

TEXT CLASS REVIEW

TEMAS A TRATAR EN LA CUE:

- ¿Qué es Vuex?
- ¿Para qué y cuándo utilizar Vuex?
- Similitudes entre un componente y Vuex.
- Instalar Vuex por vue/cli.
- States, utilización de state. Map State.

¿QUÉ ES VUEX?

Es una librería con patrón de manejo de estados, para aplicaciones vue.js. Sirve para almacenar centralizadamente, datos para todos los componentes en una aplicación, con reglas que aseguran que el estado de éstos, puedan ser cambiados únicamente en una forma predecible.

¿PARA QUÉ Y CUANDO UTILIZAR VUEX?

En el caso de querer comunicar datos entre componentes que no son parientes directos. Esto lo haremos consumiendo datos desde un mismo almacén, que será compartido por dichos componentes indirectos.

Algunos ejemplos donde podríamos utilizar Vuex. serían:

- Carrito de compras.
- Reservas de horas.
- Aplicaciones de monitoreo.
- Entre otros.

Vuex está compuesto por:

- **Estates**(estados): son la fuente de datos de los componentes.
- **Getters**: al igual que los estados, son fuentes de datos procesados. Actúan similar a las propiedades computadas.
- **Mutations**: son funciones que nos permiten modificar estados.
- **Actions**: son métodos encargados de ejecutar mutaciones, y pueden realizar llamadas asíncronas.

SIMILITUDES ENTRE UN COMPONENTE Y VUEX

- En state almacenaremos los datos, sería similar a lo que es data dentro de un componente.
- Las acciones (Actions) en Vuex, serán donde podremos llamar datos de una API, y las mutaciones(mutations), generando lógica; es similar a los métodos de un componente.
- Los getters, serán datos calculados a partir de states; debemos pensar en ellos como las propiedades computadas de Vuex.
- Las mutaciones(mutation), nos sirven para acceder a los estados y poder modificarlos; es la única forma de poder cambiar(mutar) datos del state de Vuex.

```
const app = new Vue({
  data: {
    ...
  },
  methods: {
    ...
  },
  computed: {
    ...
  }
})
```

```
const store = new Vuex.Store({
  state: {
    ...
  },
  mutations: {
    ...
  },
  actions: {
    ...
  },
  getters: {
    ...
  }
})
```

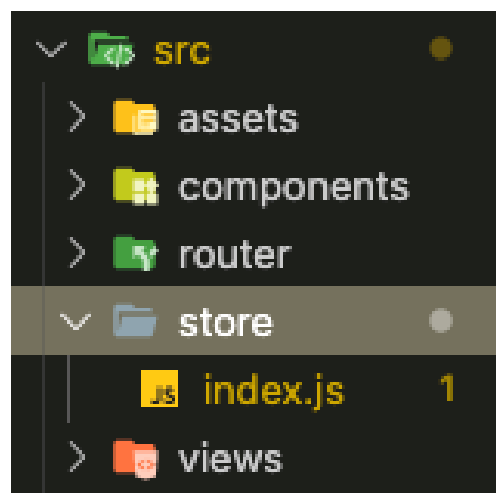


INSTALAR VUEX POR VUE/CLI

Cuando creamos un nuevo proyecto a través de vue/cli, nos pregunta si deseamos agregar Vuex.
De esta forma, nos creará automáticamente el archivo necesario para trabajar.

```
? Please pick a preset: Manually select features
? Check the features needed for your project:
  ● Choose Vue version
  ● Babel
  ○ TypeScript
  ○ Progressive Web App (PWA) Support
  ● Router
  > ● Vuex
  ○ CSS Pre-processors
  ● Linter / Formatter
  ○ Unit Testing
  ○ E2E Testing
```

Al terminar la instalación, vamos a obtener una nueva carpeta dentro de "src", llamada: "store".



Al abrir index.js, veremos la estructura vacía de Vuex.

```
1  import Vue from 'vue'
2  import Vuex from 'vuex'
3
4  Vue.use(Vuex)
5
6  export default new Vuex.Store({
7    state: {
8    },
9    mutations: {
10   },
11   actions: {
12   },
13   modules: {
14   }
15 })
```

El último archivo en ser modificado es raíz main.js, veremos que se importa store, y se agrega en la instancia de Vue.

```
1  import Vue from 'vue'
2  import App from './App.vue'
3  import router from './router'
4  import store from './store'
5
6  Vue.config.productionTip = false
7
8  new Vue({
9    router,
10   store,
11   render: h => h(App)
12 }).$mount('#app')
```