**VUE**

Instala glogar httpserver: npm instal --global http-server

De manera sin instalar:

<script src="https://cdn.jsdelivr.net/npm/vue@2/dist/vue.js"></script>

Correr en terminal: http-serve

Se crea la instancia de Vue:

<script>

        new Vue({

            el: '#app'

        })

    </script>

Y encima la división con la etiqueta

<div id="app">

        <h1>Hey</h1>

    </div>

El manejo de los datos se crea en forma de función     <script>

        new Vue({

            el: '#app',

            data(){

                return {

}

            }

        })

    </script>

Para mostrar la variable de pone el la división de la etiqueta con {{ }}

<div id="app">

        <h1>{{nombre}}</h1>

    </div>

    <script>

        new Vue({

            el: '#app',

            data(){

                return {

                    nombre: "Carlos"

                }

            }

        })

    </script>

**COMPONENTES**

Se crea carpeta de components dentro, un archivo javascript .js e implementar el componente como

Vue.component('mensaje')

Vue.component('mensaje', {

    data(){

        return{

            mensaje: "hola guapo"

        }

    }

})

Importar el mensaje del componente:

<script src="./components/mensaje.js"></script>

Traer el componente en el body/div

<mensaje></mensaje>

Crear el template para mostrar el mensaje

Vue.component('mensaje', {

    data(){

        return{

            mensaje: "hola guapo"

        }

    },

    template: `

        <div>

            <h1>{{mensaje}}</h1>

        </div>

    `

})

**Computed prop (propiedades computadas)**

Se crea el archivo computed-prop.js en la carpeta components

Propiedad de modularidad para poder usar las variables después en otros componentes.

Vue.component('computed-prop', {

    data(){

        return{

            nombre: "Carlos",

            apellido: "Torregrosa"

        }

    },

    computed:{

        nombreCompleto(){

            return `${this.nombre} ${this.apellido}`

        }

    },

    template: `

        <div>

            <h1>Esto es un ejemplo</h1>

            <h3>{{nombreCompleto}}</h3>

        </div>

    `

})

**Métodos**

Sirven principalmente para realizar acciones sobre la data ingresada en el componente, se crean como funciones y se llaman haciendo uso de los métodos como **v-on:click=” ”**