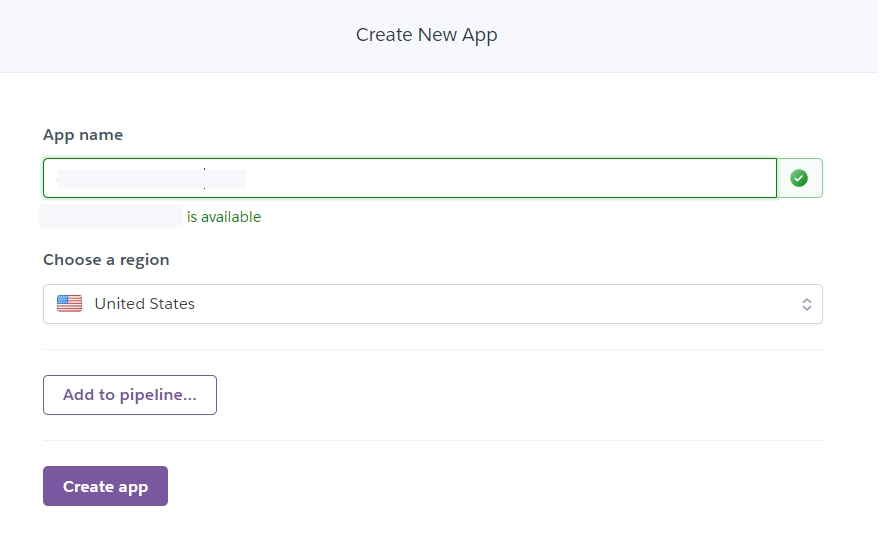
**10. Cree una app en Heroku**



Heroku usa contenedores para ejecutar y escalar todas las aplicaciones. Estos contenedores se denominan Dynos.

Remplace el nombre de la aplicación (App name) por el de su preferencia.

Diligencie el formulario y haga clic en el botón **Create app**

****

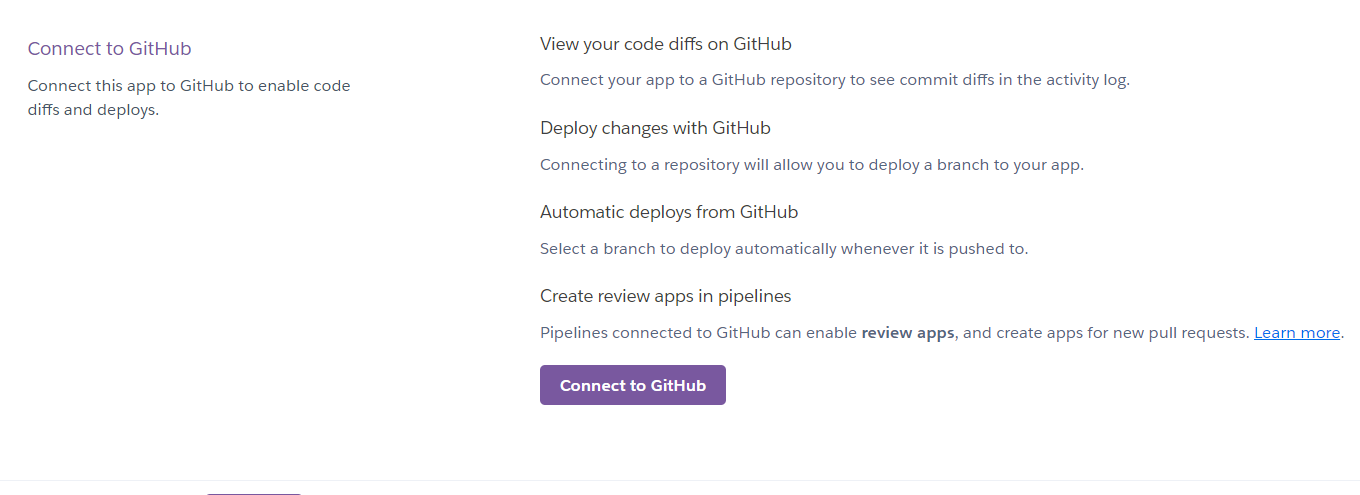
**11. Configurar App**

Para configurar la app en Heroku realice los siguientes pasos:

