

README:

Para la ejecución del parcial, simplemente se clona el proyecto, se realiza npm install y se ejecuta con npm start. Una vez se ejecuta, aparecerá el componente login que permite verificar el correo y la contraseña como indica el enunciado, al seleccionar o clickear el botón siguiente, será redirigido a la vista /parts y se imprimirá en consola el objeto json que representa el schema de datos del enunciado. En la vista /parts se despliega la información de las partes en Cards. Al hacer click en nombre de la parte, se despliega el componente Detail definido por la ruta parts/:cardName, el cual realiza una función que indica si la información se despliega en <input> o <p>. Ahora bien, para el desarrollo del parcial, se decidió crear una carpeta 'components', la cual contiene los componentes de la app, además, se agregó una carpeta 'locales' que permite realizar el proceso de internacionalización. En el componente App, se definen las rutas mediante BrowserRouter, Routes y Route, asignando los componentes específicos para cada ruta (/login, /parts, /parts/:partName), se hace uso de useState y useEffect para detectar el idioma del navegador, y poder asignar los valores estáticos de idioma mediante IntlProvider. Por motivos de tiempo, en el login no fue posible hacer que primero se ingrese el correo, y posterior la contraseña, sin embargo, el forms funciona. Aquí se utilizo useState para guardar y actualizar los valores del login (email, password y rol) y para guardar y actualizar una variable de error de email y password que permite mostrar un mensaje de error en caso de que no se cumplan las validaciones específicas. Se desarrollo un componente partes y parte, partes permite mediante useEffect realizar fetch y recuperar los datos en formato json, que se asignan a una variable denominada partes, la cual se actualiza mediante useState, posteriormente se despliegan las partes en un row, ejecutando la función map que crea componentes 'Parte', este componente simplemente despliega en un Card la información importante de la parte, y define un <Link> para dirigirse al detail de esa parte. Finalmente el detail recupera la ruta y el nombre de la parte mediante useParams, se ejecuta fetch con el uso de useEffect y se realiza la función find() con el objetivo encontrar el elemento que tenga el mismo partName recuperado mediante useParams() y despliega la información validado primero si el rol es true o false, y en caso de que lo sea, despliega la información en un <input> o <p>. Para la internacionalización, en el caso de compontes como input, es necesario exportar el componente como injectIntl(Component), lo que permite usar <FormttedMessage> y a su vez la función .formatMessage() para poder internacionalizar secciones que no se les puede pasar el componente <FormttedMessage>. Cabe aclarar que todo se realizó con la ayuda de react-bootstrap.