# Introducción a Componentes Web y Vue Js

## Elementos principales y características de Vue JS

{desafío} latam_

# ¿Qué aprenderemos en este módulo?

Implementar una interfaz de usuario web con elementos interactivos utilizando el framework Vue.js para dar solución a un requerimiento.

{desafío} latam_

***Describir los aspectos fundamentales de un framework orientado a componentes para el desarrollo de una aplicación Front-End.***

- Unidad 1: Introducción a Componentes Web y Vue Js
- Unidad 2: Binding de formularios
- Unidad 3: Templates y rendering en Vue
- Unidad 4: Manejo de eventos y reutilización de componentes
- Unidad 5: Consumo de datos desde una API REST

# ¿Qué aprenderás en esta sesión?

- *Reconoce los elementos principales y características de un componente Vue.js.*
- *Reconoce las características y beneficios de la utilización de componentes en una aplicación Front-End.*

{desafío} latam_

# ¿Has oído hablar alguna vez de los frameworks y las librerías?

{desafío} latam_

# ¿Cómo crearías templates HTML con JavaScript?

{desafío} latam_

# /* Vue JS */

# ¿Qué es?

Framework de JavaScript **progresivo** que ofrece la posibilidad de crear aplicaciones reactivas basadas en componentes con una amplia variedad de funcionalidades.


{desafío} latam_

# ¿Por qué usarlo?

Con Vue podemos crear interfaces de usuario en mucho menos tiempo de lo que nos tomaría hacerlo con JavaScript puro.

Esto es posible gracias a sus diferentes herramientas integradas, entre las cuales destacan:

- Generación de templates HTML
- Reactividad
- Componetización de la aplicación
- Formato personalizado de archivos con extensión **.vue**
- Posibilidad de crecer progresivamente

# ¿Quiénes lo utilizan?

Muchas de las empresas de alcance internacional utilizan completa o parcialmente Vue Js en sus plataformas. Entre las cuales encontramos:

Nintendo Behance upwork

Un ejemplo nacional es la plataforma Me Vacuno creada por el ministerio de salud de Chile

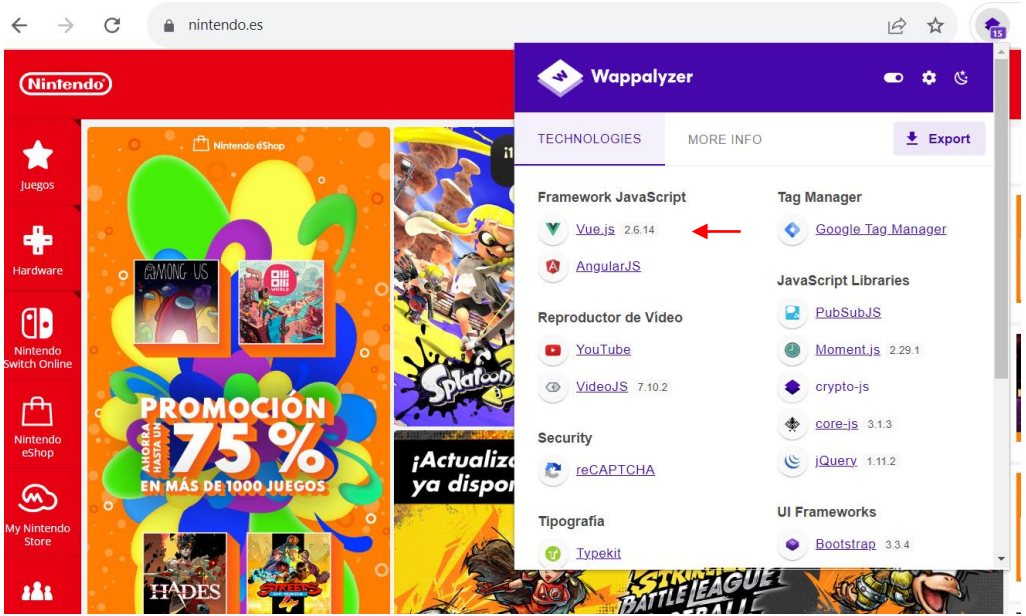
Ministerio de Salud
Gobierno de Chile

#YO ME VACUNO

# ¿Quiénes lo utilizan?

Para saber las tecnologías que utiliza una página web puedes instalar la extensión **Wappalyzer.**

Luego de instalarla, solo deberás entrar a una página web y presionar el icono de la extensión:




{desafío} latam_

# /* Reactividad */

# Reactividad

Es una característica que podemos identificar en las aplicaciones al momento que el usuario realiza una acción y de manera inmediata la aplicación **reacciona con algún cambio visible o en segundo plano**.

Además de tener beneficios en la experiencia de usuario, la reactividad **automatiza la actualización de valores en nuestras variables y templates**.

