

---

# PROGRAMAÇÃO PHP FRAMEWORK

## LARAVEL



**Professor: Paulo Silva**

# INTRODUÇÃO...

- Breve revisão sobre requisição/resposta HTTP.
- O que é um framework ?
- Introdução ao Laravel
- Ciclo de vida de uma aplicação desenvolvida no Laravel
- Estrutura básica do laravel
- Laravel em um arquivo: **Container**

# Breve revisão sobre REQUISIÇÃO HTTP

```
GET / HTTP/1.1
Host: laravel.com
Connection: keep-alive
Pragma: no-cache
Cache-Control: no-cache
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/58.0.3029.110 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Referer: https://laravel.com/
Accept-Encoding: gzip, deflate, sdch, br
Accept-Language: pt-BR,pt;q=0.8,en-US;q=0.6,en;q=0.4
Cookie: __cfduid=d11d1828a3483ed73d4d4636714794c551496409916; _bsap_daycap=69474%2C69474; _bsap_lifecap=69474%2C69474; __utmt=1; __utma=126154768.2139653786.1496409918.1498749715.1498753181.8; __utmb=126154768.7.10.1498753181; __utmc=126154768; __utmz=126154768.1498753181.8.6.utmcsr=google|utmccn=(organic)|utmcmd=organic|utmctr=(not%20provided)
```

# Breve revisão sobre RESPOSTA HTTP

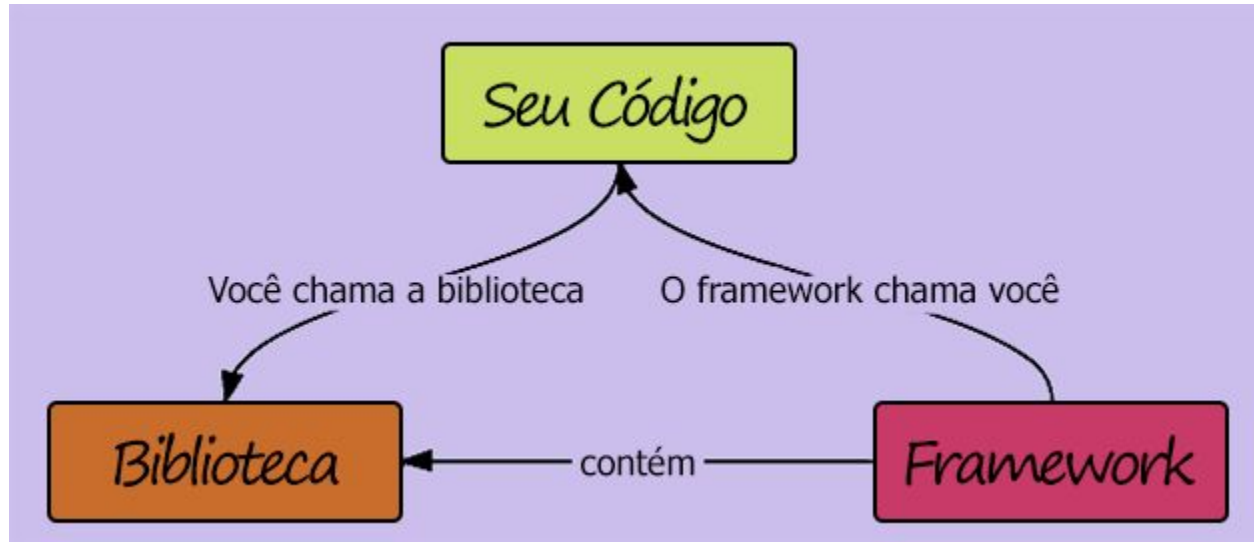
▼ **Response Headers** [view parsed](#)

```
HTTP/1.1 200 OK
Date: Thu, 29 Jun 2017 16:39:42 GMT
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive
Cache-Control: public, max-age=7200
CF-Cache-Status: HIT
Vary: Accept-Encoding
Expires: Thu, 29 Jun 2017 18:39:42 GMT
Server: cloudflare-nginx
CF-RAY: 376a52cb592c87fa-GIG
Content-Encoding: gzip
```

