



UNIVERSIDAD TÉCNICA DE ORURO
FACULTAD NACIONAL DE INGENIERÍA
INGENIERÍA DE SISTEMAS E INFORMÁTICA
ORURO - BOLIVIA



PRACTICA # 1

DESARROLLO DE APLICACIONES WEB – INF 3911 A

DOCENTE: Ing. Juan Carlos Vallejos Paniagua

AUXILIAR: Univ. Villalobos Vela Alvaro Edwin

NOMBRE: Daniela Lopez Laime

PARTE TEORICA

1. **Explica el concepto de "reactividad" en Vue.js y cómo se implementa en los componentes.**

La reactividad en Vue . js permite que la interfaz de usuario se actualice automáticamente cuando los datos cambian. Vue lo logra utilizando un sistema de observación: cada vez que una propiedad de datos cambia, Vue vuelve a renderizar el DOM que depende de esa propiedad.

En los componentes se implementa usando el objeto `data()` , donde se declaran las variables que serán "reactivadas". Vue internamente convierte estas propiedades en "reactivas" mediante un sistema de observación basado en getters y setters, lo cual permite que cuando una propiedad se modifica, todos los elementos del DOM que dependen de ella se actualizan automáticamente.

Ejemplo:

```
export default {  
  data() {  
    return {  
      mensaje: 'Hola mundo'  
    };  
  }  
}
```

cambia ,Si `mensaje` cambia, el DOM donde se usa se actualizará automáticamente.

2. **¿Cuál es la diferencia entre métodos, propiedades computadas y watchers en Vue.js?**

- **Métodos (methods):** Son funciones que se ejecutan cuando se llaman. Se usan generalmente para responder a eventos (como clics) o realizar acciones concretas. Siempre se ejecutan cuando se invocan, sin importar si los datos han cambiado o no.
- **Propiedades computadas (computed):** Son similares a los métodos, pero se comportan como propiedades. Se recalculan automáticamente solo cuando las dependencias cambian, y su resultado se cachea hasta que vuelva a ser necesario. Son ideales para derivar datos de otras propiedades.
- **Watchers (observadores):** Se usan para observar cambios específicos en datos o propiedades computadas. A diferencia de los `computed`, se utilizan para ejecutar código como efectos secundarios (por ejemplo, hacer una petición HTTP) cuando los datos cambian.

3. ¿Cuándo es recomendable usar `v-if` en lugar de `v-show`? Proporciona un ejemplo.

La directiva `v-if` se recomienda cuando el elemento no se necesita en el DOM hasta que se cumpla una condición, es decir, cuando se desea crear o destruir el elemento dependiendo del valor lógico.

En cambio, `v-show` solo oculta o muestra el elemento usando `display: none`, lo que significa que el elemento siempre está en el DOM, aunque no sea visible.

Uso recomendado:

- `v-if`: cuando el cambio es poco frecuente o si implica lógica pesada.
- `v-show`: cuando el cambio es frecuente y se necesita alternar la visibilidad rápidamente.

4. Explica cómo funciona la directiva `v-bind` y menciona algunos casos de uso.

La directiva `v-bind` permite enlazar dinámicamente valores de atributos HTML con variables definidas en el componente de Vue. De esta forma, podemos modificar atributos como `src`, `href`, `class`, `style`, entre otros, según los datos que cambian.

En la práctica, `v-bind` se puede abrir simplemente con dos puntos (`:`)

Casos de uso común:

- Asignar rutas dinámicas a imágenes o enlaces (`:src`, `:href`)
- Cambiar estilos o clases CSS (`:class`, `:style`)
- Enlazar valores a atributos personalizados de componentes

5. ¿Cómo se configuran rutas anidadas en Vue Router?

Las rutas anidadas en Vue Router permiten definir subrutas dentro de una ruta principal, lo cual es útil para aplicaciones con vistas jerárquicas. Se configuran utilizando la propiedad `children` dentro del objeto de rutas.

Ejemplo

```
const routes = [
  {
    path: '/usuario',
    component: UsuarioLayout,
    children: [
      {
        path: 'perfil',
        component: PerfilUsuario
      },
      {
        path: 'configuracion',
        component: ConfiguracionUsuario
      }
    ]
  }
];
```

Para que funcione correctamente, el componente `UsuarioLayout` debe tener un `<router-view>` donde se carguen las subrutas.

6. ¿Cuál es la diferencia entre `$route` y `$router` en Vue Router?

- `$route` es un objeto que contiene información de la ruta actual. Esto incluye propiedades como `path`, `params`, `query`, etc. Se usa cuando queremos acceder a información de la URL activa.

Ejemplo:

`this.$route.params.id` nos da acceso al parámetro `id` de la ruta actual.

- `$router` es la instancia del enrutador Vue Router. Se usa para **navegar entre rutas** o modificar la historia del navegador programáticamente.

Ejemplo:

`this.$router.push('/inicio')` nos redirige al componente con la ruta `/inicio`.

7. Explica el concepto de "state management" y cómo Vuex ayuda a manejar el estado en una aplicación grande.

En aplicaciones grandes, muchos componentes necesitan compartir información. Manejar el estado global de forma directa entre componentes puede generar caos y falta de organización. Por eso, es necesario implementar un patrón de gestión de estado.

Vuex es la solución oficial de Vue.js para la gestión del estado global. Centraliza todos los datos compartidos en un almacén único (store), desde donde cualquier componente puede acceder o modificar el estado de manera estructurada.

Vuex utiliza conceptos clave como:

- **Estado** : contiene los datos compartidos.
- **Mutaciones** : métodos sincrónicos que modifican el estado.
- **Acciones** : funciones que pueden ser asincrónicas y luego llaman a mutaciones.
- **Getters** : propiedades computadas del estado.


Con Vuex, la comunicación entre componentes es más clara, el estado es predecible y es más fácil depurar el comportamiento de la aplicación.

PARTE PRACTICA


- 1. Diseñar un sitio web siguiendo el siguiente modelo. Tomar en cuenta la estructura, colores, datos y presentación. Emplear bootstrap5 (HTML – CSS – Bootstrap5). Adjunta y envia los archivos debidamente comprimidos, un link de github o similares al inbox.

Mi Negocio

Inicio Servicios Galería Contacto




Nuestros Servicios




Servicio 1

Descripción breve del servicio 1.



Servicio 2

Descripción breve del servicio 2.



Servicio 3

Descripción breve del servicio 3.

Contáctanos

Nombre

Correo Electrónico

Mensaje

Enviar

© 2024 Mi Negocio. Todos los derechos reservados.

2. Crea un nuevo proyecto en Vue.js. Luego, personaliza la estructura por defecto realizando los siguientes cambios: modifica el contenido de la vista principal (App.vue) para incluir cualquier contenido personalizado, si es necesario organiza los componentes en la carpeta dedicada (/components) e implementa Vue Router para la navegación entre al menos dos vistas diferentes. Adjunta y envía los archivos debidamente comprimidos, con un link de github o similares al inbox.