1. Seleccione GitHub como método de despliegue.

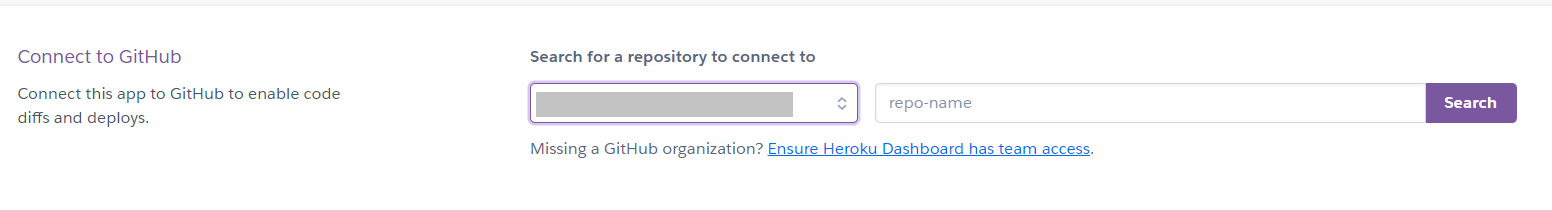


2. Presione el botón Connect to GitHub

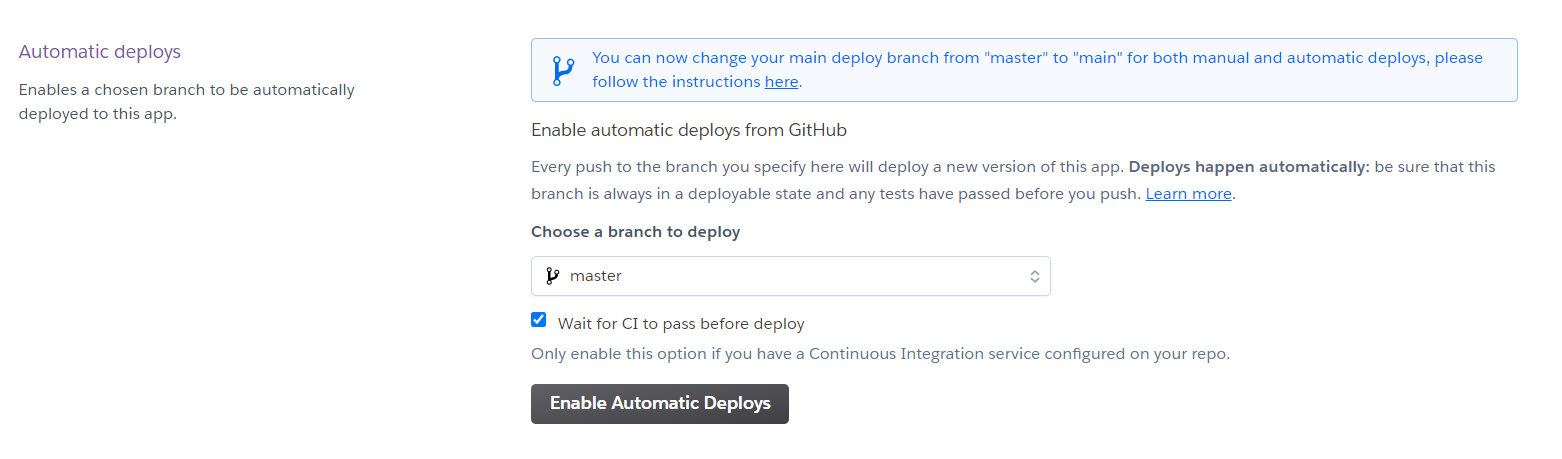
****

3. Inicie sesión con su usuario y contraseña de GitHub para conectarse a Heroku

4. Conecte la App con el nuevo repositorio

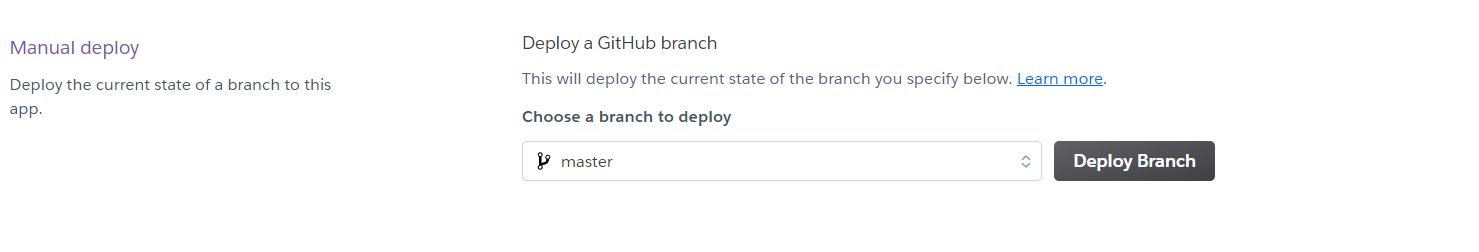
Seleccione el nuevo repositorio creado en el punto 3

5. Seleccione la rama master para despliegues automáticos.



Presione el botón **Enable Automatic Deploys** para activar los despliegues automáticos.

**12. Despliegue la App**

****

Presione el botón **Deploy Branch** para realizar un despliegue manual de la rama master.

Una vez terminado el proceso la app se encuentra online.

**13. Configure las Variables de entorno en Heroku**



En la pestaña “**Settings**” de la aplicación en la sección “**Config Vars**” configurar las siguientes variables de entorno:

**DB\_HOST**: Host donde se encuentra alojado el servidor de bases de datos mysql.

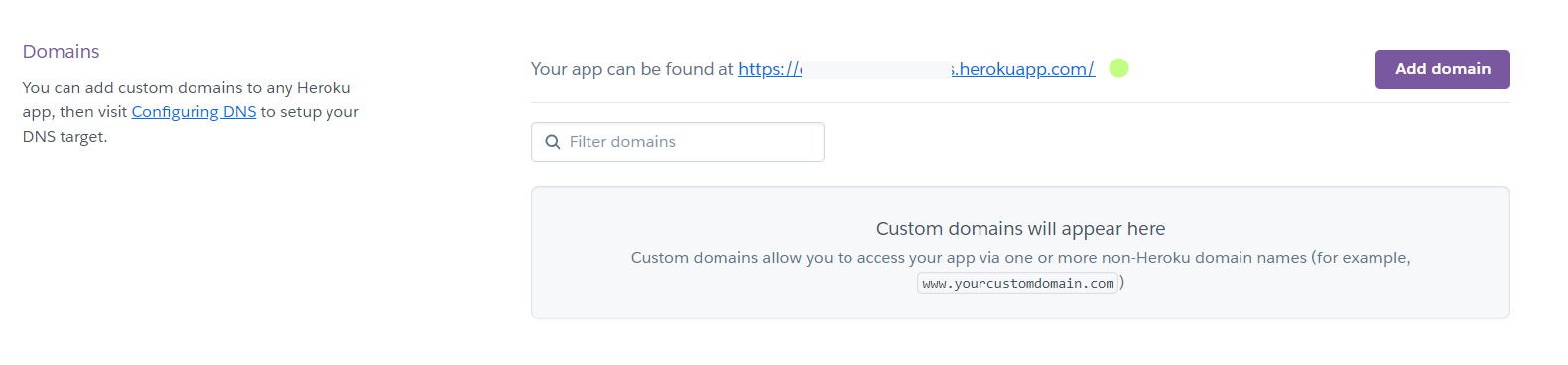
**DB\_NAME:** Nombre de la base de datos.

**DB\_USER:** Usuario de la base de datos.

**DB\_PASSWORD:** Password de usuario de la base de datos

**DORY\_WEB\_APP\_URL:** Url de la aplicación web dory (La puede obtener una vez termine la guía de instalación de la aplicación web).

**Verificación:**



En la sección **Settings** de la app encontrará la sección “**Domains**”. En esta encontrará la url base para acceder al api rest. Añada el sufijo api al final de la dirección y verifique su funcionamiento en [postman](https://www.postman.com/).

**Como modificar los componentes fijos e intercambiables de Dory web app**

**Objetivo:**

Aprender a modificar la página web Dory(modificando componentes fijos e intercambiables).

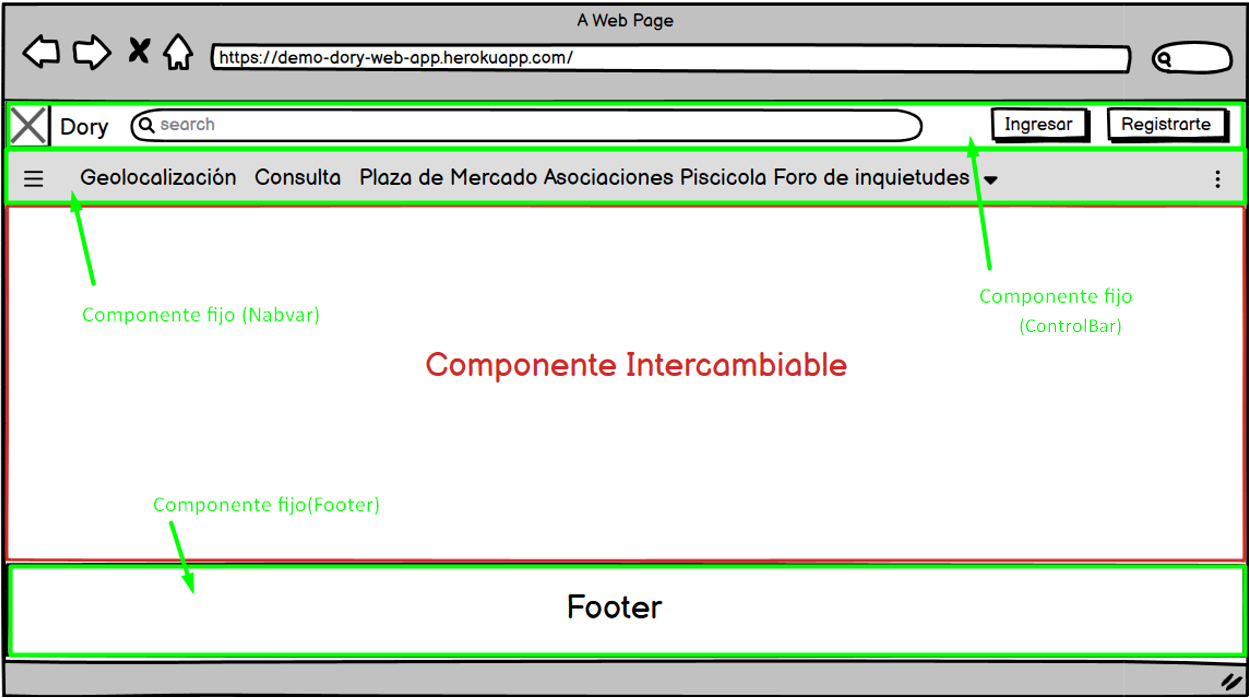
**Recursos necesarios:**

* Visual Studio Code
* Chrome, Firefox

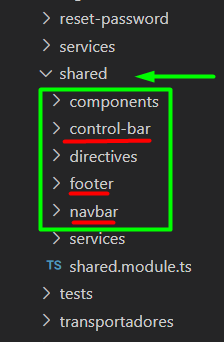
**Explicación:**

Dory tiene dos tipos de componentes. Los fijos y los intercambiables. Los que se resaltan en color verde son los fijos y los rojos son intercambiables.

