

REALITYHOUSE.

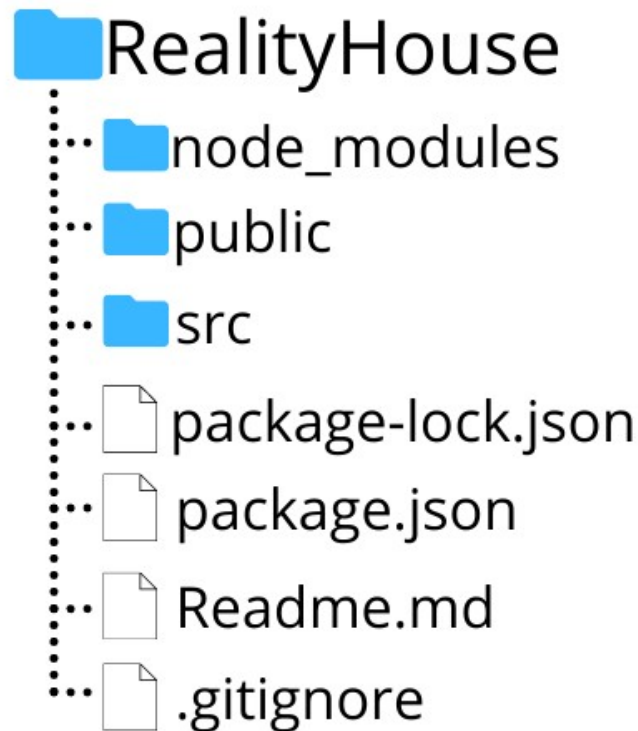


ESTRUCTURA DEL PROYECTO.

Elaborado: Alan Daniel Santiago N
19/01/2022

1- Estructura del Frontend (React).

Se vera primero la estructura general del proyecto y después se procederá a explicar cada carpeta y la funcionalidad de los archivos dentro de ella.



La carpeta principal de RealityHouse contiene la carpeta de node_modules, en esta carpeta se guardan los paquetes o dependencias de nuestro programa de una manera local, para no depender de rutas externas al programa, la siguiente carpeta es la carpeta de public, esta carpeta se encarga de guardar el index.html que es donde se almacena el elemento root que se encargara de mostrar nuestra aplicación en un servidor en linea.

La siguiente carpeta es la carpeta src, en esta carpeta irán los archivos de desarrollo (Rutas privadas, componentes, paginas entre otros), en esta carpeta se guardaran los principales archivos que le darán funcionalidad a nuestro sistema, ademas tenemos el archivo .gitignore, el cual funciona para agregar todos aquellos archivos no queremos dentro de nuestro repositorio.

Por ultimo tenemos los archivos package.json, este se encarga de guardar las versiones y el nombre de las dependencias que estemos usando en nuestro proyecto así como la estructura de nuestros scripts, este contiene la estructura de los archivos de la carpeta node_modules, de tal manera que cuando instalemos las dependencias este funcione para que la carpeta node_modules se genere correctamente.

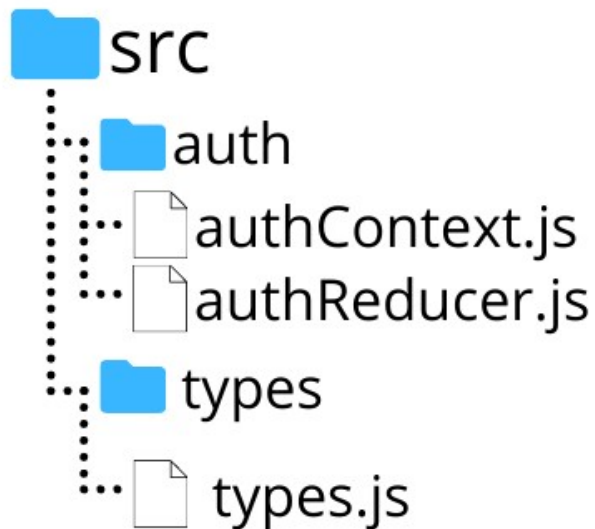
2-Estructura de la carpeta src.

La carpeta src contiene los archivos principales de la aplicación, las funcionalidades, componentes, imágenes entre otros, a continuación se va a proceder a hacer una explicación acerca de cada carpeta y su contenido.



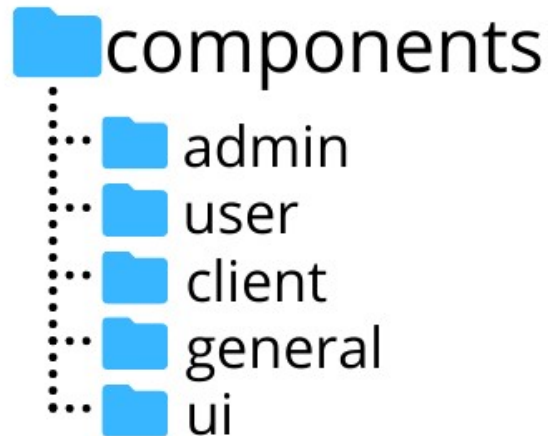
2.1-Carpeta auth y types.

