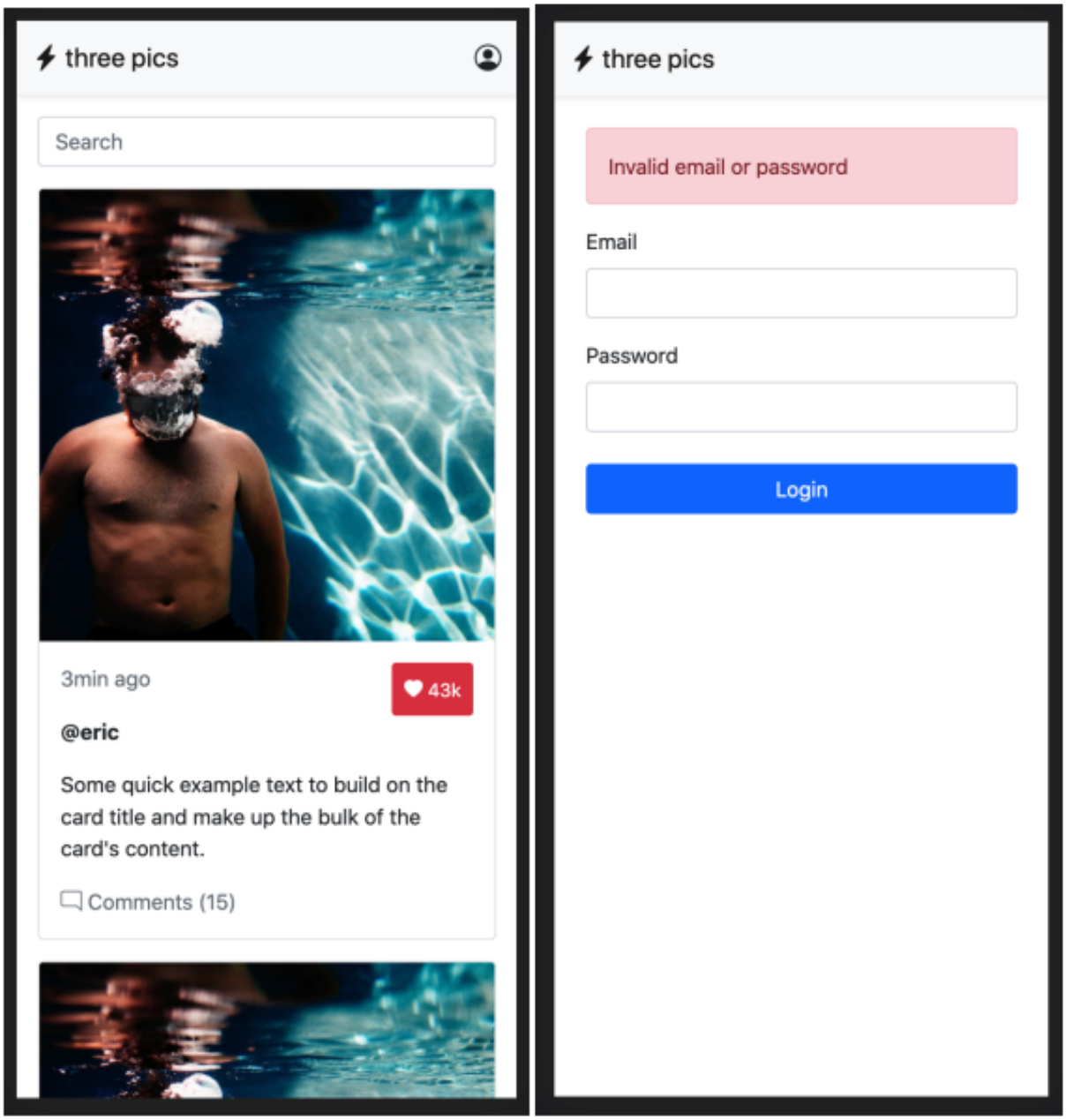
Julio García Pérez

Front End Frameworks

Ejercicio 4 – Aplicación React dinámica HTTP con React Router y Hooks

Descripción

El objetivo del ejercicio es la construcción de una aplicación web React.js con diferentes components que tenga como resultado la siguiente interfaz de usuario (Mobile first):



En esta ocasión, todos los datos (posts, perfil de usuario) deberán obtenerse por HTTP desde el API expuesto en https://thre-points.herokuapp.com/api . Junto al enunciado del ejercicio puede encontrarse una colección Postman como documentación sobre los diferentes endpoints.

Todos los componentes React deberán ser funcionales, haciendo uso de los hooks useState y useEffect para manejar el estado y ciclo de vida del componente.

