



**Serverless
Side-Rendering
com Vuejs/Nuxt.js
com AWS Lambda**

Quem sou eu?

- Nome: Júlio César Ferreira Guedes.
- Trabalho: Blueticket.
- Função: Analista/Desenvolvedor Front-end.

Linkedin:

<https://www.linkedin.com/in/julioguedes/>

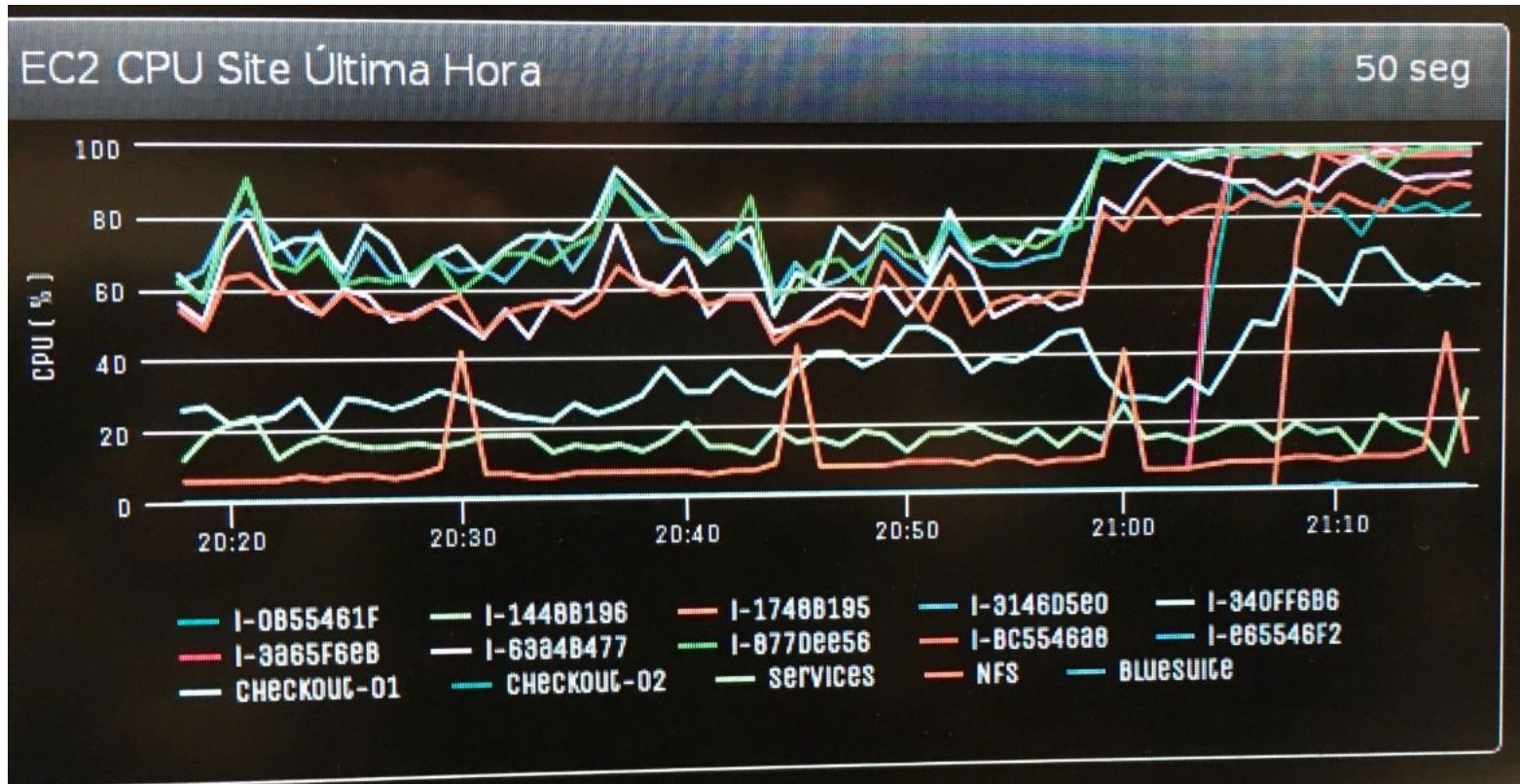
Blueticket

- +2M de usuários cadastrados
- +600 mil usuários únicos mês
- +1 milhão de acessos no total
- 200 á 20 mil usuários simultâneos

Show Guns N Roses



Show Guns N Roses - EC2



Como fica o Auto-Scale ?

