

Ingeniería Inversa

Desarrollo de Aplicaciones Web II

Grado en Ingeniería Informática

Curso 2021-2022

01/05/2022



Natanael Martínez Morales

ÍNDICE

Introducción	2
Desarrollo	3
Estructura del directorio app/components	3
Estructura del directorio app/guards-auth	5
Estructura del directorio app/modules	5
Estructura del directorio app/services	9
Conclusión	10
Referencias	11

1. Introducción

El objetivo principal de esta actividad es llevar a cabo el proceso de ingeniería inversa sobre una aplicación proporcionada, elaborando un informe que detalle el proceso realizado.

Este informe engloba aspectos como la estructura de la aplicación estudiada, en concreto la del directorio fuente, así como los elementos que forman esta aplicación, como los componentes, servicios, entre otros.

Una vez realizado el informe, se ha llevado a cabo la implementación de una aplicación que gestione la plantilla de un equipo deportivo utilizando como base la aplicación dada, de forma que el punto de partida sea esta "plantilla" con el objetivo de que el estudiante se desenvuelva en entornos donde tenga que realizar un estudio previo al existir trabajo previo en el que no ha participado y necesite comprender para poder llevar a cabo sus funciones.

2. Desarrollo

Para poder comenzar con el desarrollo del informe, en primer lugar se ha integrado el directorio *src* proporcionado en una nueva aplicación creada. Una vez integrado, se han tenido que instalar paquetes como *Bootstrap* y *FontAwesome* ya que sin ellos no sería posible la ejecución de la aplicación.

Dentro del directorio *src*, nos centraremos en la estructura y los elementos que forman el directorio *app*.

El directorio *app* contiene el fichero de *routing* que contiene las rutas a los distintos componentes del sitio web y los ficheros base del componente **app**. Además, contiene una serie de directorios explicados a continuación:

Estructura del directorio *app/components*

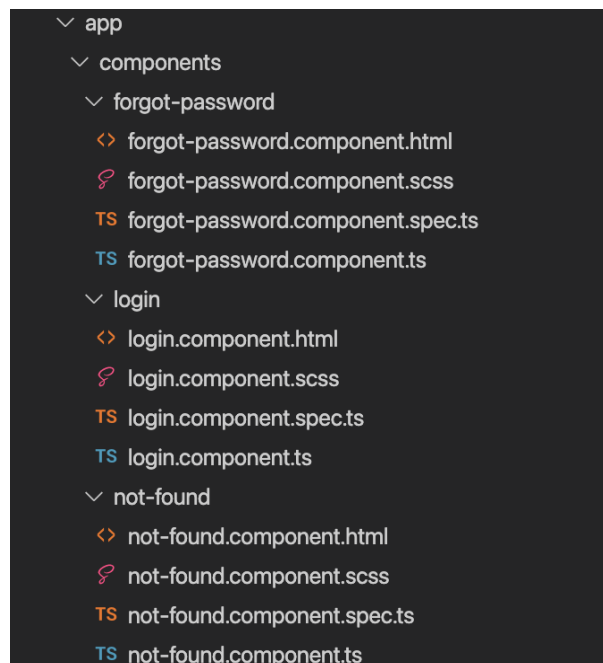
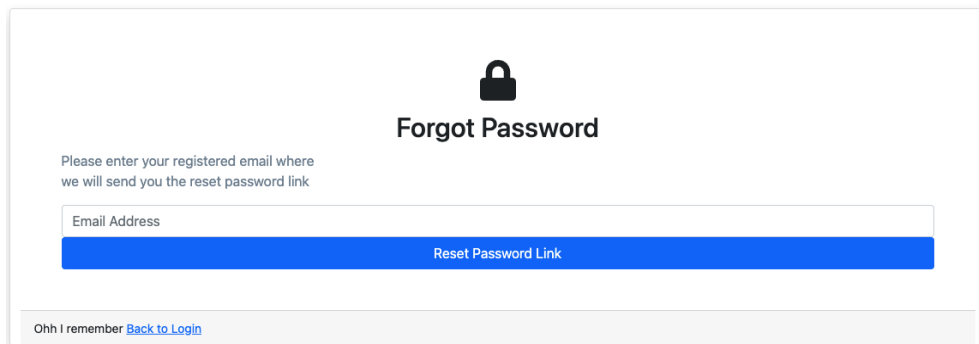


Figura 1: estructura del directorio *app/components*

Como podemos observar, existen tres componentes dentro del directorio *app/components* que se corresponden con el **forgot-password**, **login** y **not-found**.

