**Introdução**

é um framework javascript progressivo, para construção de interfaces de usuarios, e teve sua origem para se adaptar com outras bibliotecas e projetos existentes. Criação de componentes, e aplicações SPA( Single Page Application)

Noções de HTML,CSS E javascript

**Comparação Vue e React**

Ambos trabalham com o DOM, e o React 'necessário precisar aprender JSX. Já o Vue, trabalha com templates que são basicamente o html convencional, css e javascript

**Comparação Vue e ANGULAR**

necessário aprender o Typescript, para ser utilizado. Angular e Vue, em performance ambas são muitos rápidos, porém oo vue em sua compilação o projeto final fica muito reduzido em tamanho.

Vue vantagem da flexibilidade

**Iniciando com vue**

Possível escolher diversos templates.

**Iniciando com Vue Cli**

Primeiramente será necessário instalar o npm (Node gerenciador de pacotes), acessando o site do node e instalar de acordo com o sistema operacional

Após instalado, para conformar a instalação do node no terminal/prompt execute o comando

npm -v

Servirá para verificar a versão que está sendo utilizada, caso não exiba averoa, possivelmente o node não foi instalado corretamente.

Com o node rodando Ok, será necessário instalar de forma global o vue-cli, executando o comando:

npm install --global vue-cli

Após o processo estar concluído você poderá utilizar todo o potencial do vue-cli, ele será responsável por gerar um projeto novo e pronto para que apenas nos preocupemos em programar. O site no git do projeto : <https://github.com/vuejs/vue-cli>

Para instalação deveremos executar o seguinte comando:

$ vu init <template-name> <nome-projeto>

O templates disponíveis são:

**webpack :** completo webpack, com vue-loader setup com hot reload, linting, testing & css extraction e vue-router

**webpack-simples :** simples webpack com vue-loader setup para protótipos rapidos

**browserify :** completo Browserify + vueify setup com hot reload, linting, testing & css extraction e vue-router

**browserify-simple :** simples webpack com vue-loader setup para protótipos rapidos

**pwa :** PWA template for vue-cli baseado em um webpack template (Como se fosse desktop e mobile, site todos iguais)

**simple : :** o mais simples possível para o VUe Setup

Dependendo do tipo de template será solicitado no proprio console algumas diretivas.

Para iniciar de forma simples seria o seguinte comando:

$ vue init webpack-simple meu-novo-projeto-vue

ou recomendado utilizando o webpack:

$ vue init webpack meu-novo-projeto-vue

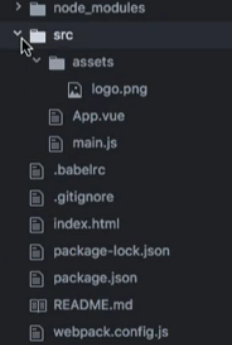
Apos isso, estara pronto o projeto, basta então agora abrir o terminal/prompt novamente na pasta que foi criada e execute a linha:

$ npm run dev

O projeto ira subir, e será inicado o projeto numa url se nada alterado na url:

<http://localhost:8080/>

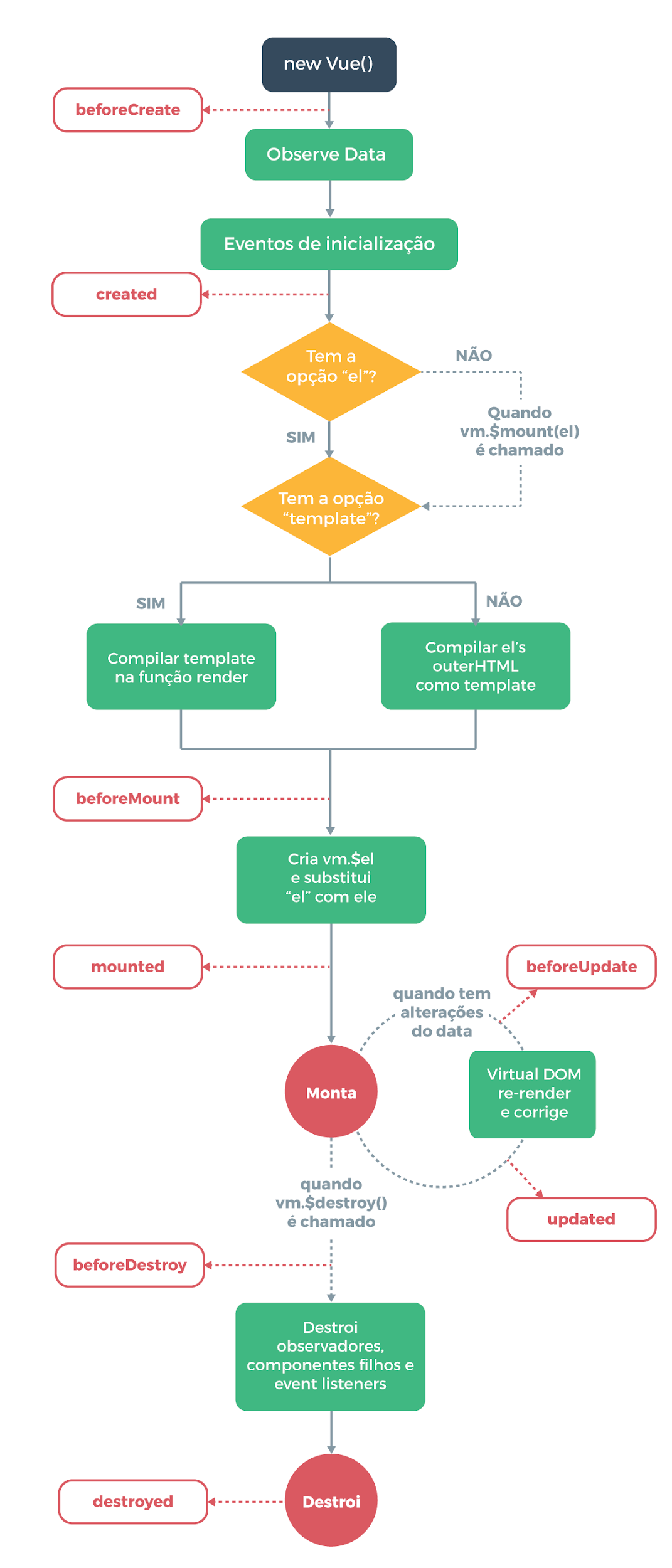
**Estrutura de um Projeto novo Criado pelo vue-cli**

