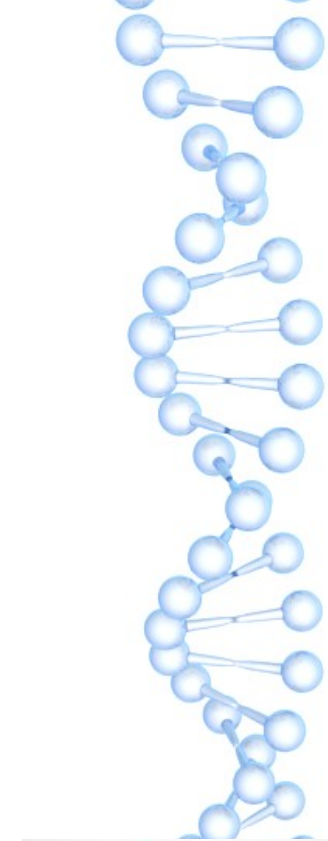


Marcos Rodrigo Momo, M.Sc.
e-mail: marcos.momo@ifsc.edu.br

Programação para Internet II

INSTITUTO FEDERAL
Santa Catarina
Câmpus Gaspar

Gaspar, maio 2021.

Roteiro

- Apresentação
- Plano de ensino
- Cronograma das aulas
- Introdução framework Laravel
- Atividade práticas

INSTITUTO FEDERAL
Santa Catarina
Câmpus Gaspar

Apresentação

- ✓1994 - FURB
- ✓2000 – BCC
- ✓2005 – Especialização TI
- ✓2015 – Mestrado em Engenharia Ambiental
- ✓2018 – Doutorando na UFSC

- ✓Professor desde 2014 (IFC, FURB e CEDUP)
  - ✓Linguagem de programação
  - ✓Sistemas operacionais
  - ✓Programação orientadas a objetos
  - ✓Sistemas distribuídos
  - ✓SOA

Apresentação

- ✓ 2009 Membro do grupo CEOPS (Sala T-106)
  - ✓ Sistemas distribuídos
  - ✓ Redes neurais (modelagem hidrológica)
  - ✓ Mapeamento de áreas de inundação
  - ✓ Coorientação de TCCs
    - ✓ Sistemas especialistas aplicado ao monitoramento do sistema de alerta
    - ✓ Previsão hidrológicas em tempo atual
  - ✓ Coordenação iniciação científica - UNIFEBE
    - ✓ RNA aplicado à modelagem hidrológica

INSTITUTO FEDERAL
Santa Catarina
Câmpus Gaspar

Apresentação

- Contatos:
  - marcos.momo@ifsc.edu.br
- Horário de atendimento
  - Quinta-feira: das 13:30 às 15:30 horas
  - Local: remotamente, agendar por e-mail para a criação de uma sala no google meet

INSTITUTO FEDERAL
Santa Catarina
Câmpus Gaspar

Plano de ensino
Tópicos

- Introdução ao Framework Laravel
- Conceitos de Rotas/Controladores/Views do Laravel
- Conceitos de MVC
- Validação de formulários
- Modelo ORM e persistência
- Aplicação de programação orientada a objetos ao desenvolvimento web

INSTITUTO FEDERAL
Santa Catarina
Câmpus Gaspar

Plano de ensino
Objetivos

- Analisar e projetar sistemas computacionais seguindo as metodologias adequadas e as recomendações de qualidade e de segurança.
- Implementar sistemas computacionais seguindo as especificações e paradigmas da lógica e das linguagens de programação.
- Implantar, manter e prestar suporte a sistemas computacionais, visando o seu uso de forma alinhada e atualizada com o seu propósito.
- **Avaliar e testar sistemas computacionais de modo a garantir que foi desenvolvido de maneira apropriada e consistente, correspondendo aos requisitos estabelecidos e que apresente comportamento esperado.**

- **Desenvolver aplicações dinâmicas para internet, adotando diferentes tecnologias.**

INSTITUTO FEDERAL
Santa Catarina
Câmpus Gaspar

Plano de ensino
Metodologia ANPs

- Sempre com uso de ferramentas
- Aulas expositiva e dialogada
- Conceituação teórica
- Aplicação prática
- Atividades parciais

INSTITUTO FEDERAL
Santa Catarina
Câmpus Gaspar

Plano de ensino
Provas e trabalhos

- Prova prática (individual)
- Atividades práticas (individual)
- Trabalhos intermediários com defesa (individual)
- Trabalho final com defesa (individual)

INSTITUTO FEDERAL
Santa Catarina
Câmpus Gaspar

Plano de ensino
Avaliação

- A avaliação será composto por:

- 1) Trabalhos parciais (TP) -> Teremos vários TPs
- 2) Estudo de casos (EC) -> Teremos dois ECs
- 3) Trabalho final (TF) -> Teremos um TF

- A média final será assim calculada:

MF = (Media dos TPs * 0.10) + (EC1* 0,15) + (EC2 * 0,15) + (TF * 0.60)

INSTITUTO FEDERAL
Santa Catarina
Câmpus Gaspar

Plano de ensino

- Em caso de verificação de cópia, a nota da atividade em questão será **ZERADA**, tanto para o aluno que copiou, quanto para aquele que deixou copiar.