La carpeta auth contiene el archivo authContext.js este se encarga de crear un estado global (contexto) para nuestra aplicación y así poder manejar la información del usuario desde cualquier componente sin necesidad de pasarlo por las props de los componentes, authReducer contiene una función que se encarga de manejar el estado del login del usuario, y types contiene un archivo del mismo nombre con los tipos de login existentes en el sistema (login, logout).



2.2- Carpeta Components

En la carpeta component es donde tenemos los componentes de nuestra aplicación, esta se divide en componentes de usuario, administrador, cliente, generales y ui, cada componente ayuda a reutilizar código y a su vez a hacer más dinámica la aplicación, las carpetas admin, user y client tienen componentes específicos para sus respectivas páginas mientras que la carpeta general abarca componentes que pueden ser utilizados en todas las páginas de la aplicación, por último tenemos la carpeta ui, esta contiene los componentes que forman parte de la interfaz del sistema como el footer o el menú de navegación (navbar).



2.3- Carpeta functions

La carpeta functions contiene un archivo en donde se encuentra la función que se encarga de hacer las peticiones al servidor, dentro de este archivo es donde vamos a insertar la ruta de nuestro servidor para que nuestro programa haga las peticiones correctamente y no genere errores de interfaz, a su vez este archivo contiene la variable urlServer, que es la misma ruta de donde generaremos las peticiones pero esta le ayuda a las páginas y algunos componentes a concatenar las rutas de imágenes o información que se visualizará en ellos.



2.4- Carpeta pages

La carpeta pages contiene las carpetas de general, user, admin y client y el archivo de App.jsx, esta carpeta contiene las páginas de cada módulo del sistema, es decir la carpeta admin contiene las páginas del administrador y así sucesivamente, la carpeta general contiene la vista del login y recuperación de contraseña mientras que el archivo App.jsx es la base en donde cargará la aplicación y el resto de las vistas mediante react-router-dom.



2.5- Carpeta routes

La carpeta routes contiene las rutas para cada módulo del sistema así como los archivos de protección de rutas para las rutas de admin y user y que no cualquiera pueda acceder a ellas, mientras que deja las del cliente (client) a la vista libre ya que estas no es necesario protegerlas.



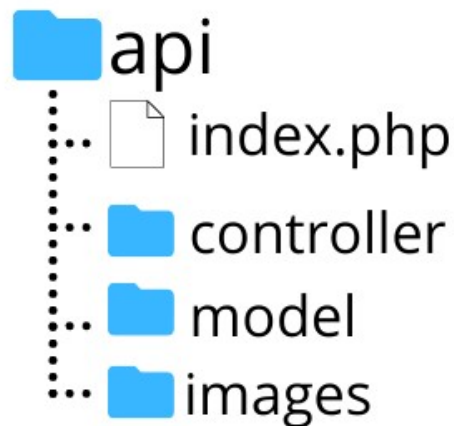
2.6-Carpeta img

La carpeta img simplemente contiene las imagenes utilizadas por la interfaz del sistema.



3-Estructura del backend (PHP).

Ahora tenemos la estructura del backend el cual consiste en un api realizada en PHP, esta carpeta contiene un archivo index.php que es el que se encarga de recibir las peticiones y procesarlas según sea el caso, a continuación podremos ver su estructura y la funcionalidad de sus diferentes ficheros.



3.1 Carpeta controller

Contiene los archivos ctrQuery y ctrCorreo, estos archivos se encargan de recibir la información enviada a index.php desde el frontend y procesarla o trabajar con ella para poder enviarle la información esperada a los archivos dentro de las carpetas de model, en pocas palabras aquí es donde se encuentra la lógica del backend.



3.2 Carpeta model

La carpeta model contiene los elementos mdlQuery y mdlCorreo, estos se encargan de recibir la información ya tratada anteriormente por los archivos del controller y almacenar o ejecutar las consultas y retornar el estado de estas, ya sea satisfactorio o fallido en el caso de mdlQuery, mientras que mdlCorreo simplemente se encarga de procesar la solicitud de mensajes y si no existe mayor inconveniente mandara un resultado exitoso, retornara un reporte con el error, a su vez esta carpeta contiene el archivo de connection.php, este nos permite realizar la conexión de nuestro backend con la base de datos, lo que nos permite generar las consultas desde el backend.



3.3 Carpeta images

finalmente tenemos la carpeta images en esta carpeta se almacenan las imágenes de los usuarios, estas pueden ser sus imágenes de productos, imagen de perfil, o bien los codigosQR de los productos de los usuarios, en general sirve para almacenar las imágenes dentro del servidor.

