**Lista de Requerimientos (Prueba Técnica LEAL)**

Entregables:

1. Desarrollar pagina web con inicio de sesión
2. Alertar al usuario con un error en caso de que el inicio de sesión falle
3. Mostrar después del inicio de sesión el historial de las transacciones del usuario
4. Filtrar las transacciones por fecha de inicio y fecha de fin
5. Filtrar las transacciones por criterio (Descripción)
6. Actualizar la lista con un delay de 500ms después de que se presiona la tecla
7. Mostrar la información del elemento seleccionado.

Documentación:

1. Desarrollar página web con inicio de sesión

Se desarrollaron los elementos necesarios para un inicio de sesión, el usuario tendrá que ingresar el email y la contraseña , todo esto se desarrolla en el archivo app.js pues este será el encargado de hacer todas las peticiones al API, para desarrollar esto app.js se autentica mediante el email y la contraseña, una vez autenticado se almacena en el local storage el token del usuario, este token nos servirá mas adelante para realizar la petición de las transacciones del usuario .Por otra parte, la clase app.js será la encargada de obtener la información de las transacciones del usuario , esta información será enviada a la clase transactions.jsx mediante props.

1. Alertar al usuario con un error en caso de que el inicio de sesión falle

Haciendo uso del local storage , el programa pregunta si existe un token almacenado en el local storage ya que cuando se produce una autenticación exitosa se almacena el token , por lo tanto cuando no haya un token almacenado se le notifica al usuario que la autenticación no fue exitosa , de lo contrario el usuario ingresara y se le notificara de su ingreso exitoso.

1. Mostrar después del inicio de sesión el historial de las transacciones del usuario

Una vez que se autentica el usuario y se obtiene su token, se hace uso de este para hacer la petición a la API y que nos de las transacciones de este usuario , la clase App es la encargada de guardar estos datos en su state y se lo envía mediante props a la clase Transactions , para que esta ultima pueda hacer el render de estos elementos en una tabla.

1. Filtrar las transacciones por fecha de inicio y fecha de fin
2. Filtrar las transacciones por criterio (Descripción)
3. Actualizar la lista con un delay de 500ms después de que se presiona la tecla
4. Mostrar la información del elemento seleccionado.