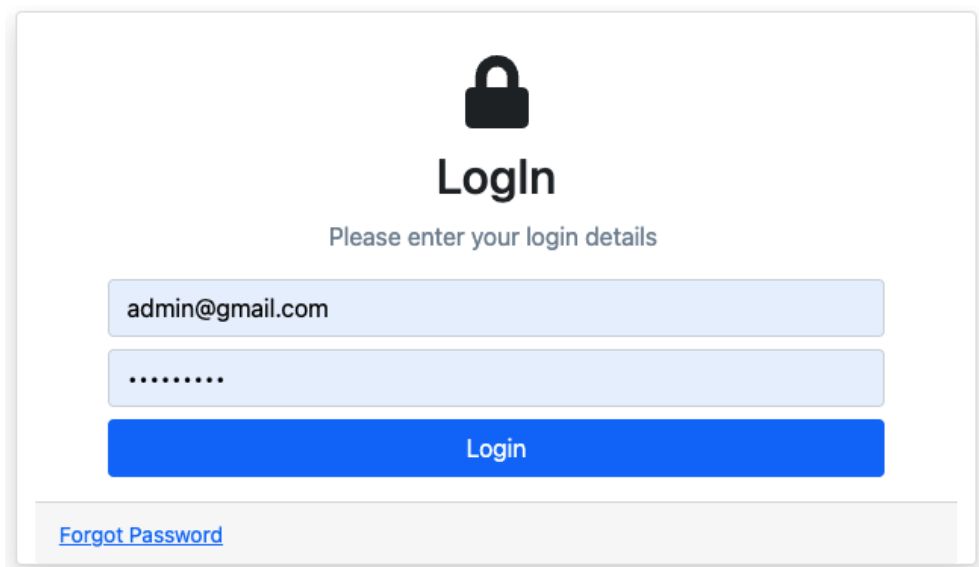
El componente **forgot-password** tiene el objetivo de permitir al usuario recuperar la contraseña en caso de pérdida. Para ello, muestra un formulario que permite introducir el correo electrónico del usuario para poder restablecerla (figura 2).



The image shows a 'Forgot Password' form. At the top, there is a black padlock icon. Below it, the title 'Forgot Password' is centered. A message reads: 'Please enter your registered email where we will send you the reset password link'. There is a text input field labeled 'Email Address'. Below the input field is a blue button labeled 'Reset Password Link'. At the bottom, there is a link that says 'Ohh I remember [Back to Login](#)'.

Figura 2: componente **forgot-password**

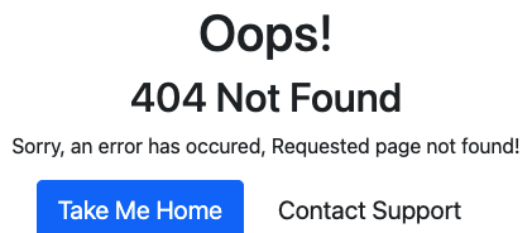
El componente **login** (figura 3) es aquel que permite que los usuarios puedan acceder a la aplicación haciendo uso de sus credenciales. Además, contiene el enlace correspondiente al componente **forgot-password** descrito anteriormente.



The image shows a 'Login' form. At the top, there is a black padlock icon. Below it, the title 'Login' is centered. A message reads: 'Please enter your login details'. There are two text input fields: the first contains 'admin@gmail.com' and the second contains '.....'. Below the input fields is a blue button labeled 'Login'. At the bottom, there is a link that says '[Forgot Password](#)'.

