**Proyecto Individual Soy Henry, DOGS**

*Primeros Pasos:*

* Crear un repositorio nuevo en github.
* Descomprimir el archivo Zip que contiene el PI.
* Hacerle git Bash a la carpeta  
  insertar comandos:
* git Init, git add Readme.md, git commit –m “first commit”, git branch –M main, git remote add origin “https://github.com/Mati69lbt/PI-Dogs-Ft16a.git”, git push -u origin main.
* Creamos una cuenta en la página de Dogs y vamos a buscar la ApIKey en nuestro correo electrónico.

Base de Datos

* Instalamos las dependencias q vamos a necesitar: npm i nodemon express morgan sequelize pg dotenv axios.
* Creamos en la carpeta API, el archivo .env, donde ponemos nuestras credenciales para ingresar a la Base de Datos, este archivo lo va importar db.js y va a hacer dicha conexión.
* Creamos la Base de datos en SQL: CREATE DATABASE dogs.  
  Creamos los modelos como lo pide el README, en la ruta api\src\models\Dog, además creamos otro archivo ahí, lo llamaremos Temperament, en el archivo Dog, definiremos la tabla dog con id, nombre, altura, peso, años de vida, además le crearemos un atributo q nos ayudara para saber si esta creado por nosotros o viene de la api, yo lo llame madeInBD, cada atributo tiene su tipo seteado en string, excepto el ultimo q es un booleano, además cada uno tiene su característica por si es o no requerido obligatoriamente. Al segundo modelo, Temperament, solo la armamos con el name.
* En api\src\db, importamos los archivos de los modelos y vamos hacer las relaciones muchos a muchos (belongsToMany), y la tabla intermedia entre los dos modelos creados.

RUTEO

* Para crear el ruteo, primero creamos una carpeta de Controllers, ahí haremos un archivo donde estará la función que nos traerá la información de la API, en este archivo importaremos los modelos de la base de datos, la apikey para ingresar a la url y dicha url también, que la guardamos en un archivo llamado Constantes. Esta función le hará una petición a la URL con un axios, donde guardaremos esa información en una variable, a esa variable la mapearemos para sacarle solo la información que nosotros queremos (altura, peso, esperanza de vida, imagen, id y los temperamentos) , por ultimo retornaremos la variable con la información procesada. La segunda función q tenemos en este archivo nos retornara la información en nuestra base de datos. Es asíncrona y utilizamos findAll donde incluiremos el modelo de Temperamentos y le pedimos el atributo de name sobre la tabla atributos.
* Y por último, la última función asincrona les pedirá la información a las dos primeras funcione, las concatenara y retornara toda la información procesada.
* En la capeta de ruta crearemos los archivos de las rutas y la uniremos en el Index.
* El archivo q nos traerá todos las razas será un get q nos responderá con la información que traemos de los controllers, aquí también le agregue la ruta por ID, esta vendrá por params, y me ayudara a filtrar la información y devolver solo el perro que estoy buscando. EN este mismo archivo tengo otro get donde llamo a la función que cree en controllers, esta me devolverá toda la información procesada.
* Otro archivo nos responderá con la raza de perro según le pasemos por query, esta filtrara toda la información que trajimos de los controllers, para no tener errores usaremos la función toLowerCase.
* En la carpeta controllers tenemos otro archivo q solo nos traerá los temperamentos de los perritos, debido a que la información nos llega en un string con varios elementos adentro, la tendremos que procesar con varias funciones hasta que nos devuelve de forma ordenada la información, una de las funciones importantes es el findOrCreate nos guardara los temperamentos en el modelo y no las repetirá, por ultimo las retornaremos, y las capturamos en el archivo de que esta en ROUTES en el get que tiene el archivo (all\_temperaments).
* En la carpeta ROUTES tenemos el arcrivo POST, desp de importar todos los elementos que necesitamos, crearemos el.post, este nos traerá por body toda la información que ya solicitamos en el modelo, luego crearemos con la función create los elementos que nos mandaron, a los temperamentos los buscaremos en nuestro modelos y se los agregaremos al perrito que creamos, por ultimo enviamos un send con una frase confirmando la creación.
* En el Index de routes importamos las rutas y determinaremos como usarlas.

FRONT END

Lo primero que hacemos es instalar las dependencias: redux-thunk, redux, axios, react, react-redux, react.router-dom.

Vamos a crear las carpetas q nos contengan nuestros archivos: actions, components, imgs,reducer,store,styles,redux-devtools-extension.

En el Store, en el archivo index.js importaremos todo lo q necesitemos, luego exportamos el store con sus funciones del reducer, y las demás.

En el index.js general de src/client, importaremos el provider y el strore, con estos elementos envolveremos el App.

En App.js importaremos todos nuestras funciones de los archivos del componente, y las propiedades de react-router.dom, donde el browserRouter envolverá al switch y este lo hara con todas las rutas para nuestras paginas.

En la carpeta componentes, crearemos todos los archivos.js que utilizaremos en la pagina.

En la LandigPage importaremos react, link y los estilos, la función exportara un titulo y un botón para ingresar al home.

En el index de Actions importaremos axios, y luego crearemos todas las funciones que utilizares, tenemos dos tipos de funciones:

1. Las funciones donde nos conectaremos con el back: estas nos proveerán la información que pedimos en el back a la api o a la base de datos, estas retornaran un type y un payload.
2. Las funciones de orden y filtro: estas también despachan un type y un payload.

En el index de Reducers