- Não daria tempo de subir máquinas tão rápido em um servidor tradicional
- Servidor em carga máxima seria caríssimo




Vue.js

Vuejs é um framework simples para a construção de interfaces de usuário.

fonte: vuejs.org

Github




Vuejs

 Watch ▾	5,879	 Star	136,526	 Fork	19,436
---	-------	--	---------	--	--------

React

 Watch ▾	6,654	 Star	127,699	 Fork	23,308
---	-------	--	---------	--	--------

Angular

 Watch ▾	3,294	 Star	47,395	 Fork	12,642
---	-------	--	--------	--	--------

Vuejs +
Cloudfront +
S3 bucket ?

Problemas com **SEO!**

Nuxtjs (SSR)

É um framework para criação de aplicações Vue.js, seja ela universal, gerada estaticamente ou *single page application*.

fonte: nuxtjs.org

Github

Nuxtjs

 Watch ▾	569	 Star	19,632	 Fork	1,670
---	-----	---	--------	--	-------

Instalação

Instalação - Várias opções

- Qual framework Server-Side quer usar ?
- Qual framework UI quer usar ?
- Qual framework de test quer usar ?
- Axios, eslint, prettier...

Features

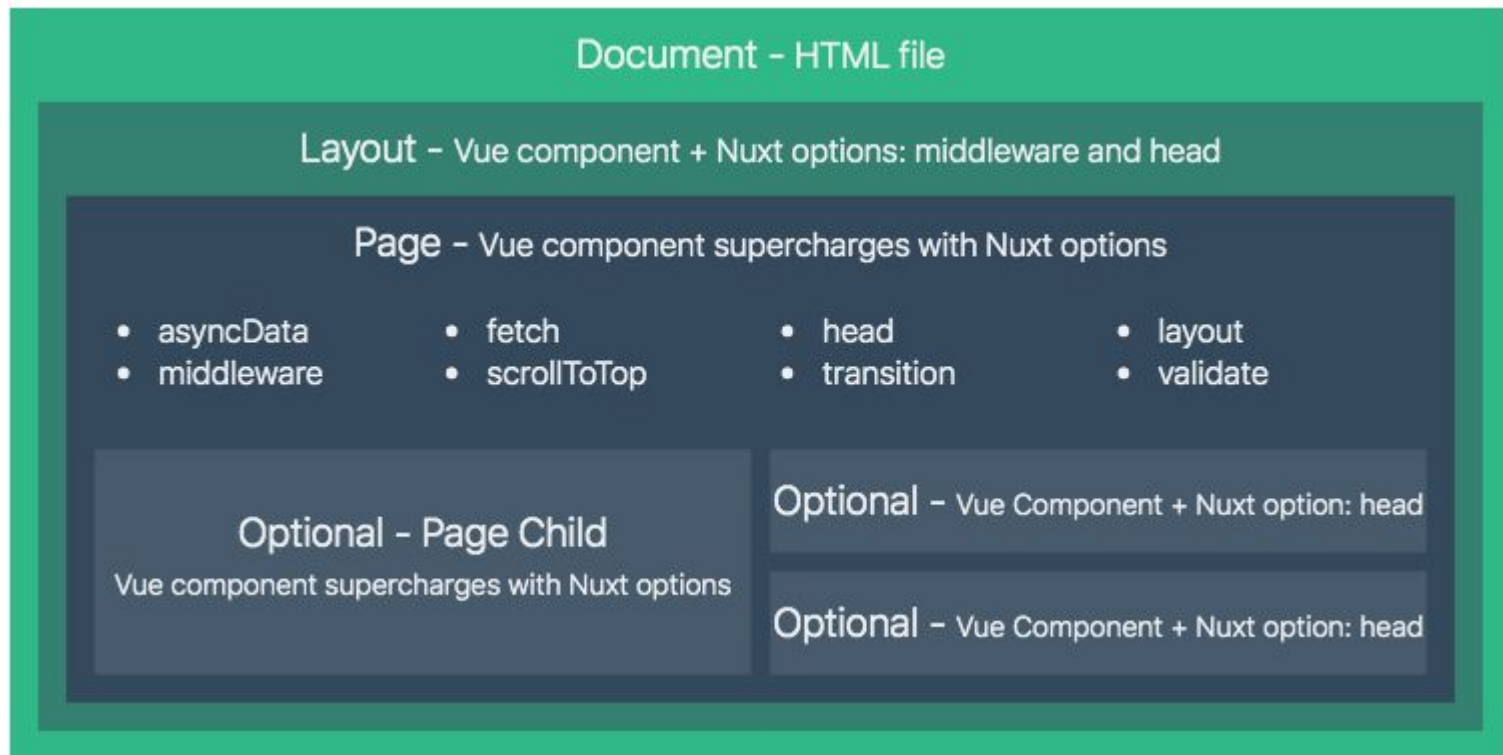
Nuxtjs - Features

- Routing system baseado em pastas e arquivos
- Hierarquia layout/página (propriedades head nas páginas)
- Middlewares
- Plugins
- AsyncData/Fetch

