



VueJS 101

Samuel Carvalho Locatelli

Quem sou eu?

- ▶ Samuel Carvalho Locatelli
- ▶ Estudante de Ciência da Computação (Unisinos)
- ▶ Desenvolvedor .NET (Atualmente em Apisul - Gerenciadora de Risco)
- ▶ LinkedIn: <https://www.linkedin.com/in/samuel-locatelli/>
- ▶ Github: <https://github.com/mukatck>
- ▶ Twitter: https://twitter.com/sml_locatelli

Sobre o que vamos tratar hoje

- ▶ Introdução
- ▶ Conceitos básicos do VueJS
 - ▶ Estrutura
 - ▶ Rendering Pipeline
 - ▶ Ferramenta de debugging
 - ▶ Componentização
- ▶ Vue vs. React vs. Angular
- ▶ Caso: Checklist Apisul

Introdução

Introdução

- ▶ VueJS é um framework JavaScript desenvolvido inicialmente por Evan You
- ▶ Código aberto, disponível no Github
- ▶ Suporte para todos os browsers
- ▶ Opção ótima e leve para criação de componentes reativos

Conceitos básicos do VueJS

Estrutura

- ▶ Toda aplicação começa com a criação de uma instância do Vue
- ▶ Renderização declarativa

index.html

```
<div id="app">  
  Hello world  
</div>
```

index.js

```
var form = new Vue({  
  el: '#app'  
});
```

Classe Vue

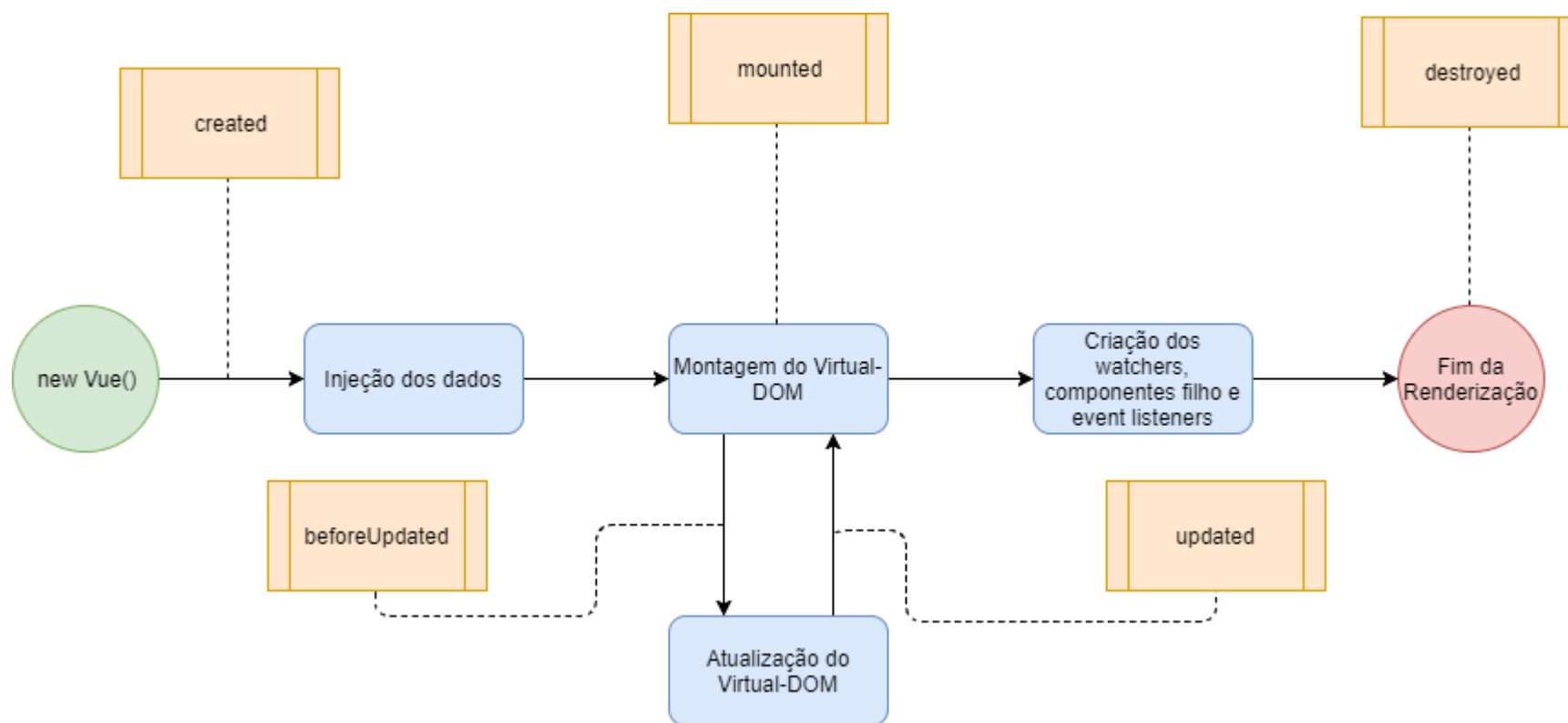
- ▶ A classe Vue é responsável por toda interação entre javascript e a página web
- ▶ Principais atributos
 - ▶ el: Utilizado para declarar o elemento raiz
 - ▶ data: Responsável pelo conjunto de dados (objetos, demais variáveis)
 - ▶ computed: Contém a definição das funções de propriedades computadas
 - ▶ watch: Engloba a declaração das funções de “vigia” de dados
 - ▶ methods: Atributo que irá conter a declaração de todas as funções da classe