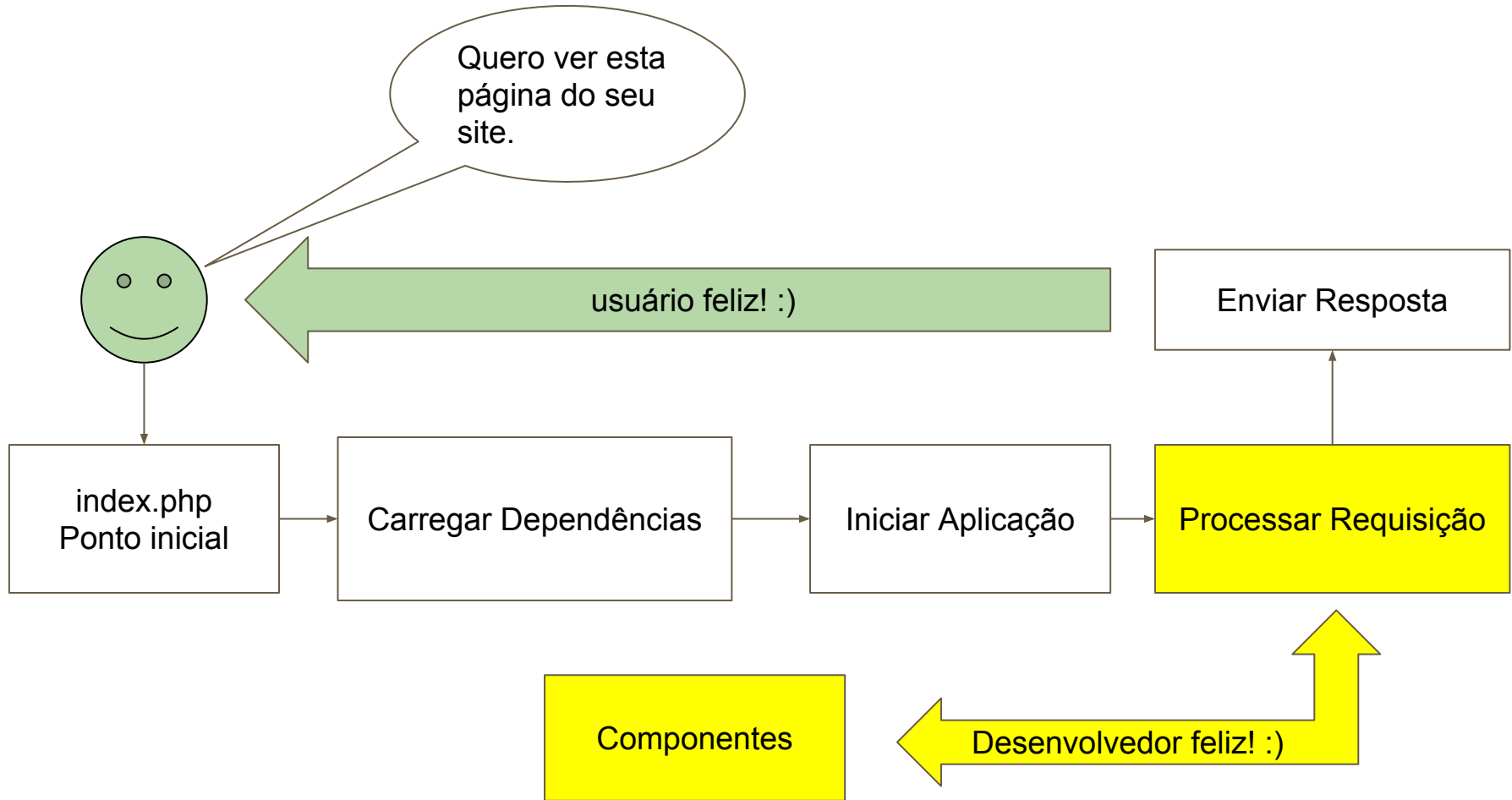
## Definição/Conceito Framework

É uma estrutura, uma fundação para você criar a sua aplicação. Em outras palavras o framework te permite o desenvolvimento rápido de aplicações (RAD), o que faz economizar tempo, ajuda a criar aplicações mais sólidas e seguras além de reduzir a quantidade de código repetido. Os frameworks também permitem que os iniciantes criem aplicações mais estáveis garantindo uma boa relação entre o banco de dados e a camada externa de exibição. Isso tudo te permite gastar mais tempo desenvolvendo a aplicação em si do que repetindo os códigos que todas as aplicações têm.