Nuxtjs - Routing (Feature)

- Basic routing
- Dynamic routing

Nuxtjs - Hierarquia Layout/Página (Feature)



Nuxtjs - Middlewares (Feature)

- São funções customizáveis que são executadas antes de renderizar uma página
- Ficam localizados na pasta **/middleware**

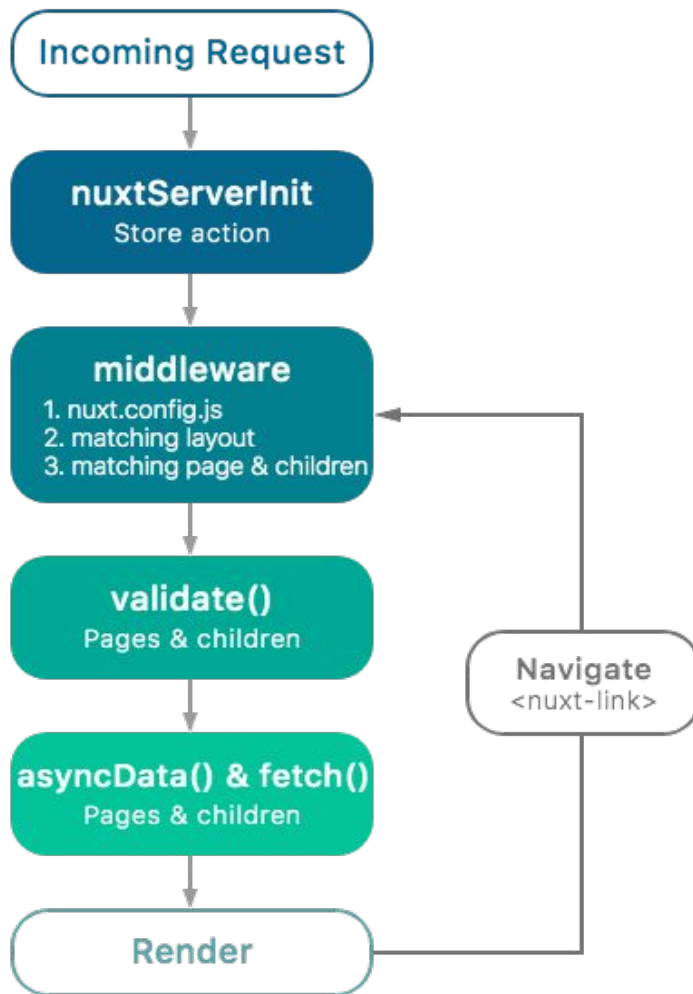
Nuxtjs - Plugins (Feature)

- São usados antes mesmo da inicialização da instância do Vue, para justamente poder ter algo injetado na aplicação.
- Ficam localizados na pasta **/plugins**

Nuxtjs - AsyncData/Fetch (Feature)

- Fazem a requisição no server-side antes de carregar a página
- A página só é carregada após o retorno da requisição

Fluxo de uma requisição Nuxtjs



Nuxtjs - Vantagens

- Grande performance (melhores práticas nodejs).
- Mais de 50 módulos (PWA, SPA e etc...)
- Soluções preparadas para sistemas complexos

Nuxtjs - Módulos de Renderização

- Server Rendered (Universal SSR)
- Single Page Applications (SPA)
- Static Generated (Pre Rendering)

Server Side Rendered (Universal SSR)

- É o modo mais popular do Nuxt
- Preparado para receber o SEO
- Comunicação Server Side/Client Side

Single Page Applications (SPA)

Este é o tradicional *Single Page Application* do Vue.js.

