* Dudas anteriores
  + Feedback
    - Es cierto que me lie un poco con Axios y Nuxt
      * Son muchos frameworks , librerías , etc , lo siento
    - Ejercicios largos y con poco tiempo
      * Entendido , hoy intentare en el ejercicio completo , e incremental , no me importa el tiempo que tardemos , pondré fases o hitos donde tendremos que llegar
    - Los cambios organizativos Nuxt y Capacitor fue para evitar que llegado el momento tuviéramos problemas de compatibilidad
* **19 Capacitor**
  + [19\_Capacitor](https://calleasesoresinformaticos-my.sharepoint.com/personal/enrique_calleasesores_es/Documents/empresas/Imagina/Curso_Vue_2_6_14/Exposicion/Sesion_07/19_Capacitor.docx)
* Ejercicio 99\_sesion\_06\_01:

Una aplicación en vue, donde se refleje los puntos mas importantes del curso

Usaremos el sistema de enrutado vue-router@3 y ficheros vue en carpetas ( “views” ) Para cambiar de página.

También tendremos un store general [vuex@3.6.2](mailto:vuex@3.6.2) donde estará toda la info en memoria

Tendremos un cliente Http que apunte al api de github

https://api.github.com/users/%%USERNAME%%/repos

Usaremos vuetify para la presentación

Hito0 : Estructura , con vuex, route

Hito1 : Un link /alumnos donde se refleje la tabla de alumnos del store

Hito2: Una directiva secret donde se ocultan las palabras marcadas con esa directiva

Hito3: Un servicio http ( axios o vue-resource o nada ) donde recoja los repositorios de github de un determinado usuario ( pasado por parametro)

Nada: fetch

Vue-resource: “^1.5.3”

* npm i vue-resource
* revisar main.js
* this.$http.get ( Devuelve el payload en body )

Axios:”^0.18.0”

* npm install --save axios vue-axios
* npx vue add axios ( quitar el options de axios.js)
* revisar main.js

Hito4: Un componente boton , donde se vea el nº de proyectos que tiene el user almacenado y al darle click provoque su refresco

Hito5: Una tabla , donde se reproduzcan los proyectos de github de un usuario que este almacenado en el store , este nombre de usuario puede ser cambiado por desde la tabla

* Pasos:
  + Montamos un core Vue ( npx vue create 99\_sesion\_06\_01 )
  + Agregamos vuetify ( npx vue add vuetify )
  + Montamos un store ( npm i [vuex@3.6.2](mailto:vuex@3.6.2) )
    - Lo registramos en el main.js

import Vue from 'vue'

import App from './App.vue'

import store from './store'

Vue.config.productionTip = false

new Vue({ store, render: h => h(App)}).$mount('#app')

* + Necesitaremos enrutar a varias paginas /componentes ( npm install vue-router@3 ) ( Cuidado la versión de vue 2.6.14 solo admite hasta la versión 3 )
    - Utilizamos las carpetas views para las paginas
      * Similar al ejemplo 13\_routing que hiciemos en la sesión 3
    - Montamos una carpeta router/index.js similar

import Vue from 'vue'

import VueRouter from 'vue-router'

import HomeView from '../views/HomeView.vue'

import NotFoundView from '../views/NotFoundView.vue'

Vue.use(VueRouter)

const routes = [

{ path: '/404', component: NotFoundView },

{

path: '/proveedores',

name: 'proveedores',

component: ProveedoresView

},

{

path: '/',

name: 'home',

component: HomeView

},

{

path: '/about',

name: 'about',

component: () => import('../views/AboutView.vue')

},

{ path: '\*', redirect: '/404' }

]

const router = new VueRouter({

routes

})

export default router

* + - En main.js incorporamos
      * import router from './router'
      * new Vue({ router, render: h => h(App)}).$mount('#app')