Además, las diferentes secciones se mostrarán bajo una ruta del navegador, utilizando la biblioteca React Router:

Ruta	Componente	Requisito
/	PostList	Usuario previamente autenticado
/login	Login	
/profile	Profile	Usuario previamente autenticado

La aplicación contará con autenticación HTTP basada en Bearer token.

Si cualquier petición HTTP devuelve código 401, deberá borrarse el localStorage del navegador, limpiar el estado *currentUser* y redirigir al usuario a la ruta /login

- Al escribir sobre la barra de búsqueda se deberán mostrar solo los *posts* que contengan la cadena de búsqueda en su contenido
- Los *posts* deberán mostrarse después de haber cargado la interfaz. Mostrando el texto "Loading…" durante el tiempo de carga.
- El botón "Me Gusta" deberá incrementar el contador de *likes* sobre el *post* utilizando el API HTTP
- Al pulsar sobre el icono de perfil el usuario será redirigido a la ruta /profile

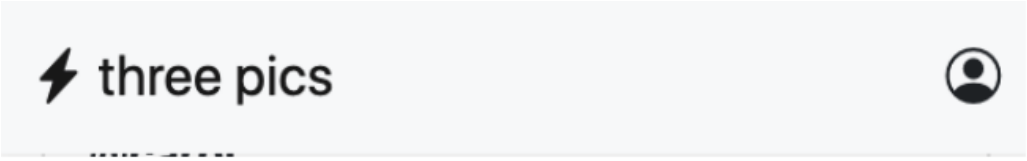
@alex

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed maximus fermentum mi, vitae varius quam. Sed fringilla dignissim sollicitudin.

- Se incluirá un botón "Salir" en la página de perfil. Al hacer click sobre él se cerrará la sesión del usuario eliminando el token de acceso.
- Al pulsar sobre el logo "Three pics" se mostrará el listado de posts

Se deberán implementar al menos los siguientes componentes, utilizando las props y estados que se indican

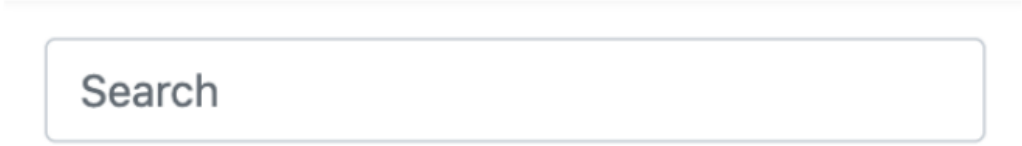
NavBar



Props:

- onLogoClick (function)
- onProfileClick (function)

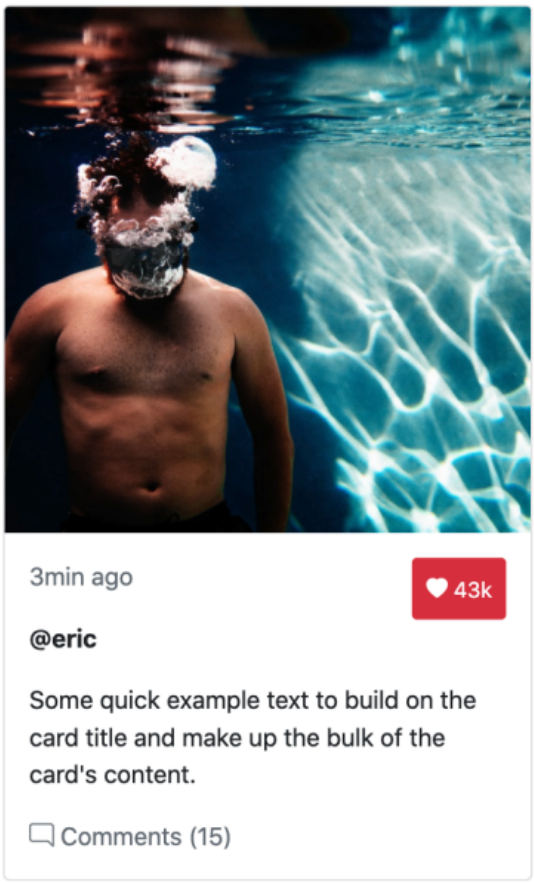
SearchBar



Props:

- value (string)
- onSearch (function)

Post



Props:

- createdAt (Date)
- autor (string)
- text (string)
- comments (number)
- image (string)

State:

- likes (number)

PostList

Contenedor de components “Post”

Props:

- posts (object array)