Los componentes fijos se renderizan en la primera carga y solo se cargan nuevamente cuando el usuario actualiza la pagina(F5). Por el contrario los componentes intercambiables van siendo remplazados por otros a medida que el usuario va navegando por las opciones de menú del componente **Navbar**, los controles del **ControlBar**, los enlaces del **Footer** y redirecciones desde un componente intercambiable a otro.



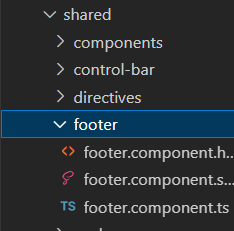
Los componentes fijos se encuentran en la carpeta “shared”.



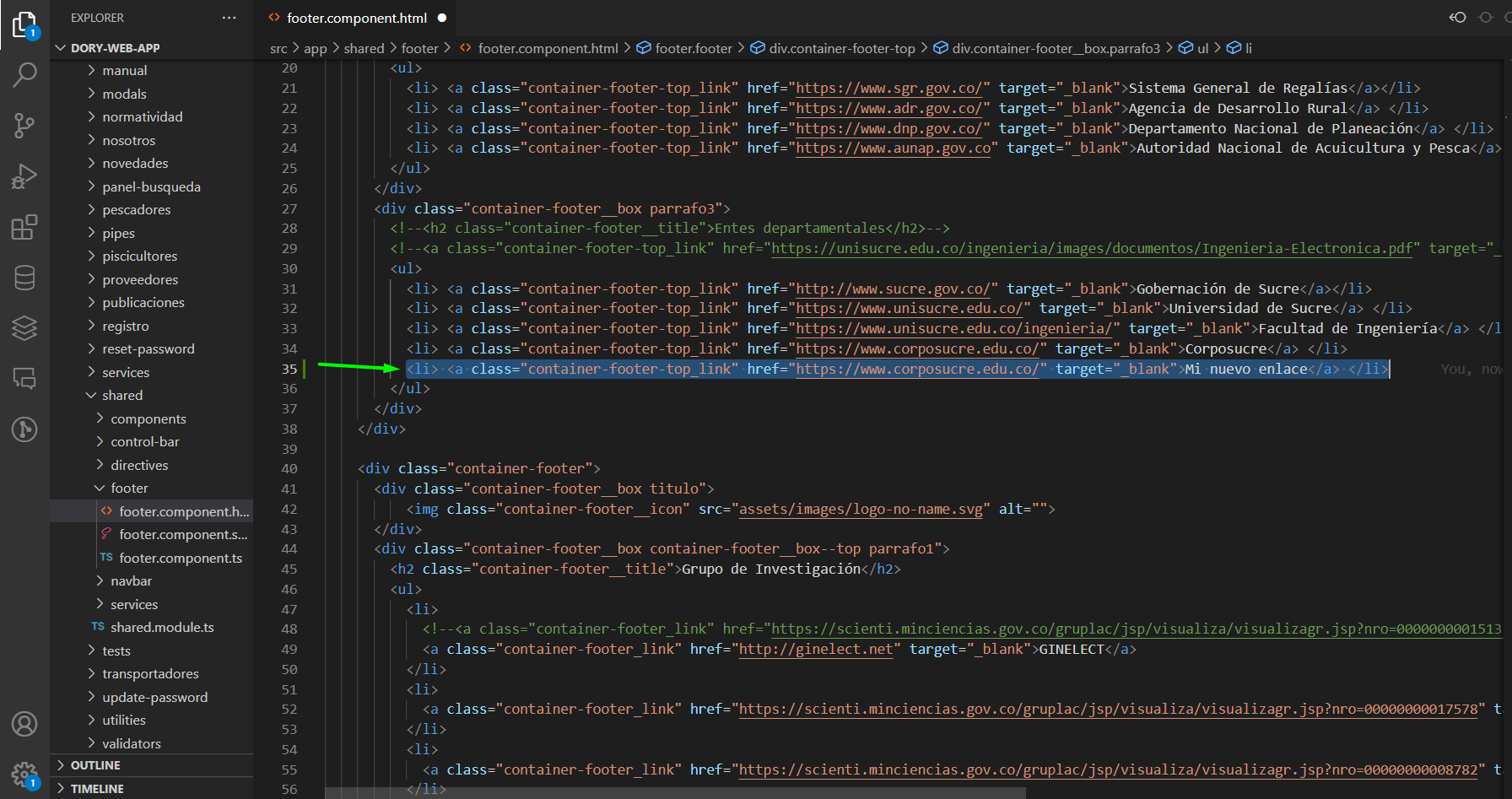
Supongamos que queremos agregar una opción en el **Footer** donde indica la flecha



