

# Ciclo de Vida de Vue 2

## Guía clara y práctica con ejemplos reales

El ciclo de vida de Vue describe las etapas por las que pasa un componente desde que se crea hasta que se destruye. Entenderlo te permite saber dónde colocar lógica, llamadas a API y manipulación del DOM correctamente.

### 1. data() — Nacimiento del componente

Aquí Vue define qué datos existirán en el componente. No se ejecuta lógica compleja, solo se declaran valores iniciales.

```
data() {
  return {
    mes: null,
    anio: null,
    proveedores: []
  }
}
```

### 2. created() — La lógica comienza

En esta etapa los datos ya existen y pueden ser modificados. Es ideal para lógica inicial, cálculos y llamadas a APIs.

```
created() {
  const fecha = new Date()
  this.mes = fecha.getMonth() + 1
  this.anio = fecha.getFullYear()
  this.obtenerProveedores()
}
```

### 3. mounted() — El componente ya está en pantalla

Aquí el DOM ya existe. Se utiliza para interactuar con elementos visuales, refs o plugins.

```
mounted() {
  console.log('El componente ya se renderizó')
}
```

### 4. updated() — Re-render por cambios

Se ejecuta cada vez que un dato reactivo cambia y el DOM se vuelve a renderizar.

```
updated() {
  console.log('El DOM fue actualizado')
}
```

### 5. destroyed() — Fin del componente

Se ejecuta cuando el componente se elimina. Ideal para limpiar eventos, intervalos o listeners.

```
destroyed() {
  console.log('Componente destruido')
}
```

## Resumen rápido

- 1 data(): define las variables
- 2 created(): inicializa lógica y APIs
- 3 mounted(): interacción con el DOM
- 4 updated(): responde a cambios
- 5 destroyed(): limpieza final