

Reporte de Avance

Telemetría

Área: Eléctrica
Encargada: Catalina Domínguez
Subárea: Telemetría
Encargado: Maximiliano Vargas
Integrantes: Analía Banura
Maximiliano Vargas
Fecha: 30 de Marzo, 2020

1. Resumen de avance

1. Se termina el curso de VueJS, Vuex y Quasar por lo cual se comienza a iterar sobre el front-end a utilizar para eolian fénix y eolian áuriga. Tutoriales
2. Se da con una librería para usar Chart.js en Vue. Vue-Chartjs
3. Se juega con la interfaz logrando diferenciar layouts de pages, y se generan los layouts base para ver información básica y completa de cada versión eolian.
4. Se cambia el modo del router de VueJS de hash a history para evitar el // al iniciar la app en localhost. Sucedió que la app se iniciaba en localhost:8080// y las rutas se cargaban como localhost:8080//otraruta. Diferencia de Modos de Ruta Video explicativo
5. Se juega con el Store de Vuex para reflejar cambios en la interfaz mediante un botón.
6. Antes de la visualización se necesita saber qué datos visualizar, para ello se trabajará entonces en el recorrido de los datos
7. Se da con una librería que extiende socket.io. Vue-Socket.io-Extended
8. Se trabajarán los datos de forma global sobre Vuex, para así ocupar sólo los que se necesiten en cada componente de la App. Así Vuex nos garantiza eficiencia sobre los componentes de la App
9. Se modulariza una store de Vuex para cada versión eolian. El store se divide en archivos como lo sugiere Quasar: actions.js, getters.js, index.js, mutations.js y store.js.
10. En el state se generan los pares (componente, [valores]) correspondientes a eolian fénix revisando los valores enviados desde el Arduino
11. Surge la idea que este archivo tipo ECMAScript Module (Se investiga que es de este tipo) sea el archivo que consuma el Protocolo en Java para iniciar los Componentes. Para eso habría que transformarlo a JSON
12. Se crea la clase JSONToComponent en Java para poder lograr lo anterior, sin éxito al pasar un archivo ESMModule, pero con éxito al pasar un archivo JSON. Java read JSON and write JSON
13. Se aprenden las principales diferencias entre archivos ES y JSON. Diferencias
14. Surge la idea de transformar a mano un ES a JSON con un script de Java usando lectura línea a línea, sin embargo se descarta y se buscan más opciones.
15. Se invierte la propuesta: Pasar de un JSON a ES, para que el Protocolo consuma el JSON y Vuex consuma el ES.
16. Se encuentra una librería de NodeJS para pasar de JSON a ES, por lo que bastaría un script en JS para ganar lo que se quiere. Librería
17. Otra idea: Importar directo el JSON a la store.js con un require. Require en Store Import JSON in ES6 Export in JSON then import using ES6

18. Se identifica un problema: Al iniciar el Componente (de Java), aunque venga configurado del JSON, ¿Cómo sabría la clase que lee el CAN updatear los valores? ¿Mapear las funciones de update y asignarlas a los componentes con el id del Componente?
19. Fix menor en código de Arduino de eolian Fénix
20. Se actualizan los Github con los cambios realizados
21. Se incluye una carpeta Reportes con el presente reporte en el repositorio de eolian Áuriga

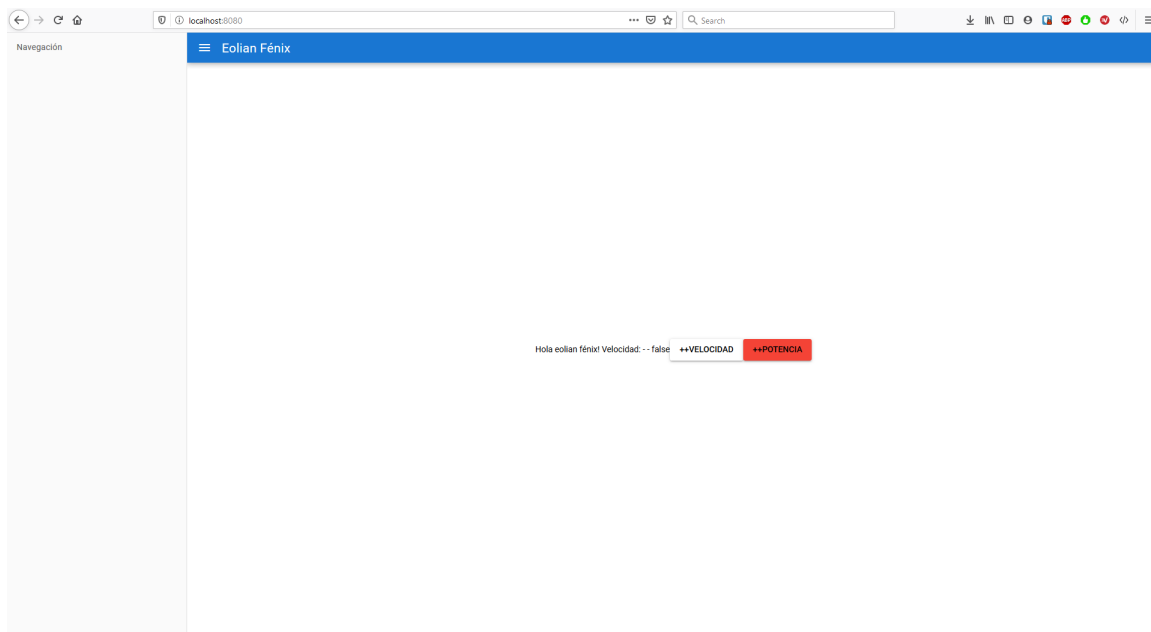


Figura 1: App vacía de eolian Fénix