# Clase 1: Introducción a Vue.js

## Framework Vue.js

- Creador: Evan You.

- Inspiración: Basado en conceptos de Angular.js, React.js y Rivets.js.

- Descripción: Framework progresivo para construir interfaces de usuario.

- Progresivo: Puede usarse para algo básico o expandirse para aplicaciones complejas.

- Capa de la vista: Se integra fácilmente con otros proyectos o bibliotecas existentes.

## Características de Vue.js

1. Framework Progresivo de Código Abierto:

- Permite la inclusión de herramientas según las necesidades del proyecto.

2. Virtual DOM:

- Representación en memoria del DOM real para minimizar los cambios costosos en rendimiento.

- Los cambios en el DOM real ocurren solo después de aplicar un conjunto de modificaciones.

3. Representación Declarativa:

- Permite crear variables y mostrarlas directamente en la vista.

4. Componentes:

- Unidades de código que representan secciones específicas de la página.

- Poseen sus propios datos, lógica y estilos, y pueden ser reutilizables.

5. Adaptabilidad:

- Fácil de añadir a aplicaciones web existentes sin grandes problemas de instalación.

6. Patrón MVVM (Modelo-Vista-Vista Modelo):

- Modelo: Representa los datos y la lógica de negocio.

- Vista: Muestra la información a través de elementos visuales.

- Vista Modelo: Interactúa constantemente con la vista, manteniéndola actualizada.

7. Sistema Reactivo:

- Mantiene una interacción constante con su entorno mediante eventos.

## Cómo Usar Vue.js

- Desde CDN:

- Enlace directo desde la página oficial de Vue.js.

- De manera local:

- Descarga el archivo Vue.js y colócalo en la carpeta de scripts de tu proyecto.

## Herramienta para Debugging

- Extensión para navegador:

- Vue.js Devtools: Disponible para Chrome y Firefox.

# Manos a la Obra con Vue.js

1. Incorporar Vue a un Proyecto:

- Usa Vue.js de manera local para asegurar su funcionamiento sin conexión a internet.

- Añade un archivo JavaScript donde desarrollarás la lógica de Vue.

2. Ejemplo Básico:

var app = new Vue({  
 el: '#app',  
 data: {  
 mensajito: 'Dato Dinámico'  
 }  
})

- `el`: El elemento del DOM donde Vue se montará (no puede ser `html` o `body`).

- `data`: Objeto que contiene las variables que se usarán en la aplicación.

3. Interpolación:

- Muestra el valor de las variables en la vista usando doble llave (`{{ variable }}`).

- Ejemplo:

<div id="mostrar">  
 <p>{{ mensajito }}</p>  
</div>

- Los datos y el DOM están vinculados, por lo que los cambios en `mensajito` se reflejan automáticamente en la vista.

## Directivas en Vue.js

1. Directivas Comunes:

- `v-once`: Renderiza el elemento solo una vez para optimizar el rendimiento.

- `v-show`: Muestra un elemento de manera condicional (se aplica `display: none` si la condición es falsa).

- `v-html`: Permite mostrar contenido HTML en lugar de texto plano.

2. Directivas de Interacción Avanzada:

- `v-cloak`: Oculta los elementos hasta que Vue.js finalice la compilación.

[v-cloak] { display: none; }

- `v-pre`: Omite la compilación para el elemento y sus secundarios (útil para mostrar código sin procesar).

- `v-bind`: Enlaza dinámicamente atributos o propiedades del DOM a expresiones de Vue.

<img v-bind:src="imagen">

- Sintaxis abreviada: `:src="imagen"`.

3. Manipulación de Clases y Estilos:

- Clases Dinámicas:

<p v-bind:class="[mal, error]"></p>

- Estilos Dinámicos:

<label v-bind:style="booleano ? 'background-color: green; color: white;' : 'background-color: red; color: white;'">  
 Valor del checkbox:  
</label>

4. `v-model`:

- Enlaza elementos de formulario (`input`, `select`, `textarea`) con datos en `data`.

- Ejemplo con checkbox:

<input type="checkbox" v-model="booleano">

## Resumen

- Instancia Raíz de Vue: Creada con `new Vue`, organiza la aplicación en un árbol de componentes reutilizables.

- Sistema Reactivo: Las propiedades en `data` son parte del sistema reactivo de Vue, actualizando la vista automáticamente.

- Vinculación entre Vista y Modelo: A través de interpolaciones, los valores de las variables se muestran en la vista.

- Directivas: Añaden comportamientos especiales al DOM, permitiendo la interacción y dinamismo en la aplicación.