Figura 3: componente **login**

Finalmente, el último componente se llama **not-found** y su objetivo es mostrar un mensaje de error al usuario cuando trata de acceder a una ruta que no esté registrada, ofreciendo la posibilidad de volver a la página principal (figura 4).



The image shows a '404 Not Found' error page. At the top, the text 'Oops!' is displayed in a large, bold font. Below it, the text '404 Not Found' is displayed in a large, bold font. A message reads: 'Sorry, an error has occurred, Requested page not found!'. There are two buttons: a blue button labeled 'Take Me Home' and a text link labeled 'Contact Support'.

Figura 4: componente **not-found**

Estructura del directorio *app/guards-auth*

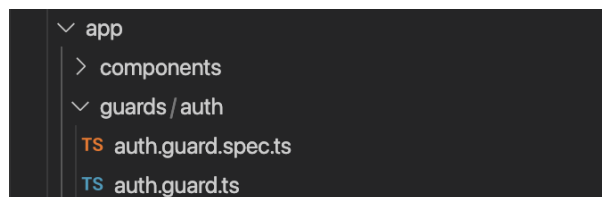


Figura 5: estructura del directorio *app/guards-auth*

Este directorio se corresponde con un **guard**, que básicamente es una interfaz que permite proteger rutas de forma que solo puedan acceder aquellos usuarios que contengan permisos para ello, por lo que si el usuario no dispone de los privilegios necesarios, se actúa en consecuencia, redireccionándolos a otra página o realizando las acciones especificadas en el fichero.

Estructura del directorio *app/modules*

Este directorio se subdivide en dos grandes directorios: **admin** y **user**. El directorio **admin** contiene los componentes y ficheros de routing y módulo correspondientes a los usuarios con privilegios de administrador y el directorio **user** contiene los ficheros de módulo y routing correspondientes al resto de usuarios.

```

  app
  ├── components
  ├── guards
  └── modules
      ├── admin
      │   ├── components
      │   │   ├── about
      │   │   │   ├── about.component.html
      │   │   │   ├── about.component.scss
      │   │   │   ├── about.component.spec.ts
      │   │   │   └── about.component.ts
      │   │   ├── admin-dashboard
      │   │   │   ├── admin-dashboard.component.html
      │   │   │   ├── admin-dashboard.component.scss
      │   │   │   ├── admin-dashboard.component.spec.ts
      │   │   │   └── admin-dashboard.component.ts
      │   │   ├── contact
      │   │   │   ├── contact.component.html
      │   │   │   ├── contact.component.scss
      │   │   │   ├── contact.component.spec.ts
      │   │   │   └── contact.component.ts
      │   │   ├── footer
      │   │   │   ├── footer.component.html
      │   │   │   ├── footer.component.scss
      │   │   │   ├── footer.component.spec.ts
      │   │   │   └── footer.component.ts
      │   │   ├── header
      │   │   │   ├── header.component.html
      │   │   │   ├── header.component.scss
      │   │   │   ├── header.component.spec.ts
      │   │   │   └── header.component.ts
      │   │   ├── home
      │   │   │   ├── home.component.html
      │   │   │   ├── home.component.scss
      │   │   │   ├── home.component.spec.ts
      │   │   │   └── home.component.ts
      │   │   └── services
      │   │       ├── services.component.html
      │   │       ├── services.component.scss
      │   │       ├── services.component.spec.ts
      │   │       └── services.component.ts
      │   ├── admin-routing.module.ts
      │   ├── admin.module.ts
      └── user
          ├── user-routing.module.ts
          └── user.module.ts

```

Figura 6: estructura del directorio *app/modules*

El directorio ***admin/components*** contiene los componentes accesibles a los usuarios administradores, que son:

- **about:** Contiene elementos de ejemplo pero el objetivo principal sería la de ofrecer información sobre la empresa, como la típica página 'Sobre nosotros'. Se ubica en la ruta */admin/about*.