Rendering Pipeline

- ▶ O que acontece quando instanciamos uma classe Vue?
 - ▶ Injeção de dados no escopo do elemento ou do template, no caso de componentes
 - ▶ Montagem e atualização do virtual-DOM
 - ▶ Criação dos *watchers*, componentes filhos e *event listeners*
 - ▶ Finalização do processo de renderização
- ▶ Por que eu preciso saber disso?
 - ▶ Tudo é manipulável

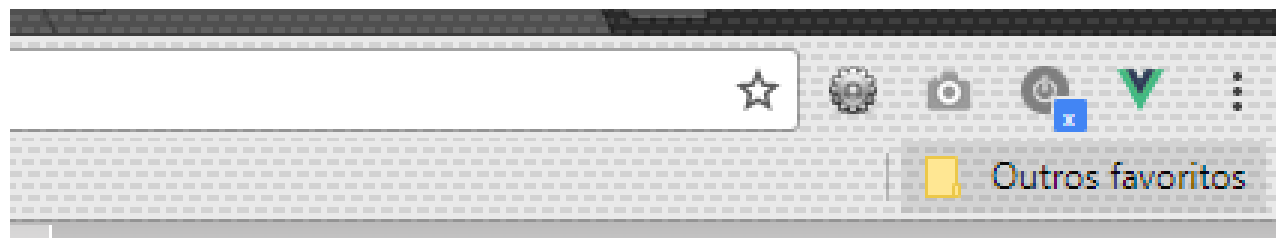
Rendering Pipeline

- ▶ A classe Vue provê meios para execução de funções em vários pontos do processo de renderização

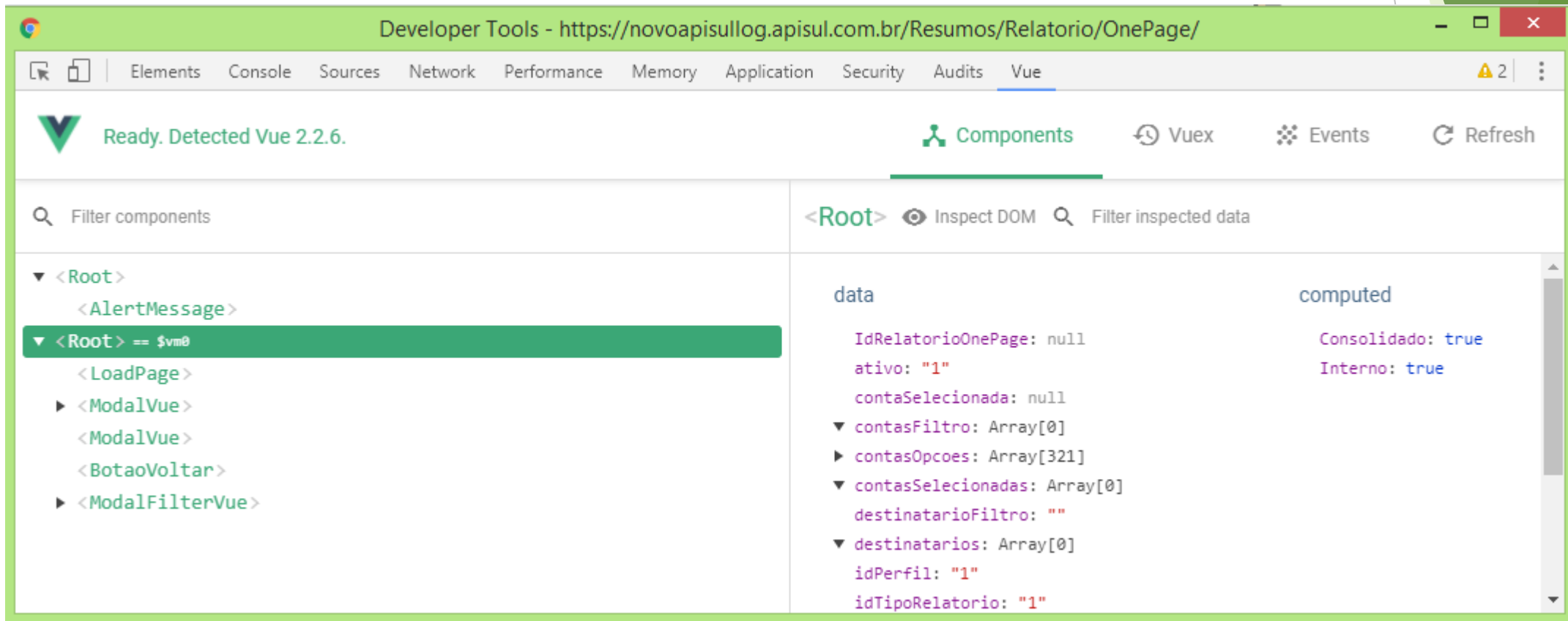


Ferramenta de debugging

- ▶ O google chrome possui uma extensão chamada Vue Devtools
- ▶ Através dela, podemos observar o estado das nossas instâncias do Vue
- ▶ Detecta a presença de Vue na página



