

# ADM

Dudas?



# ADM

Dudas?



## Validar elementos del Formulario

- Dependiendo del **tipo de campo** y de la **información que queramos almacenar**, hay que tener presente el **aviso al usuario** para facilitarle la tarea.

El título es obligatorio.

Debe seleccionar un elemento.

El año es obligatorio.

# VueJS

- Indicarle al usuario, lo que nosotros esperamos de ese campo, le dará la **retroalimentación necesaria** y tendrá una **buenas experiencia**.
- Desde vuejs podemos hacerlo sin problemas.
- Cuando el usuario **quiera enviar el formulario**, si hay errores, le comunicaremos **qué es lo que debe corregir**.
- Sabemos que gracias a html5, el browser nos permite realizar **validaciones**, pero vamos a correrlo de esta responsabilidad para que **vue se encargue de esto**.

# VueJS

```
<form v-on:submit.prevent="guardar" novalidate >
```

- El atributo **novalidate**, permite anular las validaciones típicas del browser por medio de los atributos de html5
- Podemos generar un **array de errores** e inicializarlo desde el data del componente form
- Iremos verificando si estos elementos, tienen contenido, o si son null, o si cumplen con determinada cantidad de caracteres, etc
- Si no cumple un campo con lo que estamos esperando que ingrese el usuario, haremos un push al array de errores y creamos el mensaje que verá el usuario.

# VueJS

```
data:function(){  
    return {  
        titulo:null,  
        anio:null,  
        errores:[],  
    }  
}
```

Desde nuestro componente inicializamos las variables asociadas a los **v-model** y el **array que será responsable de ir mostrando los errores**

- *Una vez que el usuario quiera enviar el form ejecutaremos las comprobaciones necesarias.*

```
if (!this.titulo) {  
    this.errores.push('El título es obligatorio.');//  
}  
if (!this.anio) {  
    this.errores.push('El año es obligatorio.');//  
}
```

# VueJS

- Desde la **propiedad template del componente** tendremos el html para mostrarlos

```
<div v-if="enviado === true">
  <div v-if="this.errores.length > 0" class="classerror">
    <ul>
      <li v-for="x in errores" >{{x}}</li>
    </ul>
```

Si hay errores, recorrerá el array de errores y los mostrará en la iteración

```
if(this.errores.length == 0){
  nuevoObj = {
    comentario: this.comentario,
    titulo: this.titulo,
    seleccion: this.seleccion,
    anio: this.anio
  }
```

Si no hay errores, creamos un objeto en la lógica con todos los v-model de los controles del formulario

# VueJS

- Caso contrario, significa que no hay errores y mostrará el mensaje de "Enviado con éxito"

```
<div v-else class="enviado">  
    <span>Enviado con éxito</span>  
</div>
```

Enviado con éxito

- Una vez que se cumplan las validaciones y todo esté ok, verificaremos que `this.erros.length` sea == 0
- Recién en ese momento verificaremos si existe el key en `localStorage` y haremos el push del nuevo juego



# VueJS

- Parsearemos la información y guardamos en localStorage

```
nuevoObj={comentario: this.comentario,  
           titulo: this.titulo,  
           seleccion: this.seleccion,  
           anio: this.anio,}  
  
if(!localStorage.dato){  
    this.arr=[]  
}else{  
    this.arr=JSON.parse(localStorage.getItem("dato"))  
}  
this.arr.push(nuevoObj)  
localStorage.setItem("dato",JSON.stringify(this.arr))
```

# VueJS

- *Con esta lógica, solo veremos lo que se va ingresando cuando se ejecute la función de guardar.*

## VueJS

```
<div v-if="this.arr.length > 0">
  <h2>Datos</h2>
  <ul>
    <li v-for="item in arr">
      {{item.titulo}}, {{item.comentario}},
      <span v-for="x in item.seleccion">{{x}}, </span>{{item.anio}}</li>
    </ul>
  </div>
```

- El condicional, nos permite controlar que el **titulo Datos no se muestre si no hay elementos que mostrar**
- En cuyo caso, caemos en el **else** y le mostraremos al usuario algún mensaje para que cargue datos.

```
<div v-else>
  <p>Empezá a cargar tus juegos!</p>
</div>
```

# VueJS

```
<div v-if="this.arr.length > 0">
  <h2>Datos</h2>
  <ul>
    <li v-for="item in arr">
      {{item.titulo}}, {{item.comentario}},
      <span v-for="x in item.seleccion">{{x}}</span>{{item.anio}}</li>
    </ul>
  </div>
```

- El condicional, nos permite controlar que el **titulo Datos no se muestre si no hay elementos que mostrar**
- En cuyo caso, caemos en el **else** y le mostraremos al usuario algún mensaje para que cargue datos.

```
<div v-else>
  <p>Empezá a cargar tus juegos!</p>
</div>
```

# Dudas?



# ADM

Dudas?