Lo que debemos hacer es entrar a la carpeta /shared/footer

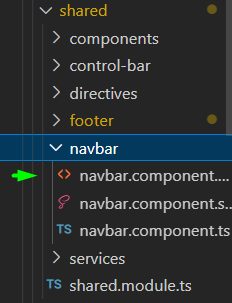


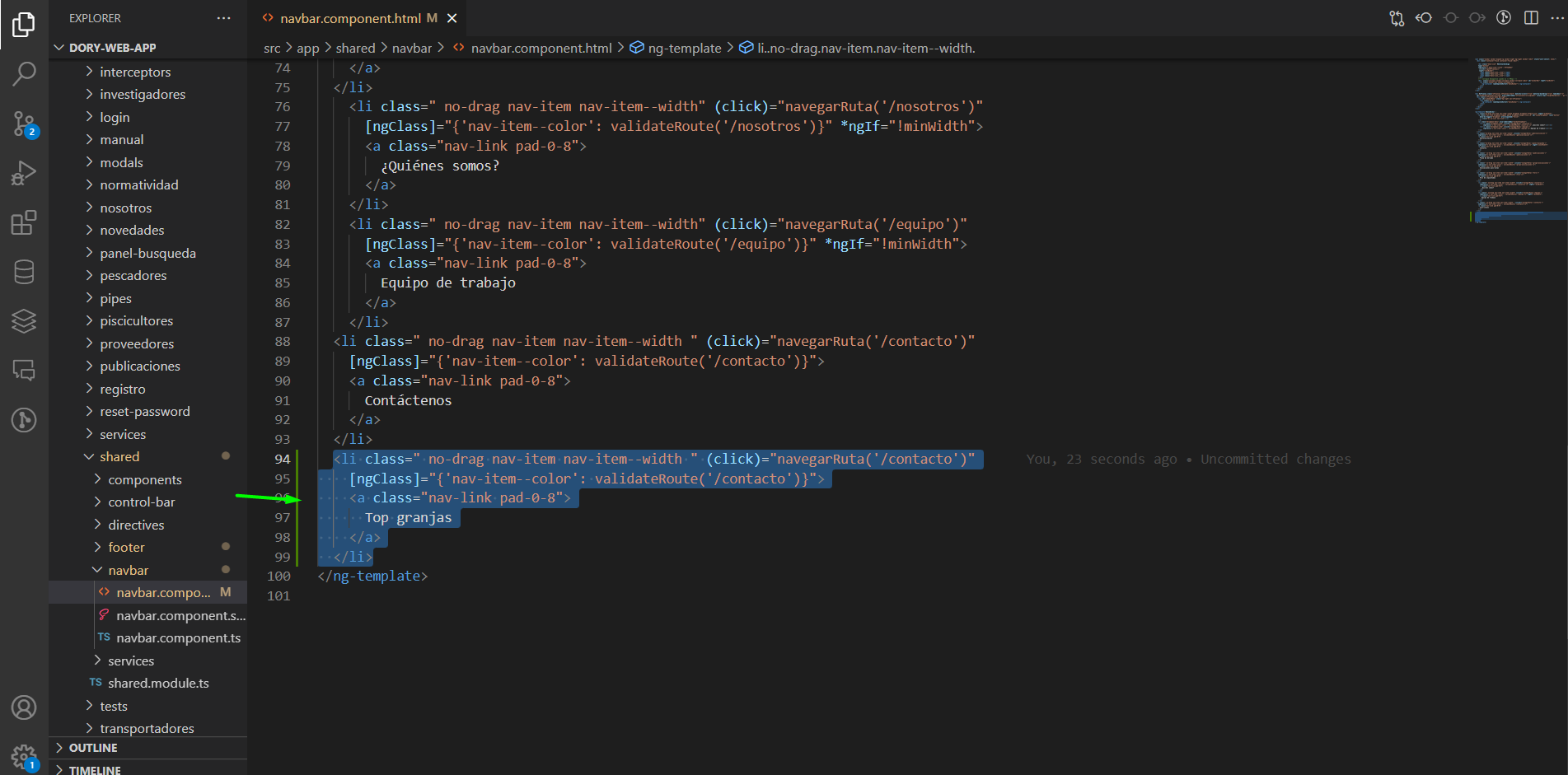
Y abrir el archivo footer.component.html y añadir un nuevo tag **<li></li>** con el nombre del enlace.

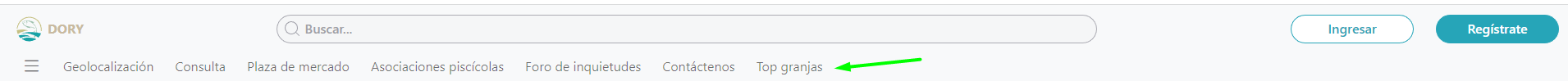




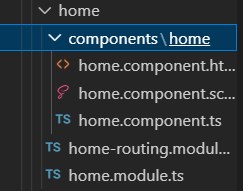
Ahora si queremos adicionar una nueva opción en el **Navbar** llamada “Top granjas” en la cual que mostrarán las 10 granjas mejor calificadas por los usuarios, debemos ir al archivo navbar.component.html