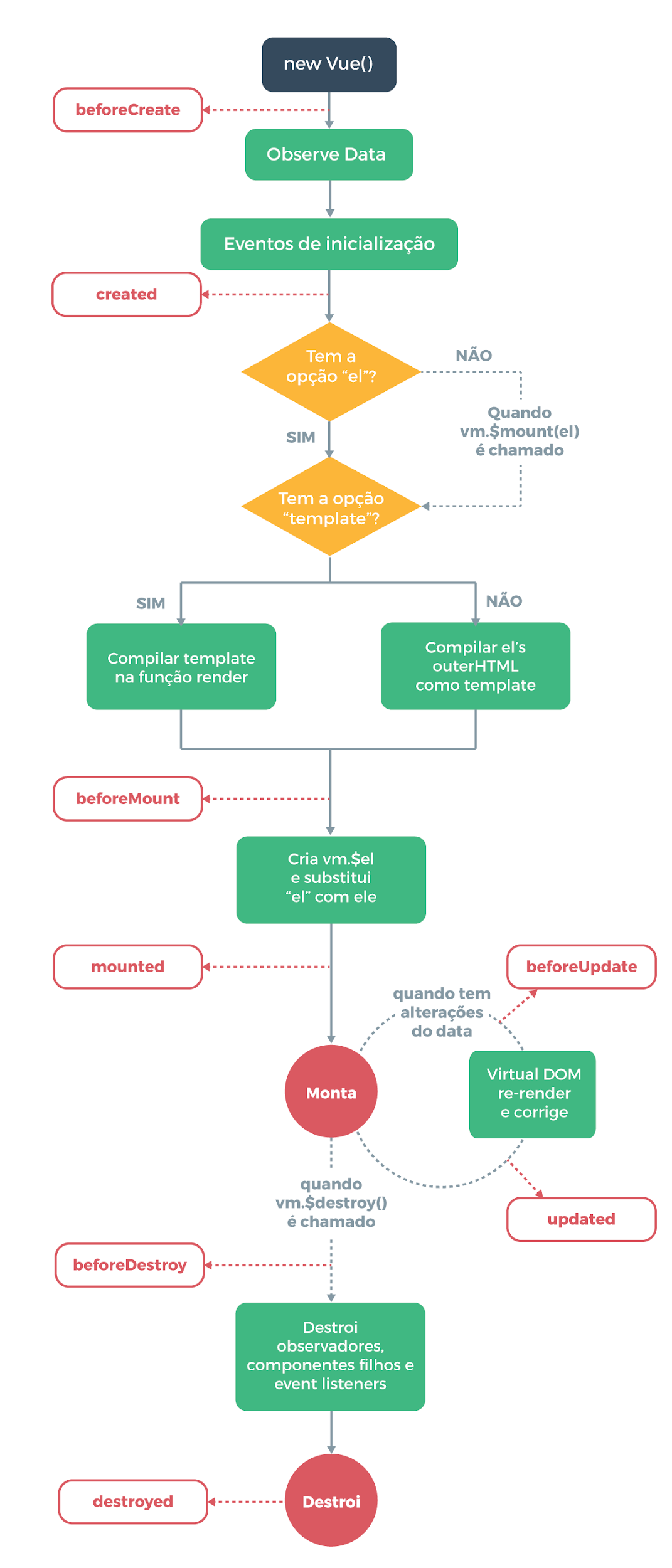
****

**Src:** Onde ficará tudo quer será desenvolvido, componentes, páginas, arquivo main.js

**node\_modules:** aqui ficara todas as dependências e módulos prontos instalados no projeto, ao exportar o projeto esta pasta deve ser ignorada, pois com o arquivo na raiz package.json é o que contem todas as dependências do projeto. Sendo assim ao abrir seu projeto sem a pasta, basta executar o comando: npm install

**Ciclo de Vida**





**Instalando um Servidor ApI JSON**

<https://github.com/typicode/json-server>

Instalar:

$ npm install -g json-server

Crie um arquivo db.json file:

{  
 "posts": [  
 { "id": 1, "title": "json-server", "author": "typicode" }  
 ],  
 "comments": [  
 { "id": 1, "body": "some comment", "postId": 1 }  
 ],  
 "profile": { "name": "typicode" }  
}

Inicie o JSON Server

$ json-server --watch db.json

agora acesse o link<http://localhost:3000/posts/1>, e pronto veja

{ "id": 1, "title": "json-server", "author": "typicode" }

**Requisição ao Servidor**

Diversas opções para uso, o vue-resource, deixou de ser nativo, para que que utilize fique de livre escolha. O mais conhecido e utilizado pelo comunidade no momento é o axios.

Porém neste curso, será utilizado o vue-resource, para isso basta executar o comando :

$ npm install vue-resource

Após instalação será necessário configurar os arquivos na aplicação/projeto, abra o arquivo main.js localizado na raiz da pasta src e coloque o seguinte trecho nele.

import VueResource from 'vue-resource';

Vue.use(VueResource)

e depois para fazer a utilização em seus componentes, utilizar como exemplo de get

{  
 // GET /someUrl  
 this.$http.get('/someUrl').then(response => {  
  
 // get body data  
 this.someData = response.body;  
  
 }, response => {  
 // error callback  
 });  
}

Para mais informações : <https://github.com/pagekit/vue-resource> repositorio do projeto

**Trabalhando com rotas**

Trabalhar com rotas no vue pode ser feito manualmente ou mais simples usando o componente vue-router, que é o mais recomendado.

Para mais informações: <https://github.com/vuejs/vue-router> e <https://br.vuejs.org/v2/guide/routing.html>

Forma mais simples é utilizando o npm

após instalar abra o arquivo main.js e faça a importação do vue-router da seguinte forma:

import VueRouter from 'vue-router'

Vue.use(VueRouter);

depois vá para o arquivo index.html e dentro da tag app, faça a inclusão da tag routerview, ficando:

<div id="app">

<router-view/>

</div>

Após isso abra o seu arquivo main.js novamente e configure as rotas de acordo com seu aplicativo, no meu caso do curso ficou da seguinte forma:

import Vue from 'vue'

//import App from './App.vue'

import VueResource from 'vue-resource';

Vue.use(VueResource);

import VueRouter from 'vue-router'

Vue.use(VueRouter);

import Home from './paginas/Home.vue'

const Contato = {template:'<p>Página de Contato</p>'}

const SemPagina = {template:'<p>Essa página não existe</p>'}

const routes = [

{ path:'/',component:Home},

{ path:'/contato',component:Contato}

]

const router = new VueRouter({

mode: 'history',

routes

})

const app = new Vue({

router

}).$mount('#app')

**Subida para Produção**

para subir para produção você deverá executar o comando npm run build, caso o run dev esteja ativo ainda desative, e rode no terminal:

$ npm run build

Após finalizar o processamento será gerada uma pasta na raiz da aplicação com nome de “dist”.

Copie esta pasta dist, e o arquivo index.html ambos na raiz da aplicação e copie para o servidor.

**Simulando Servidor Local**

Caso não tenha um servidor para testar, ou apache e somente o php. Crie uma pasta na raiz do projeto. Acesse ela pelo terminal e execute o comando

$ php -S localhost:8099

O comando irá iniciar um servidor para ser acessado através de : <http://localhost:8099/>

**Feito isso ai.**