## Definição/Conceito Framework



# Ciclo de Vida de uma aplicação desenvolvida no Laravel



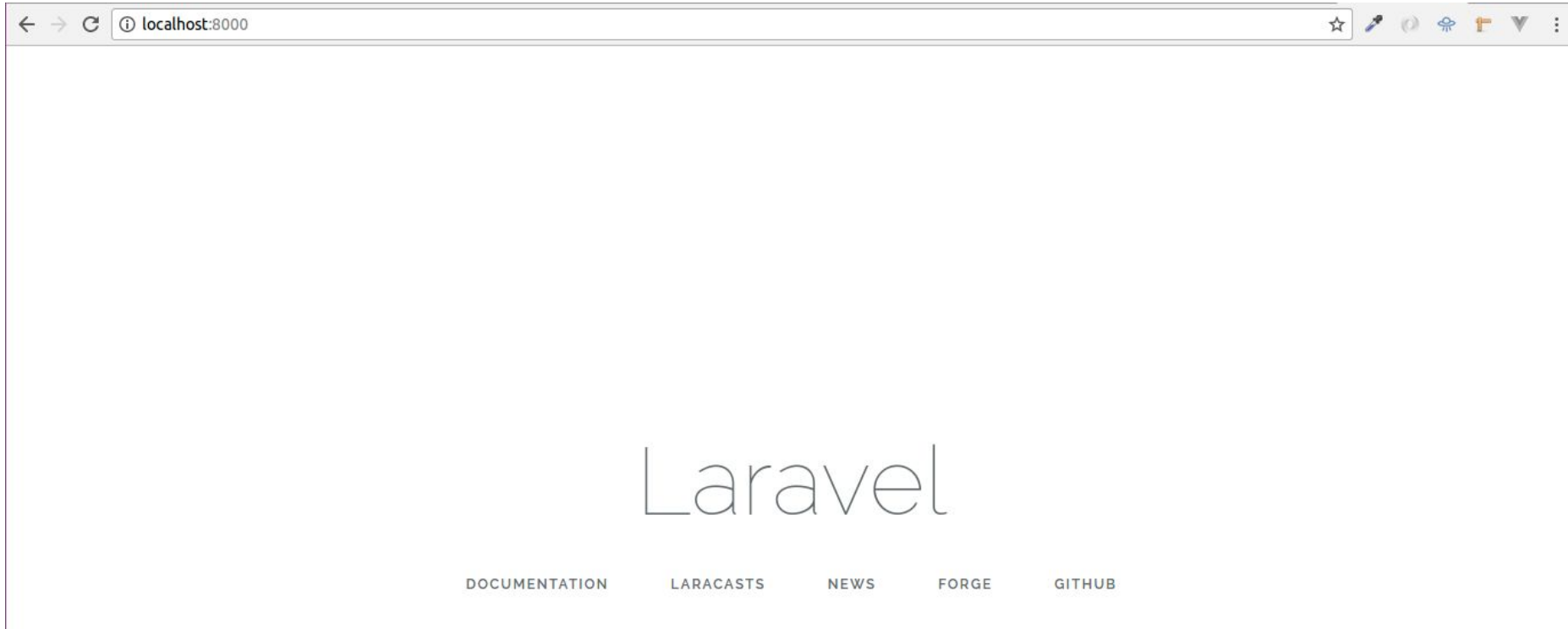
# PRÁTICA - INSTALAÇÃO DO LARAVEL

```
1 git clone https://github.com/laravel/laravel
2 cd laravel
3 cp .env.example .env
4 composer update
5 php artisan key:generate
6 php artisan serv|
```



# PRÁTICA - INSTALAÇÃO DO LARAVEL

- Acesso: **http://localhost:8000**



# Rotas

Todas as rotas do laravel são definidas nos seus arquivos de rotas, que são localizados no diretório `routes`. Esses arquivos são automaticamente carregados pelo framework. O arquivo `routes/web.php` definem as rotas para sua interface web. Essas rotas já são preparadas para tal finalidade. As rotas no arquivo `routes/api.php` são desenhadas para trabalhar com integração entre sistemas, e também são preparadas para esta finalidade.