Static Generated (Pre Rendering)

É o melhor dos dois mundos:

- Não há necessidade de um servidor nodejs rodando
- Benefício do SEO configurado no seu site.

Nuxtjs - Resumindo

- Você tem 3 aplicações complexas em um só sistema.
- Mude a medida de sua necessidade.

Nuxt + AWS Lambda Serverless ? Sim!

Primeiro: O que é
Serverless ?

Serverless

Existem:

- Serverless Computing
- Serverless (framework)

Serverless Computing

- Veio para simplificar o deployment do código em produção
- Não precisa provisionar ou gerenciar o servidor
- Aplicativos ganham escalabilidade automaticamente
- Foco no desenvolvimento da aplicação.



Virtual Machines

Virtual Machines run constantly at the provisioned capacity



Serverless

Function-as-a-Service respond and scale to meet demands

Serverless(Framework)

É um framework serverless que ajuda no build, deploy e auto-scaling do seu sistema baseado em funções.

É free e Open Source!

Como Funciona esse Serverless Framework?

Serverless(Framework) - Passos

- **Passo 1:** Você seleciona um Servidor Provider: AWS, Google Cloud, Azure..
- **Passo 2:** Autenticação.
- **Passo 3:** Build & deploy!

AWS Lambda

- Serviço Serverless da AWS
- Modelo do tipo "pay-as-you-go", ou seja, pague o que usar
- Acionado por eventos
- Execução ilimitada de execuções paralelas
- Suporte a várias linguagens, exemplo: java, python, nodejs, e etc...

AWS Lambda - Quanto custa?

Nível gratuito

1 MILHÃO DE SOLICITAÇÕES

por mês

400.000 GB-SEGUNDO

de tempo de computação por mês.

O nível gratuito do Lambda não expira automaticamente ao final do período de 12 meses do nível gratuito da AWS, ele fica disponível para os clientes novos e existentes indefinidamente.

Solicitações

1 MILHÃO DE SOLICITAÇÕES GRÁTIS

O primeiro milhão de eventos no mês é gratuito.

0,20 USD POR 1 MILHÃO DE SOLICITAÇÕES E DEPOIS

0,0000002 USD por solicitação.

Duração

400.000 GB-SEGUNDO POR MÊS GRÁTIS

Os primeiros 400.000 GB-segundo por mês, até 3,2 milhões de segundos de tempo de computação, são grátis.

0,00001667 USD PARA CADA GB-SEGUNDO USADO POSTERIORMENTE

O preço depende da quantidade de memória que você alocar para sua função.

A tabela abaixo mostra os segundos de nível gratuito e o preço aproximado por 100 ms associado a diferentes tamanhos de memória.

Memória (MB)	Segundos de nível gratuito por mês	Preço por 100 ms (USD)
128	3.200.000	0,000000208
192	2.133.333	0,000000313
256	1.600.000	0,000000417
320	1.280.000	0,000000521
384	1.066.667	0,000000625
448	914.286	0,000000729
512	800.000	0,000000834
576	711.111	0,000000938
640	640.000	0,000001042
704	581.818	0,000001146
768	533.333	0,000001250

Como o **Nuxt** se
comunica com o
AWS Lambda ?

Através de uma função **handler**

```
exports.myHandler = function(event, context, callback) {  
  console.log("value1 = " + event.key1);  
  console.log("value2 = " + event.key2);  
  callback(null, "some success message");  
  // or  
  // callback("some error type");  
}
```



Web Browser
script



HTML

Calls
service
methods

Returns
response



AWS SDK
for JavaScript

Invokes
function

Returns
result



Lambda function
(Node.js)



AWS Lambda

Pode se usar em
todas as aplicações ?

Tem alguns
caso que
Não.

Quais os casos que NÃO se deve usar ?

Em todos os casos de conexões persistentes como *websockets, stream, social real-time chat* e etc...

Em todos os outros... **Sim!**



Em resumo...

- Você paga apenas o que usar
- Menos tempo cuidando do servidor
- Expectativa de economia de +50% em comparação

Dúvidas ?

Email: julio.guedes@blueticket.com.br

Linkedin: <https://www.linkedin.com/in/julioguedes/>



Fim!
Obrigado pela
atenção!