# Ejercicio guiado

# "Demostración breve de reactividad con Js puro"


{desafío} latam_

# Reactividad con Js

Prueba este código:

```
<input type="text" />

<h1 id="titulo"></h1>

<script>
 const input = document.querySelector("input");

 input.addEventListener("keyup", (e) => {
   const { value } = e.target;
   document.querySelector("#titulo").innerHTML = value;
 });

</script>
```


{desafío} latam_

# /* SPA */

# SPA

## *Single Page Application*

Vue js se conoce como un framework progresivo por la posibilidad de instalar librerías e incluso otros frameworks adaptados a su ecosistema.



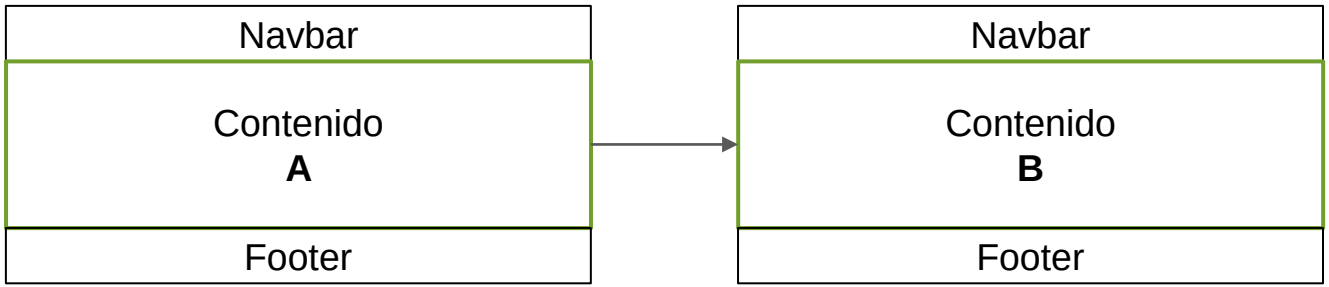

Haciendo uso de estos **plugins compatibles**, podemos crear Single Page Applications.

# SPA

***Single Page Application***

Las SPA tienen como característica principal la **recarga parcial del contenido según una navegación por rutas**.

En una SPA, el usuario permanece siempre en la misma página, sin embargo sus interacciones con la aplicación producen cambios en solo una parte de la página, mejorando inmensamente la optimización de recursos y el rendimiento de la aplicación.



# ¿Qué SPA conoces o utilizas a diario?

# Ejemplos de SPA

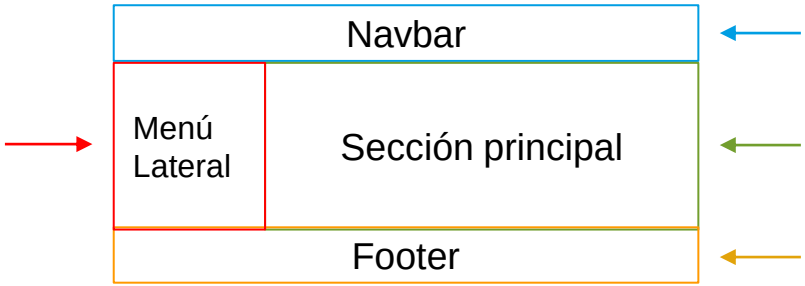
Estas aplicaciones web son SPA:

NETFLIX

# /* Componente web */

# Componente Web

Los componentes web se pueden entender como piezas de un rompecabezas.




En Vue js, los componentes se caracterizan por **contener su propio código HTML, CSS y JavaScript**, además de ser exportables e importables para su uso descentralizado.

# ¿Por qué usarlo?

La componentización nos permite **reutilizar código** y **agilizar** el desarrollo de nuestras aplicaciones.

Así como Bootstrap, podemos crear nuestros propios componentes reutilizables que mantengan una misma identidad en toda la aplicación.

```
<div id="app">
  <Navbar/>
  <HeroSection/>
  <Footer/>
</div>
```

En código, los componentes se incluyen en el template como si se tratasen de etiquetas HTML

# ¿Por qué usarlo?

Además de reutilizar el código, podemos ocuparlos como estructuras preparadas para recibir datos conocidos como ***props*** y ocuparlos internamente.

Este comportamiento lo podemos comparar con los parámetros que utilizamos cuando creamos funciones en JavaScript.

```
<div id="app">
  <BotonPerdonalizado text="Ingresar" />
  <BotonPerdonalizado text="Registrarme" />
  <BotonPerdonalizado text="Cancelar" />
</div>
```

# Ejercicio guiado

# "Creación de un componente con Js puro"


{desafío} latam_

# Creación de un componente con Js puro

Utiliza el siguiente código de ejemplo y crea tu propio componente:

```
<div class="botones"></div>

<script>
 const botonera = document.querySelector(".botones");
 const textos = ["Ingresar", "Registrarme", "Cancelar"];

 const crearBotonPersonalizado = (text) => {
   return `<button> ${text} </button>`;
 };

 for (let text of textos) {
   botonera.innerHTML += crearBotonPersonalizado(text);
 }
</script>
```


{desafío} latam_

# ¿Cómo te resultó la creación de un componente con Js puro?

# Próxima sesión...

- *Reconoce las características y beneficios de la utilización de componentes en una aplicación Front-End (continuación)*

{desafío}
latam_

{desafío}
latam_

Academia de
talentos digitales