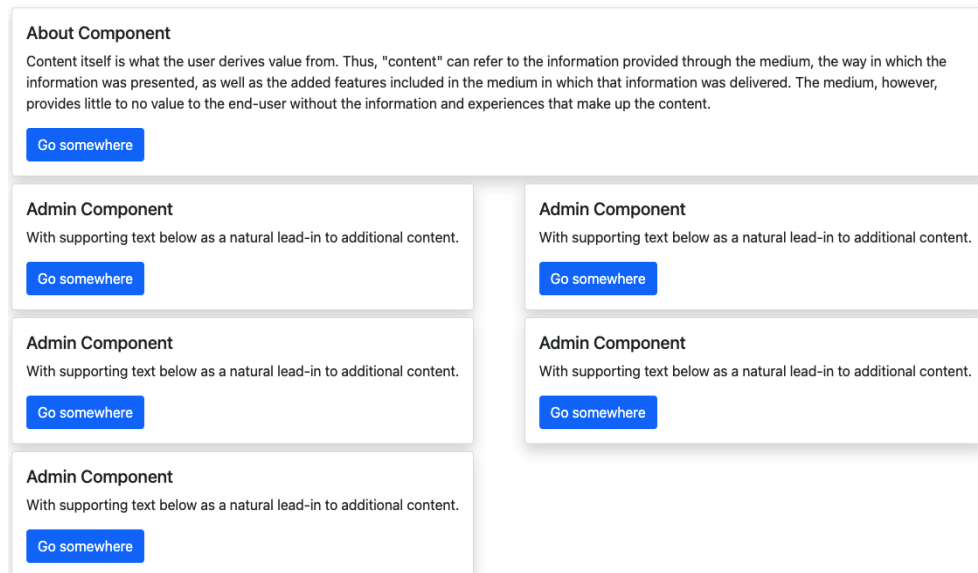


Figura 7: componente **about**

- **admin-dashboard:** Constituye el componente principal de esta aplicación bajo la ruta */admin/home* y contiene los componentes **header** y **footer**.

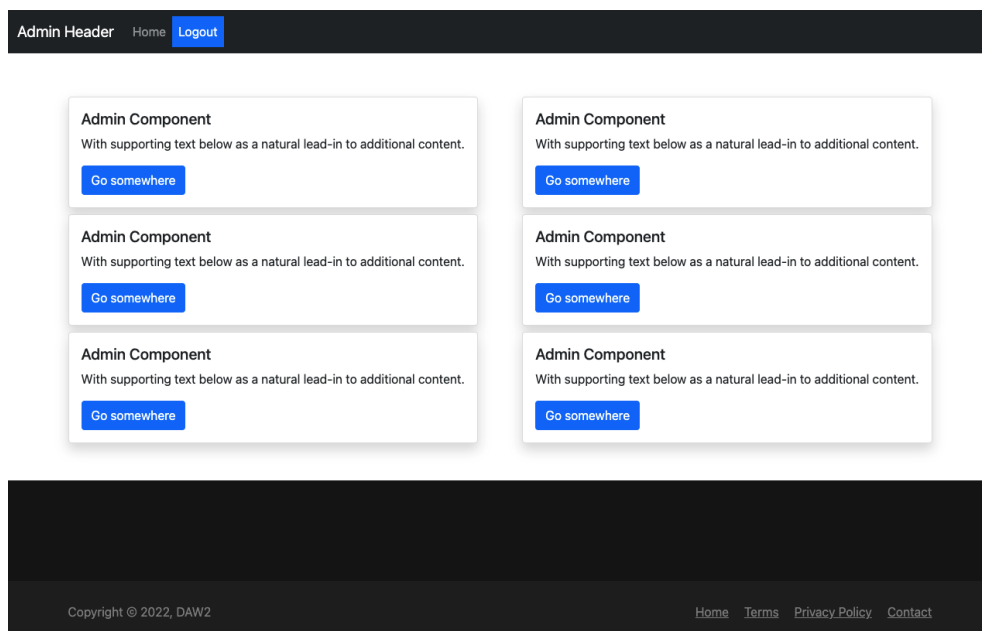


Figura 8: componente **admin-dashboard**

- **contact:** Es un componente cuyo objetivo es ofrecer los métodos de contacto establecidos. Se encuentra en la ruta */admin/contact*.

