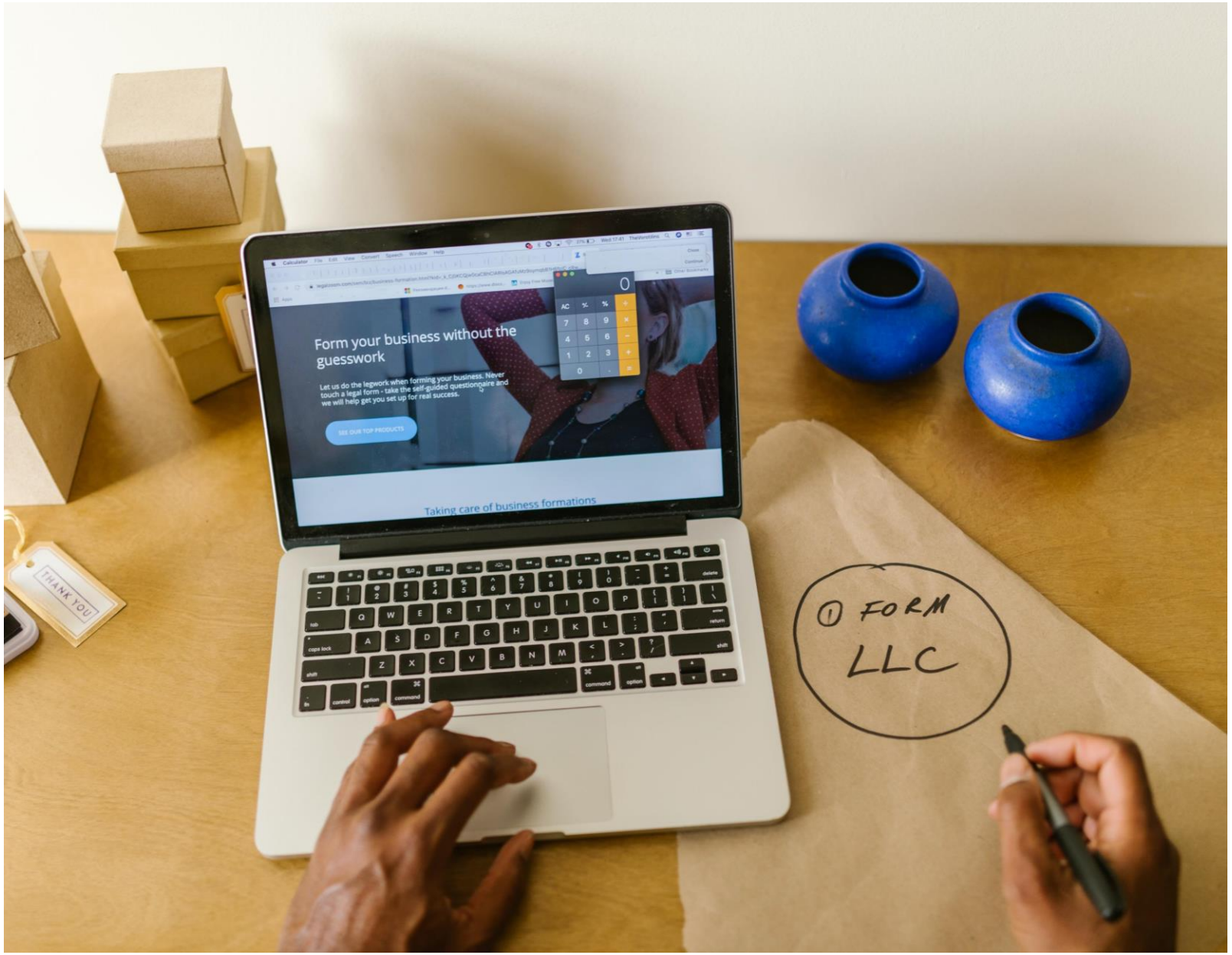

CICLO DE VIDA E OBSERVADORES

DESENVOLVIMENTO WEB III

PROF. DIEGO ROSSI

CICLO DE VIDA

- O ciclo de vida de um componente Vue é dividido em diferentes etapas.
- Cada etapa representando um momento específico na vida do componente.
- Os principais eventos de ciclo de vida no Vue.js incluem:

CICLO DE VIDA

- **beforeCreate:** Este é o primeiro evento acionado antes que o componente seja inicializado. Neste ponto, o componente ainda não tem acesso às opções de dados (data) e métodos (methods).
- **created:** Após o beforeCreate, o componente é criado, e as opções de dados e métodos já estão disponíveis. No entanto, o template ainda não foi montado ou renderizado na página.
- **beforeMount:** Este evento é acionado antes que o template do componente seja montado no DOM.
- **mounted:** O componente foi montado no DOM. É neste ponto que você pode interagir com o DOM, fazer requisições de dados, ou realizar outras operações que requerem a presença do componente na página.
- **beforeUpdate:** Antes que os dados do componente sejam atualizados e o DOM seja renderizado novamente, este evento é acionado.
- **updated:** Após a atualização dos dados e a renderização do DOM, este evento é acionado.
- **beforeDestroy:** Antes que o componente seja destruído, este evento é acionado. É um bom lugar para limpar recursos, como ouvintes de eventos ou timers.
- **destroyed:** Após a destruição do componente, este evento é acionado. O componente não é mais acessível após este ponto.

UTILIZANDO PROPRIEDADES COMPUTADAS PARA FILTROS

```
beforeCreate(){
    ...
}

created(){
    ...
}

beforemount(){
   ...
}

mounted(){
    ...
}
```

```
beforeUnmount(){
    ...
}

unmounted(){
    ...
}
```

OBSERVADORES

- Os observadores são uma das principais características que tornam o Vue.js um framework reativo.
- Eles permitem que você observe e reaja a mudanças em dados e executem código específico quando essas mudanças ocorrem.

OBSERVADORES

- Para criar um observador é necessário criar o objeto watch, dentro desse objeto criamos um método com o nome da propriedade que desejamos observar.

```
watch: {
    message(newMessage, oldMessage) {
      console.log(`Nova mensagem: ${newMessage}, Antiga mensagem: ${oldMessage}`);
    },
 },
```