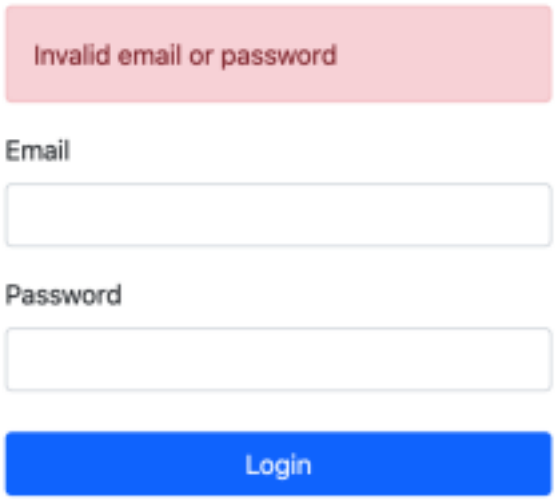
App

State:
- search (string)
- posts (object array)
- section (string)
- currentUser (object)

Profile

props:
- avatar (string)
- username (string)
- bio (string)
- onLogout (function)

Login



props:
- onLoginComplete (function)

state:
- error (boolean)

Al pulsar sobre “login” deberán enviarse los datos del formulario con una petición POST a la siguiente url:
https://three-points.herokuapp.com/api/login

Body: { “username”: “…”, “password”: “…” }
Content-Type: application/json

Usuario registrado para pruebas:
{ “username”: “john”, “password”: “P4ssW0rd!#” }

Si la petición HTTP devuelve un código 200 (éxito), se guardará el token del usuario en el localStorage del navegador y se mostrará el listado de posts. En caso contrario, se activará el estado de error y se mostrará el mensaje de error en pantalla.

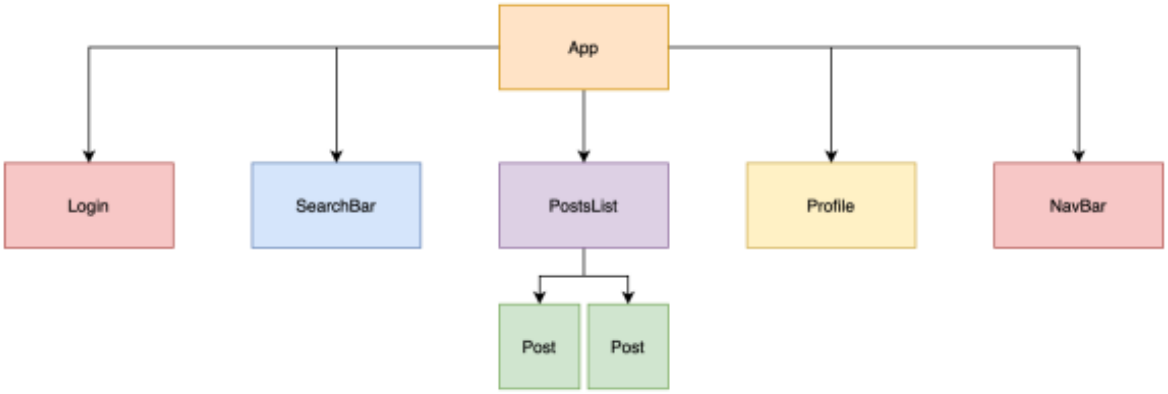
El resto de peticiones HTTP deberán usar la cabecera "Authorization: Bearer {token}" para resolver la autenticación.

Cada vez que se inicie la aplicación web se deberá verificar si existe el token del usuario en el localStorage del navegador. Si existe, se accederá al listado de posts. En caso contrario, se mostrará la pantalla de login.

Junto al enunciado del ejercicio podrá encontrarse una colección Postman para estudiar el API expuesto en https://three-points.herokuapp.com/api

--

El Proyecto deberá seguir la siguiente jerarquía de components:



Se recomienda el uso de *bootstrap* biblioteca para CSS e iconos.

Se recomienda el uso de *axios* como biblioteca JavaScript para cliente HTTP

Entrega

Se deberá entrgar un enlace a repositorio git público adjuntando el identificador del commit. La entrega **no debe contener** el directorio node_modules

Se deberá adjuntar un video de corta duración con una grabación de pantalla donde el alumno haga una breve **demostración** de la aplicación resultante sobre el navegador, explicando además el código desarrollado sobre el editor de código, poniendo especial foco en las novedades con respecto al ejercicio anterior. En el video se deberá ver cómo las diferentes peticiones HTTP XHR aparecen en la pestaña "Network" del navegador conforme se interactua con la web, mostrando los datos enviados y/o recibidos.

Se valorará positivamente el estilo y formato del código. Se recomienda utilizar *linters* de código javascript como jshint.

Al instalar (npm install / yarn) y ejecutar (npm install / yarn start) el proyecto React se espera visualizar en el navegador (http://localhost:3000) la aplicación web propuesta.