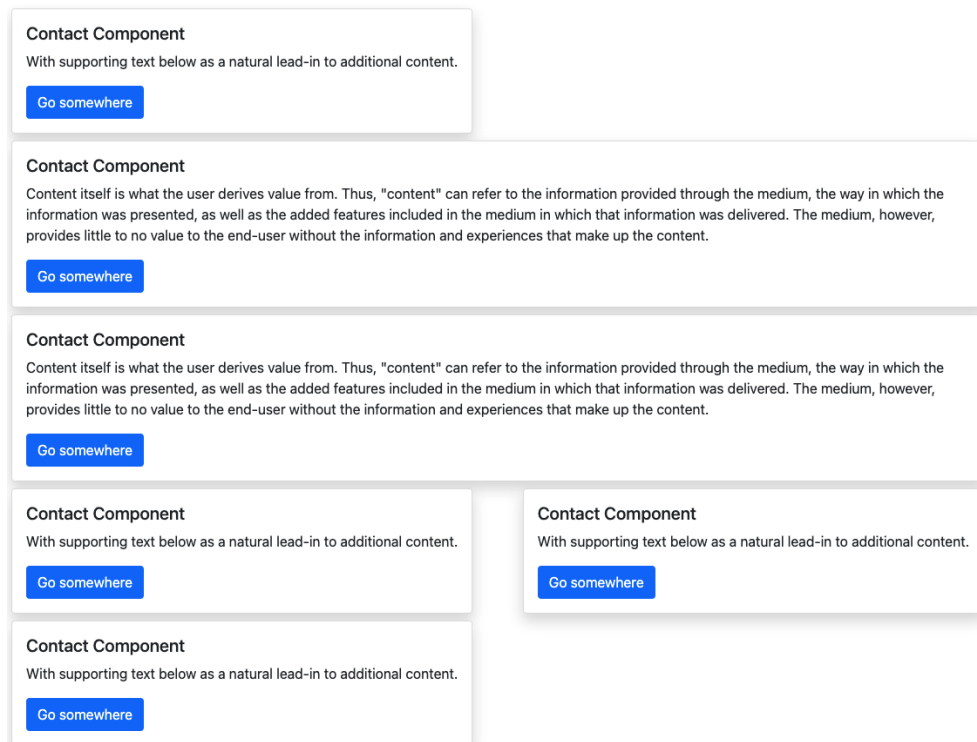


Figura 9: componente **contact**

- **footer:** Es el componente correspondiente al footer del sitio web.



Figura 10: componente **footer**

- **header:** Es el componente correspondiente al header del sitio web y que contiene la barra de navegación.



Figura 11: componente **header**

- **home:** Componente que contiene el contenido de **admin-dashboard** descrito anteriormente.
- **services:** Constituye el componente dedicado a mostrar los servicios a los que está dedicado el sitio web. Se ubica en la ruta `/admin/services`.