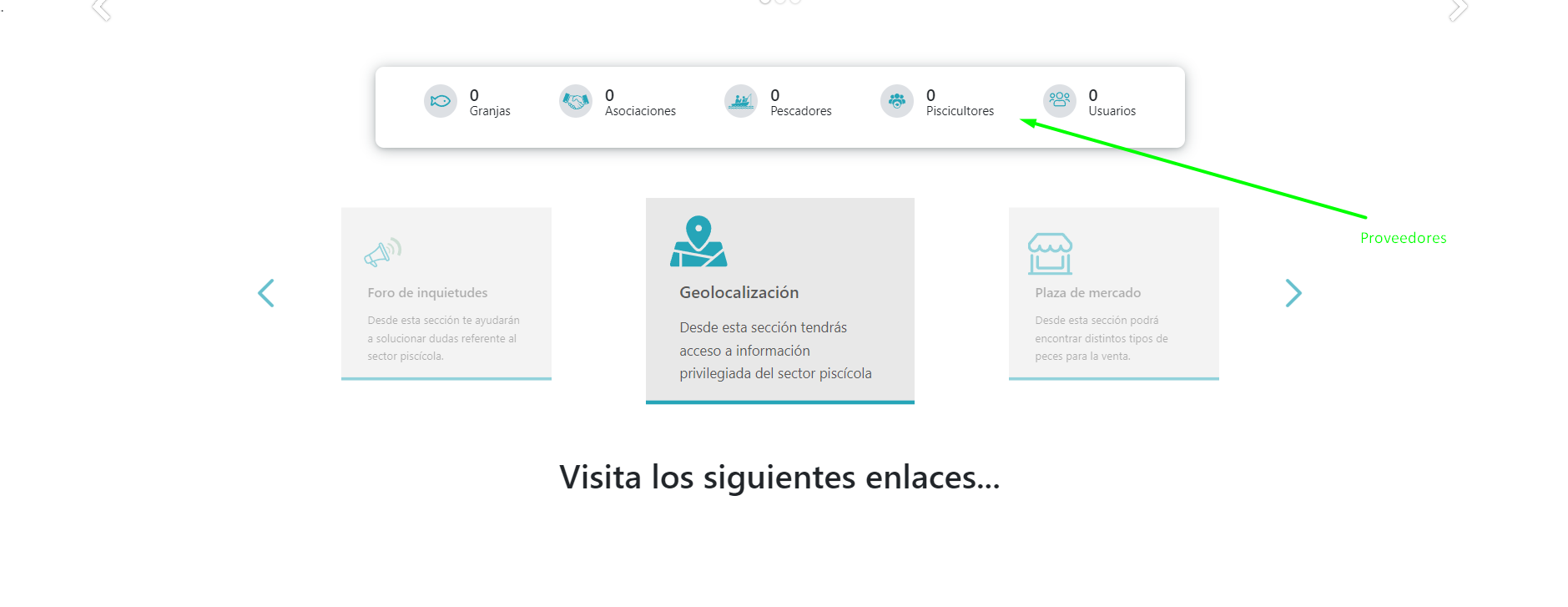






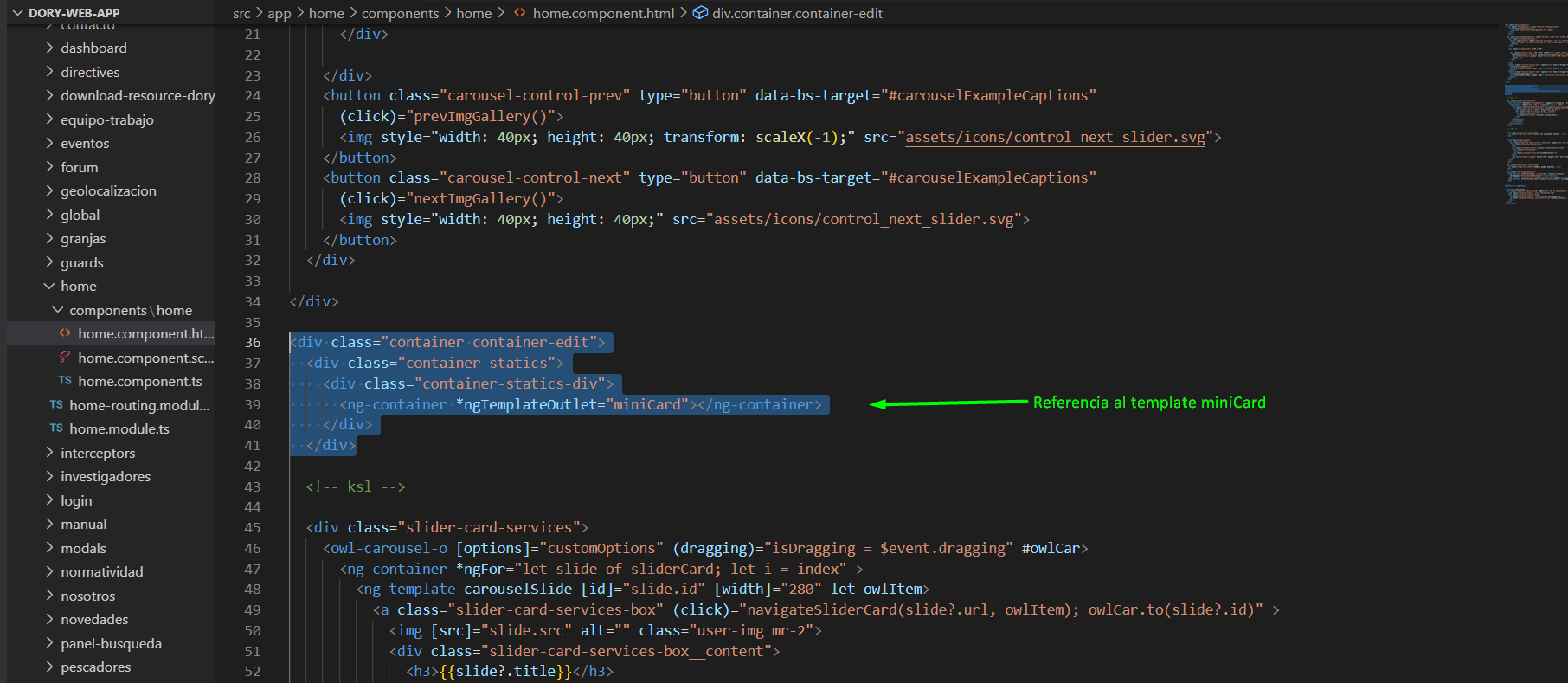
No existe diferencia entre modificar un componente intercambiable y uno fijo. Modifiquemos el componente intercambiable **HomeComponent** este es el primero que se muestra cuando se ingresa por primera ves a la aplicación. Podemos encontrarlo en el codigo fuente en el modulo home, aquí:



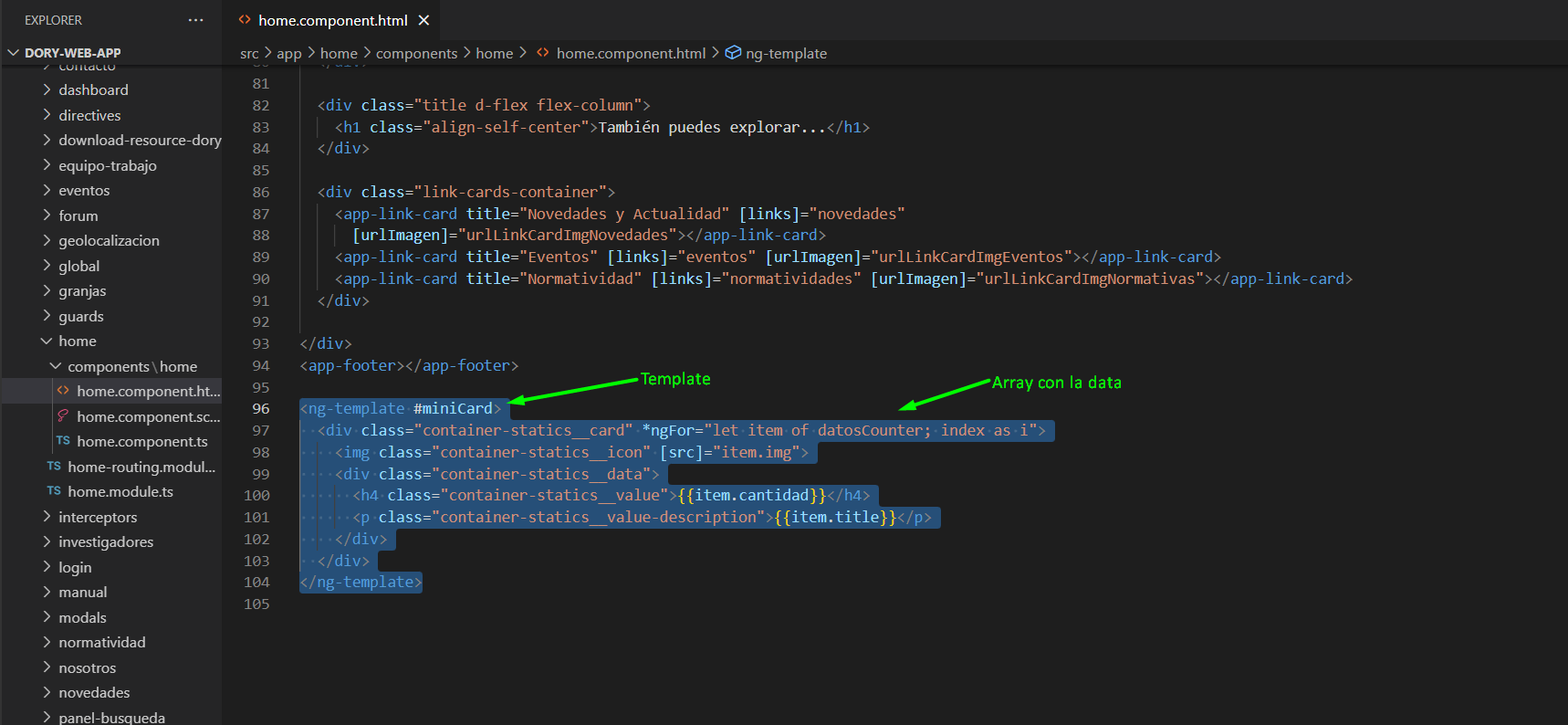
Supongamos que queremos agregar un nuevo indicador llamado **Proveedores** entre Piscicultores y Usuarios

Lo que debemos hacer es irnos a [home.component.html](http://home.component.html)