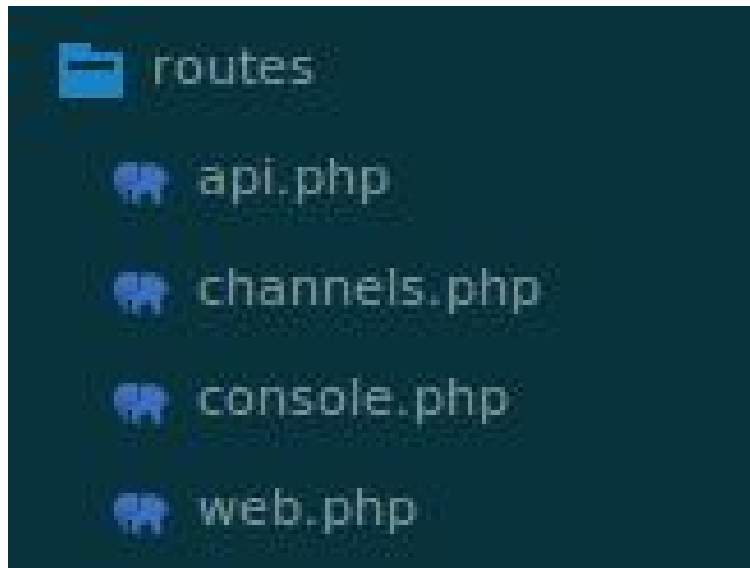
# Rotas

Na maioria das aplicações, você vai começar definindo rotas no arquivo `routes/web.php`. As rotas definidas no arquivo `routes/web.php` podem ser acessadas pela URL no navegador. Por exemplo, você pode acessar essas rotas acessando: to `http://localhost:8000/usuario` no seu navegador.