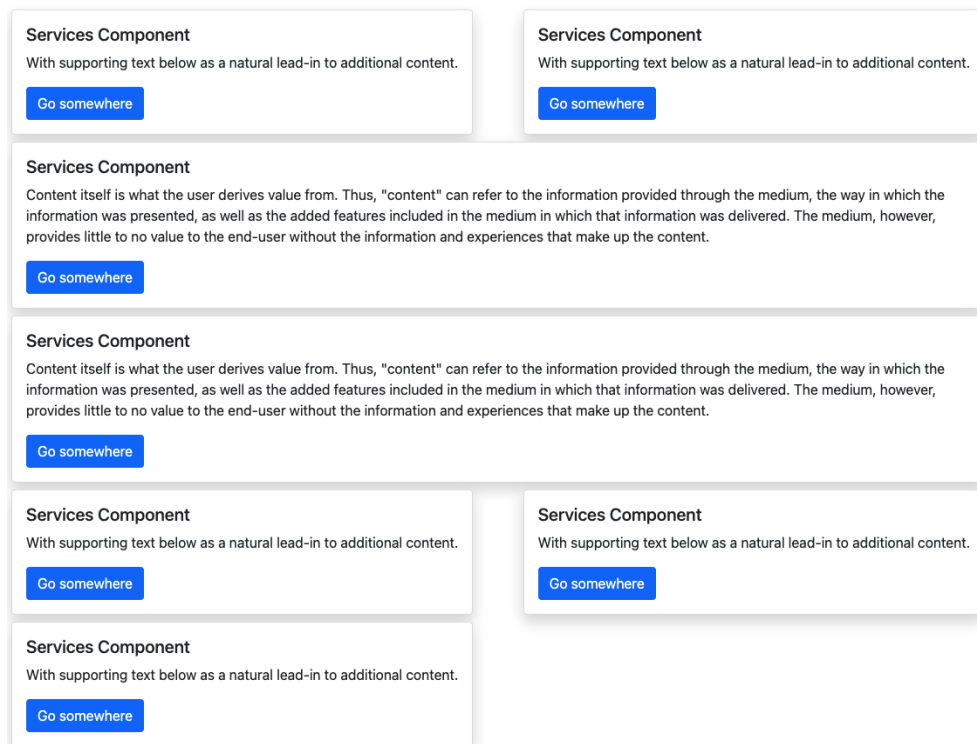


Figura 12: componente **services**

Estructura del directorio *app/services*

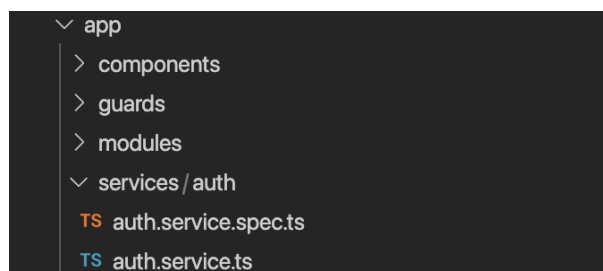


Figura 13: estructura del directorio *app/services*

Bajo este directorio se encuentra el servicio dedicado a la autenticación de usuarios a través de un token, para lo que se han creado diferentes funciones.

3. Conclusión

Para concluir, se trata de una práctica realmente interesante y útil teniendo en cuenta que normalmente en el mundo laboral te encuentras ante proyectos que ya están parcialmente realizados, de forma que tienes que llevar a cabo un proceso de estudio del proyecto para poder obtener una visión global del mismo. Gracias a la ingeniería inversa, este proceso es mucho más sencillo y permite ofrecer una perspectiva del proyecto para poder comenzar con las tareas específicas que se hayan determinado. En este caso, gracias a la elaboración de este informe, me ha sido mucho más sencillo el desarrollo de una aplicación utilizando como base la aplicación dada, por lo que considero que esta actividad me ha brindado de una nueva forma de analizar un problema, ya que hasta el momento no me había planteado el hecho de tener que estudiar un proyecto parcialmente desarrollado previamente al propio desarrollo.

4. Referencias

[1] Saul Lopez (2019, junio 17). Proteger rutas con Guards en Angular. Accedido en: 2022, mayo 1. Disponible en: https://www.google.com/search?q=guard+en+angular&rlz=1C5CHFA_enES993ES993&oq=guard+en+&aqs=chrome.2.69i57j0i512l5j0i22i30j69i60.3040j1j7&sourceid=chrome&ie=UTF-8

[2] Recursos Ingeniería Inversa. Accedido en: 2022, mayo 1. Disponible en: https://aep22.ulpgc.es/pluginfile.php/1775843/mod_folder/content/0/src_routenavAngular_Inverse.zip?forcedownload=1