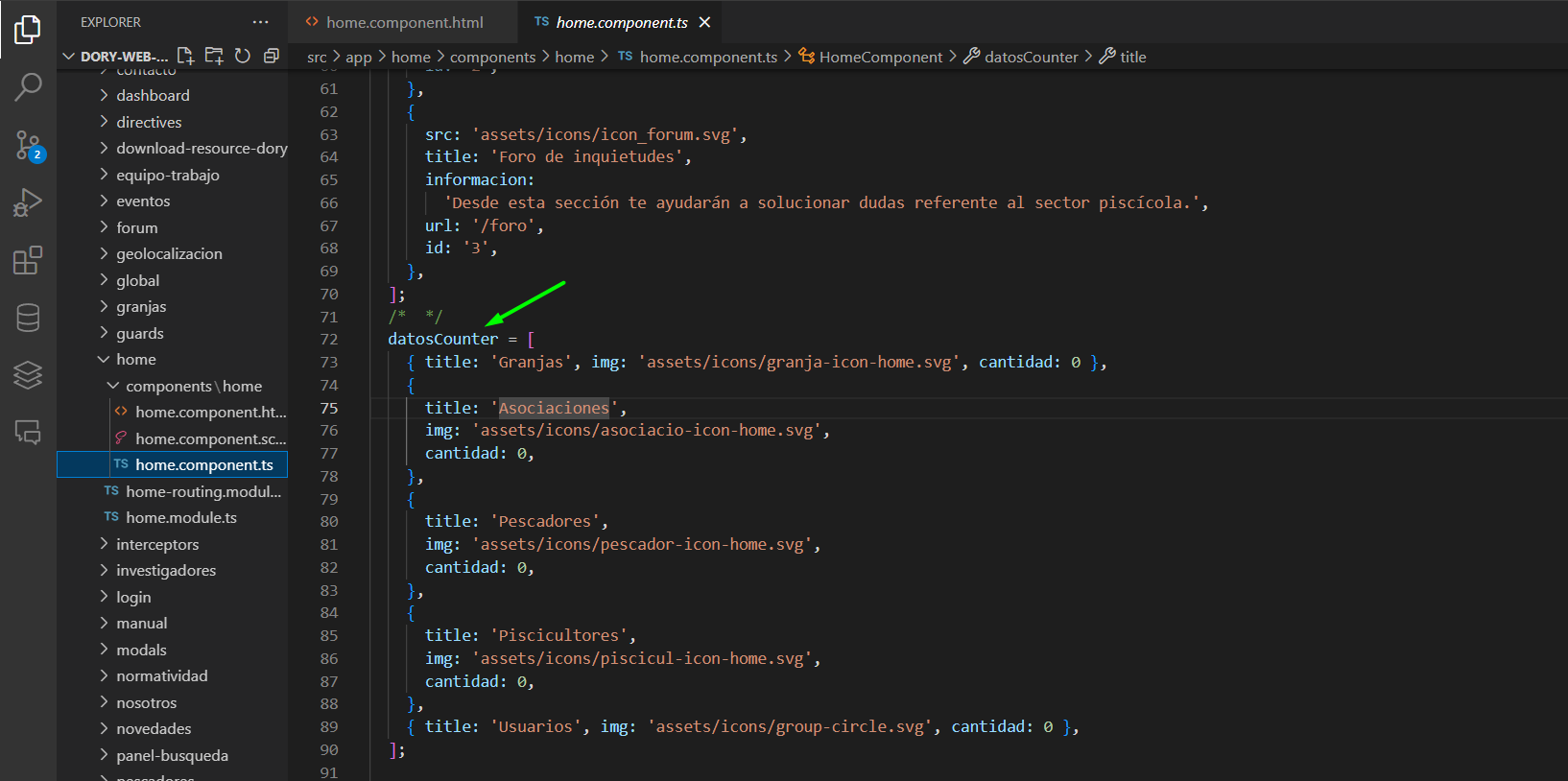
Y vemos que el siguiente es el codigo que pinta esa tarjeta y este hace referencia a un template



El template se encuentra unas líneas de código más abajo



También podemos observar que el template usa un array para obtener la data que va a pintar. Podemos observar el contenido de ese array en [home.component.ts](http://home.component.ts)



En cada posición guarda un indicador con su título, imagen y la cantidad inicializada en 0. Por lo tanto, si queremos agregar un nuevo indicador debemos agregar un nuevo objeto con los datos correspondientes.

El objeto que debemos agregar para el nuevo indicador de **Proveedores** es:

{

title:’Proveedores’,

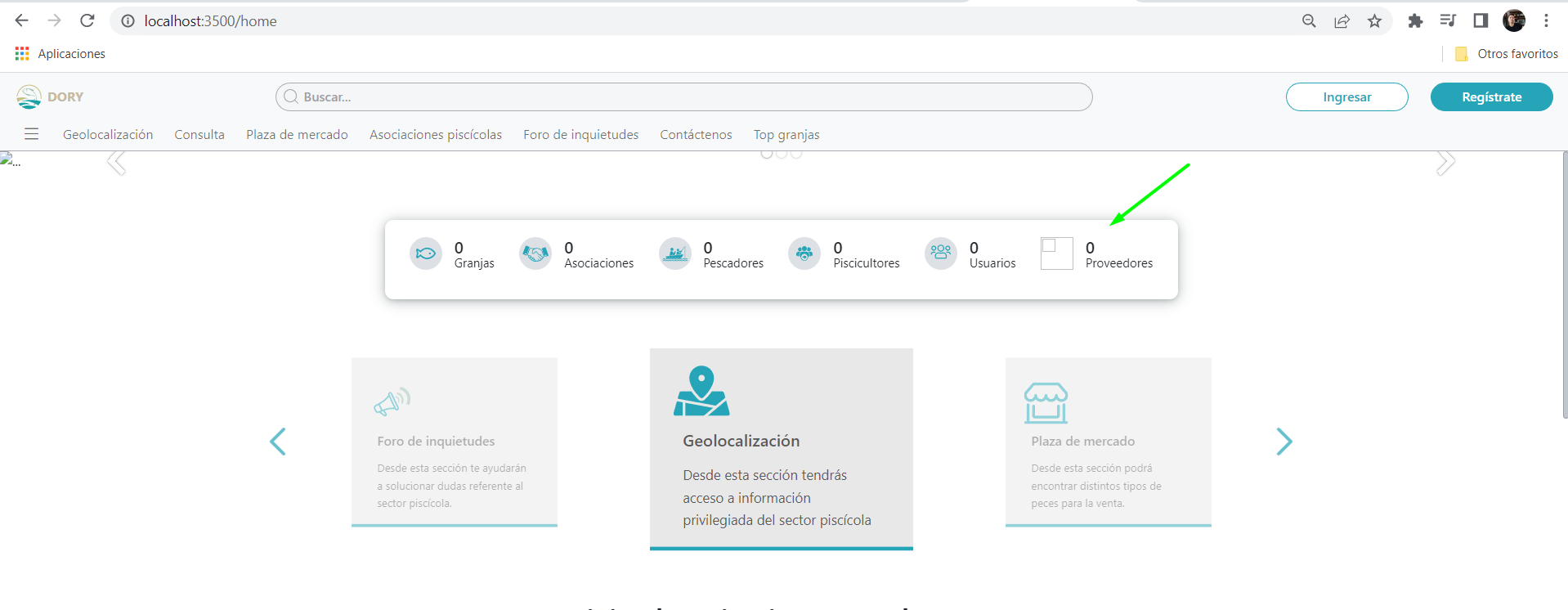
img:’’,

cantidad:0

}

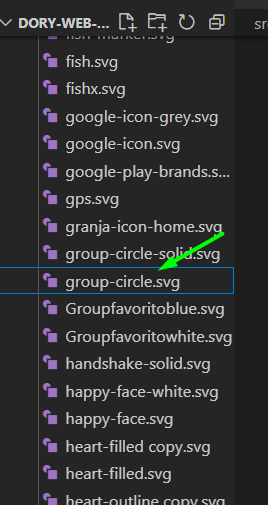


Como podemos verificar se ha adicionado el nuevo indicador en la tarjeta:

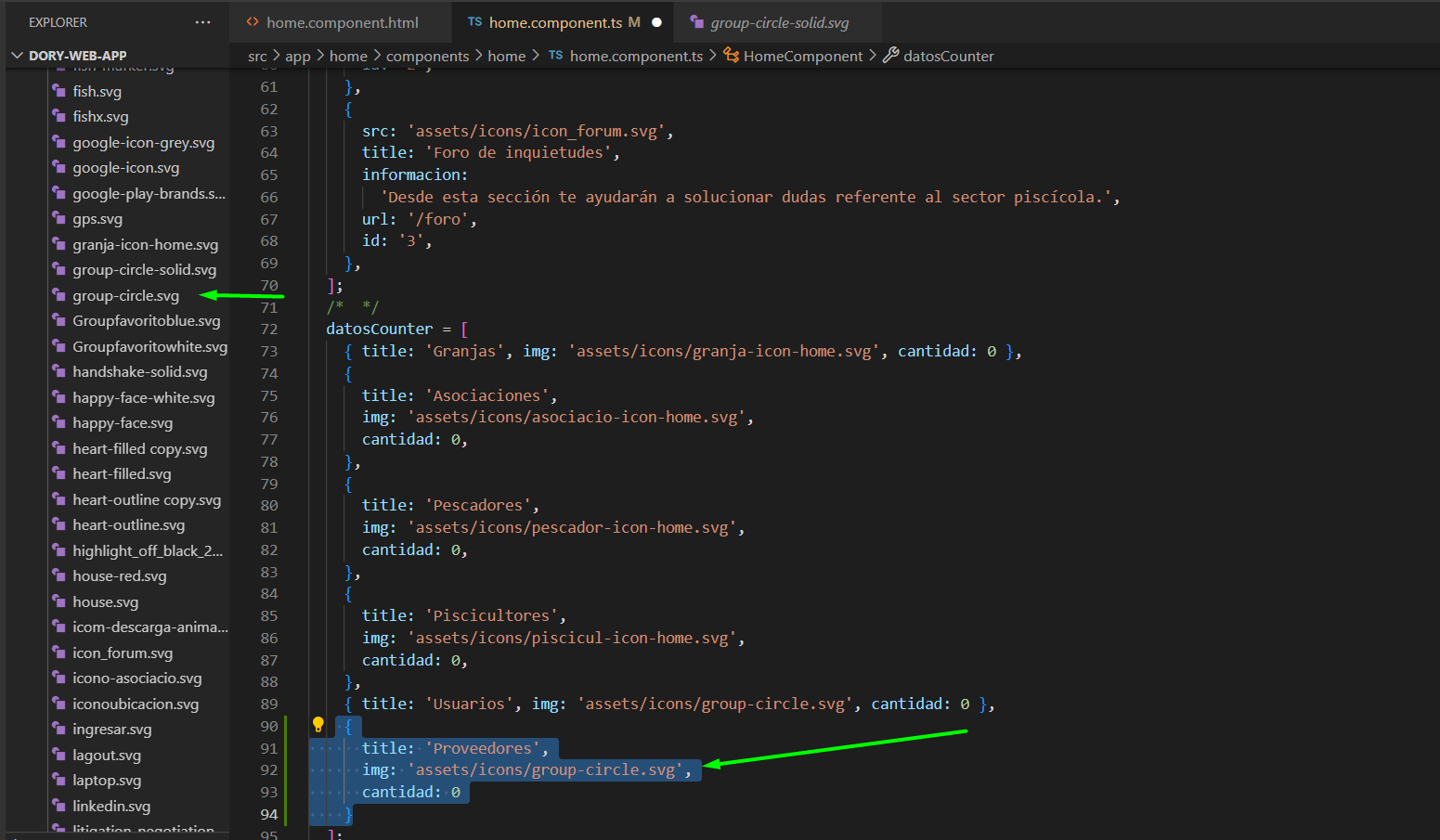


Pero aún nos falta el icono. Por lo tanto debemos descargar una imagen que represente a los proveedores y guardarla en el directorio /src/assets/icons/

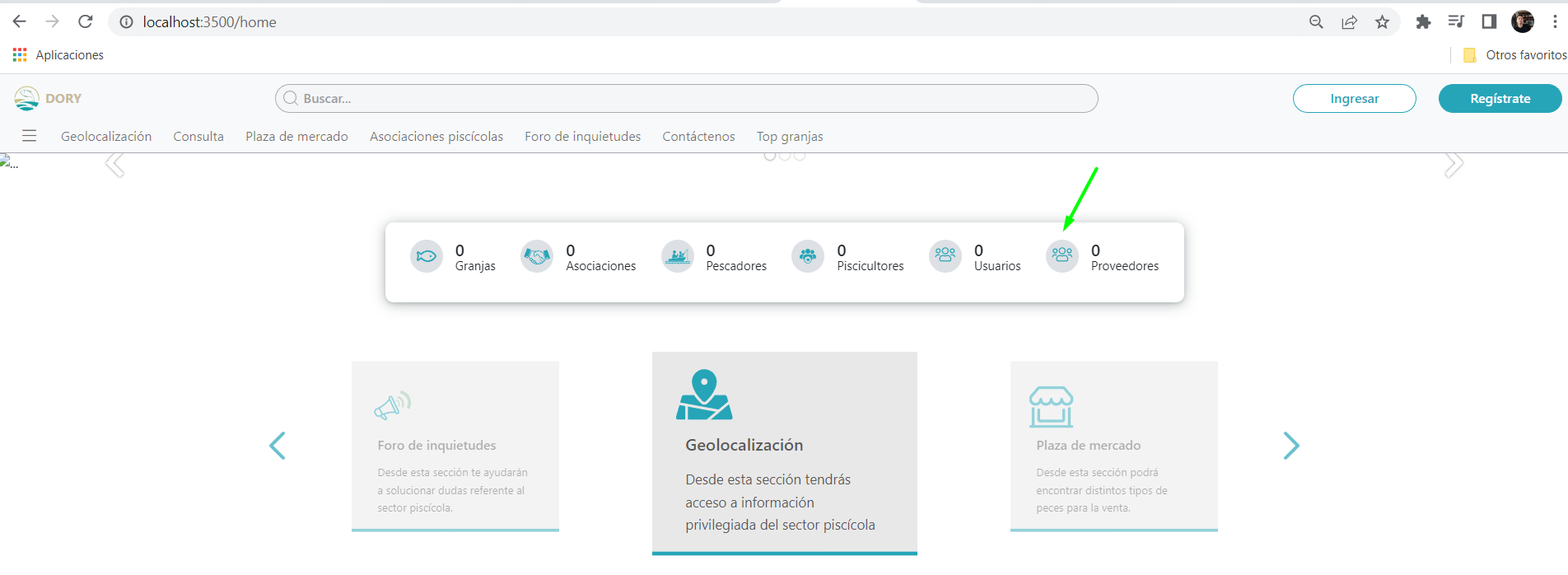
Para nuestro caso vamos a usar una que ya se encuentra allí



Modificamos el objeto y añadimos el path de esta imagen:



Aquí podemos ver los cambios reflejados en el servidor de desarrollo:



**Como desplegar cambios nuevos a Heroku**

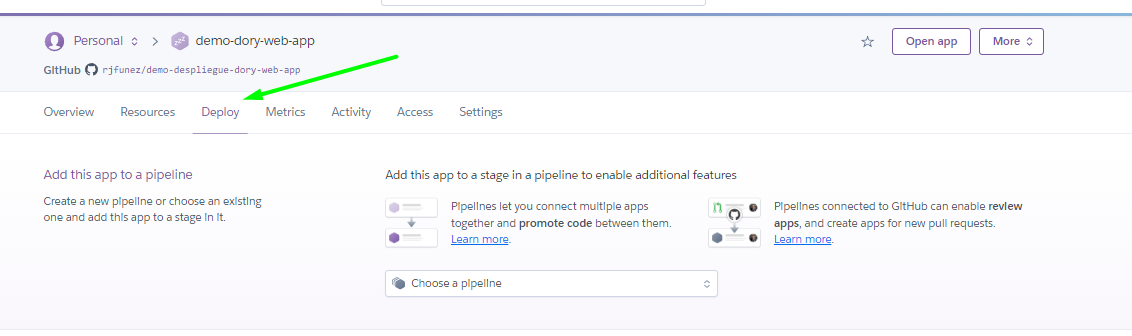
**Objetivo:** Aprender a desplegar nuevos cambios realizados al código fuente a Heroku