# Estrutura Básica do Laravel

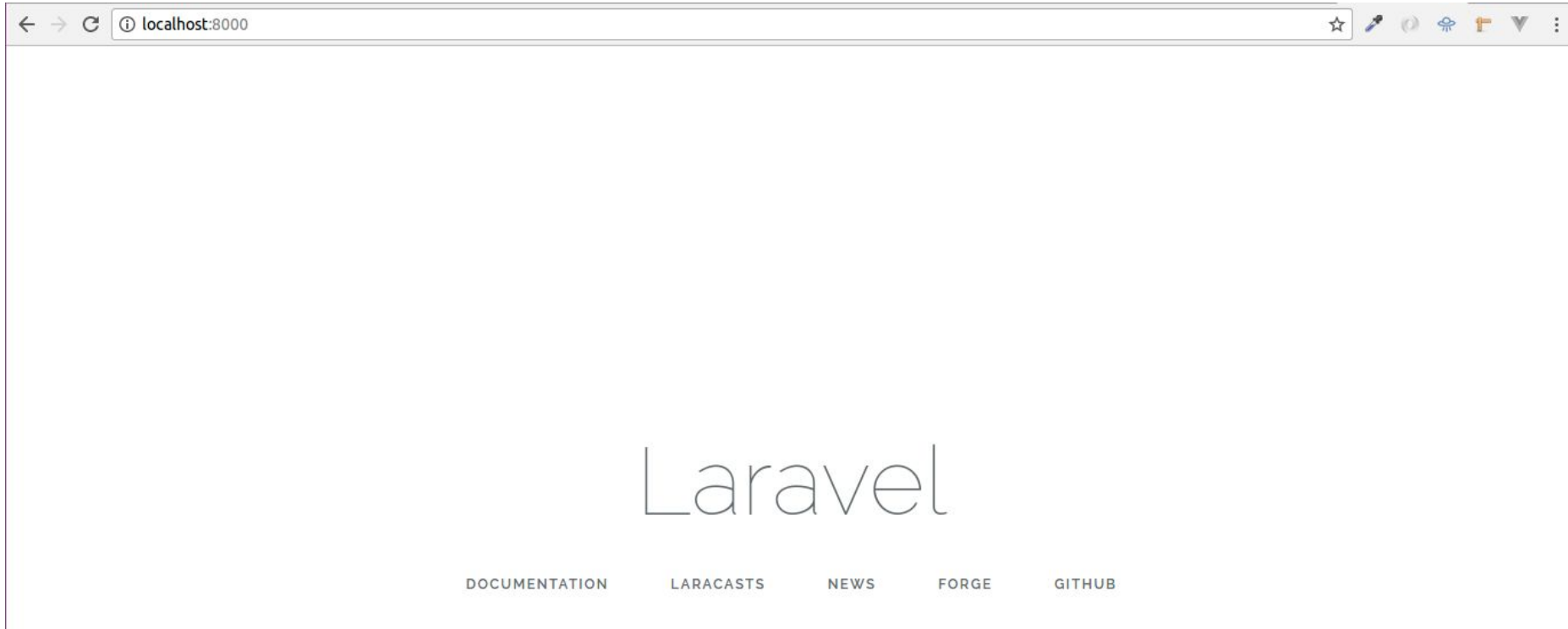
## Diretório de Rotas

- routes/



# PRÁTICA - INSTALAÇÃO DO LARAVEL

Na rota `/` o laravel exibe esta página:



# Por que o laravel exibe essa página ?

- routes/web.php
- resources/views/welcome.blade.php

```
14 Route::get('/', function () {  
15     return view('welcome');  
16 });
```

# Como criar rotas personalizadas ?

- routes/web.php

```
Route::get('/usuarios/{usuario}', function($usuario) {  
    echo "Bem vindo, {$usuario}";  
});
```

# Views

Blade é um simples, mas poderoso sistema de template criado pelo Laravel. O laravel não restringe a view apenas para Blade, pode-se usar PHP puro. Na verdade, o Blade é sempre compilado para PHP puro, e é feito o cache até que eles sejam modificado. Isso significa que o Blade não sobrecarrega a aplicação. Todo arquivo de view no Blade usa o padrão `.blade.php` como extensão, e são normalmente armazenados no diretório `resources/views`

```
14 Route::get('/', function () {  
15     return view('welcome', ['name' => 'Paulo Silva']);  
16 });  
17  
18
```



# Crie suas rotas, para entregar HOJE! VALE PONTO!!

- Crie uma rota para a página de blog
- Crie uma rota para acessar a página administrativa
- Crie uma rota para listar os usuários: “José, Maria, Pedro, João” em uma tabela HTML
- Crie uma rota que leia 5 parâmetros, e tire a média dos valores
- Criei uma rota que leia 2 nomes, e mostre se os nomes são iguais
  
- Criei uma rota que recebe 3 nomes, e exibe em uma VIEW
- Crie uma rota que exibe a soma de 2 números em uma VIEW

# Views

```
<html>

  <head>
    <title>App Name - @yield('title')</title>
  </head>
  <body>
    @section('sidebar')
      This is the master sidebar.
    @show

    <div class="container">
      @yield('content')
    </div>
  </body>
</html>
```

# Views

```
@extends('layouts.app')

@section('title', 'Page Title')

@section('sidebar')
    @parent

    <p>This is appended to the master sidebar.</p>
@endsection

@section('content')
    <p>This is my body content.</p>
@endsection
```

# Atividade Views VALE PONTO!!

Crie uma view que receba uma lista de usuários ( Thiago, Henrique, Pedro e João ), em uma tabela HTML

Crie uma view que define um template, e crie outra view que estenda este template, e adicione conteúdo em uma **@section**

Crie um view com o documento básico do html <html>...<head>... </head> (...) </html>, e crie parciais para cada área do documento básico ( CABEÇALHO, RODAPÉ E CONTEÚDO)