Ferramenta de debugging



Developer Tools - <https://novoapisullog.apisul.com.br/Resumos/Relatorio/OnePage/>

Elements Console Sources Network Performance Memory Application Security Audits **Vue** 2

Ready. Detected Vue 2.2.6. Components Vuex Events Refresh

Filter components

<Root> Inspect DOM Filter inspected data

▼ <Root>
 <AlertMessage>
 ▼ <Root> == \$vm0
 <LoadPage>
 ▶ <ModalVue>
 <ModalVue>
 <BotaoVoltar>
 ▶ <ModalFilterVue>

data

- IdRelatorioOnePage: null
- ativo: "1"
- contaSelecionada: null
- ▼ contasFiltro: Array[0]
- ▶ contasOpcoes: Array[321]
- ▼ contasSelecionadas: Array[0]
- destinatarioFiltro: ""
- ▼ destinatarios: Array[0]
- idPerfil: "1"
- idTipoRelatorio: "1"

computed

- Consolidado: true
- Interno: true

Componentização

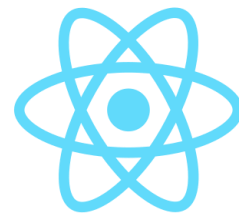
- ▶ É a propriedade essencial para qualquer framework javascript
- ▶ Vantagens da componentização
 - ▶ Melhor legibilidade do código
 - ▶ Reutilização
 - ▶ Detecção e correção de bugs
 - ▶ Implementar melhorias

Componentização

- ▶ Um componente nada mais é do que uma instância pré-definida do Vue
- ▶ Principais atributos:
 - ▶ template: html modelo do componente
 - ▶ props: atributo customizado para passagem de dados para o componente
 - ▶ Os props podem ser tipados (String, Number, Boolean, Function, Object, Array, Symbol)
 - ▶ data: responsável pelo conjunto de dados do componente
 - ▶ methods: funções internas do componente

Vue vs. React vs. Angular

Vue vs. React



- ▶ Ambos os frameworks são bastante parecidos
 - ▶ Foco em componentização e reatividade
 - ▶ Utilizam Virtual DOM
- ▶ O Vue possui algumas otimizações de renderização implementadas
- ▶ Abstração do html e CSS por parte do React
- ▶ A curva de aprendizado do React é maior

Vue vs. Angular



- ▶ Apesar de serem mais diferentes, a parte de sintaxe do html do Angular foi herdada pelo Vue
- ▶ Angular e TypeScript
- ▶ Tamanho dos arquivos
 - ▶ Vue + Vuex + Vue Router (~30KB)
 - ▶ Angular-cli (57KB ~ 130KB)

Caso: Checklist Apisul

Checklist Apisul

- ▶ Propósito da aplicação
 - ▶ Acompanhamento e teste dos sensores e atuadores de um veículo
- ▶ Tecnologias utilizadas
 - ▶ Vue
 - ▶ Bootstrap
 - ▶ WebAPI C#



Resultado

CHECKLIST

GIG8989 **Novo**

Nenhum checklist pendente.

NOVO CHECKLIST

Placa: GIG8989

Observação:

Cancelar **Salvar**


Resultado

CHECKLIST > GIG8989 


Selecione os Sensores e Atuadores para realizar o Checklist:


Sensores (2)	
Porta Motorista	<input checked="" type="checkbox"/>
Botão Pânico	<input checked="" type="checkbox"/>
Porta Carona	<input type="checkbox"/>
Porta Baú Traseira	<input type="checkbox"/>
Porta Baú Lateral	<input type="checkbox"/>

Cancelar **Iniciar**

CHECKLIST > GIG8989 Atualizado às: 17:22 

Dados | Andamento | **Resultado**

Tipo	Dt / Hr Rec.	Descrição
	08/12/2017 17:22:23	Enviar Mensagem Livre

 **COMANDOS**

Resultado

✕    ENVIAR →

Ativar/Desativar

Ativar	Bloqueio Veículo	Desativar
Ativar	Sirene	Desativar
Ativar	Setas	Desativar
Ativar	Trava Porta Baú Traseira	Desativar
Ativar	Trava Porta Baú Lateral	Desativar
Ativar	Trava Porta Baú	Desativar

Configurar

Solicitar

Enviar

CHECKLIST > GIG8989 Atualizado às: 17:24 

Dados | Andamento | Resultado

Sensores Aprovados	Sensores Reprovados
Porta Motorista	
Botão Pânico	

Resultado do Checklist:

Aprovado ▼

Observação:

Liberado para viagem

Encerrar

Perguntas e Respostas