**Recursos necesarios:**

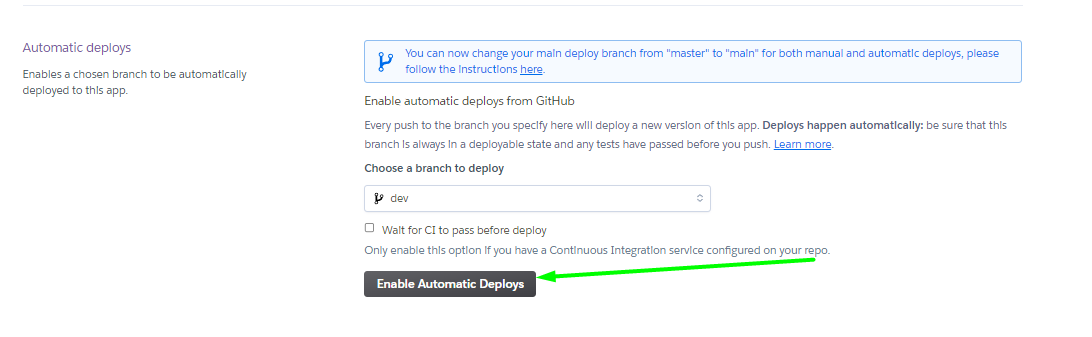
* Cuenta en Heroku con mínimo un dyno ECO

**Pasos:**

**1. Vamos al dashboard de Heroku e ingresamos en la pestaña Deploy**



**2. Habilitar los despliegues en la rama deseada.**

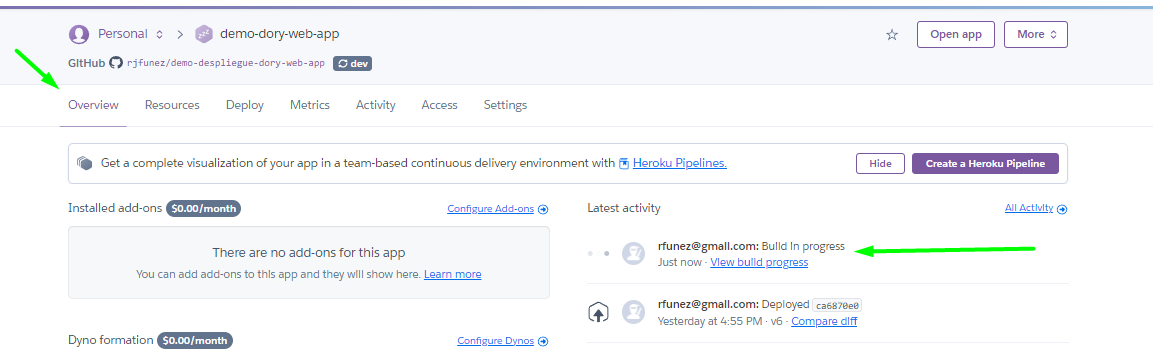
En nuestro caso desplegaremos cada vez que haya cambios en la rama dev. Hacerlo con la rama dev puede ser útil para un ambiente de pruebas. Para un ambiente productivo o de producción se recomienda hacerlo con la rama master.  


**3. Haz un commit:**  
git add .

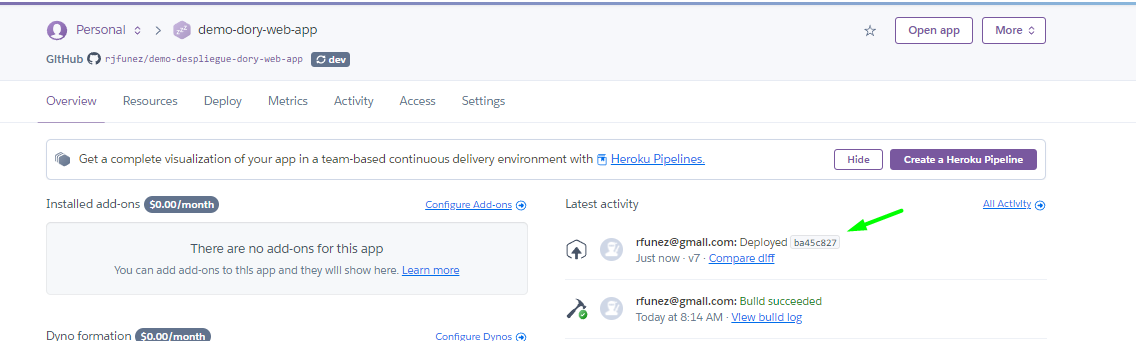
git commit -m "Nueva enlace en footer, nueva opción en navbar y nuevo indicado en home"

git push dory dev

Podemos ver en el dashboard de Heroku en la pestaña **Overview** que ya comenzó el despliegue.

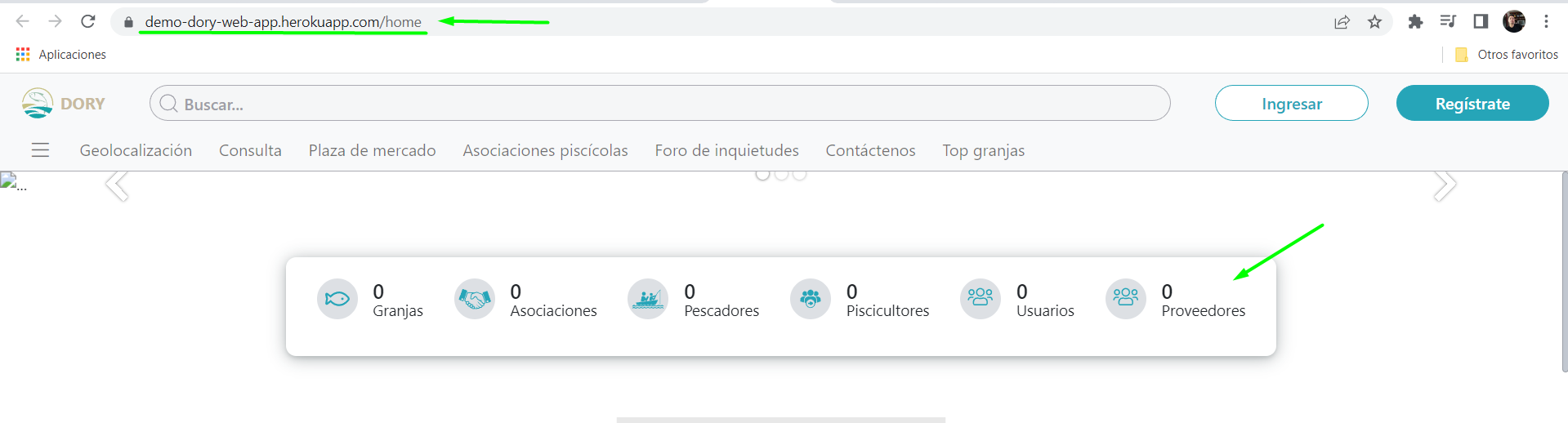


Despliegue finalizado:



Verificar:

Ingresamos al url donde Heroku desplego la aplicación



Podemos ver que ya se encuentra disponible el nuevo indicador de Proveedores en Heroku.

**Agregando una nueva opción de menú en el navbar para consultar las granjas mejor puntuadas**

Si queremos por ejemplo agregar una opción en el **Navbar** llamada …

**Agregando un nuevo servicio**

**Agregando/creando un nuevo módulo**

**Como cambiar globalmente los colores de Dory**

**Como cambiar la fuente primaria**

**Como Crear un proyecto en Firebase y generar el objeto “firebaseConfig” para configurar las variables de entorno en Heroku**