- Requisitos para a aprovação:
  - Média >= 6.0
  - Frequência >= 75%

INSTITUTO FEDERAL
Santa Catarina
Câmpus Gaspar

Plano de ensino
Recuperação paralela

- Atividades extra classe
  - Correção de trabalhos e provas em laboratório
  - Disponibilização através do ambiente de aprendizagem, lista de exercícios e atividades práticas complementares sobre o conteúdo a ser recuperado
- Avaliação baseada na aplicação de prova e/ou trabalho complementar (final do semestre)

Calendário do bloco 1
Carga horária: 80 horas

	MÓDULO 1: 26/04 – 21/05 (4 semanas)					
	2ª feira	3ª feira	4ª feira	5ª feira	6ª feira	sábado
07:20:00						prog.internet II
08:15:00						prog.internet II
09:10:00						teste software
10:25:00						teste software
11:20:00						
13:30:00						
18:30:00		prog.internet II		prog.internet II	teste software	
19:25:00		prog.internet II		prog.internet II	teste software	
20:40:00		prog.internet II		prog.internet II	teste software	
21:35:00		prog.internet II		prog.internet II	teste software	

Disponível em: https://www.ifsc.edu.br/web/campus-gaspar/horarios-ensalamento

Calendário do bloco 2
Carga horária: 80 horas

MÓDULO 2: 24/05 – 25/06 (5 semanas)						
2ª feira	3ª feira	4ª feira	5ª feira	6ª feira	sábado	
						07:20:00
						08:15:00
					prog.internet II	09:10:00
					prog.internet II	10:25:00
						11:20:00
						13:30:00
teste de software	prog.internet II	prog.internet II				18:30:00
teste de software	prog.internet II	prog.internet II				19:25:00
teste de software	prog.internet II	prog.internet II				20:40:00
teste de software	prog.internet II	prog.internet II				21:35:00

Disponível em: https://www.ifsc.edu.br/web/campus-gaspar/horarios-ensalamento

INSTITUTO FEDERAL
Santa Catarina
Câmpus Gaspar

Calendário dos blocos 1 e 2
Carga horária por semana 10 horas
8 semanas (8x10) = Total 80 horas

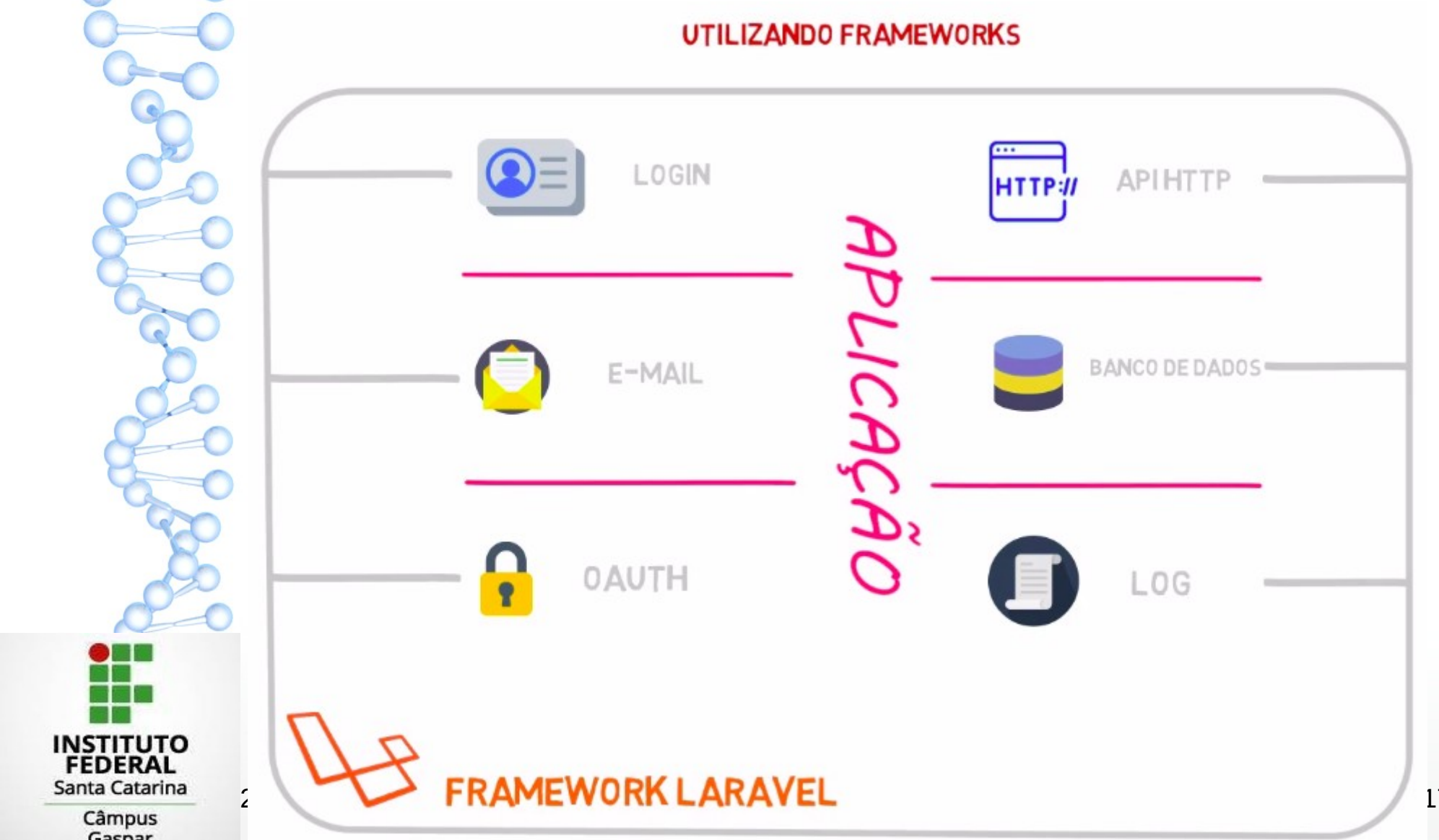
ADS 4				26/4 a 30/4	3/5 a 7/5	10/5 a 15/5	17/5 a 21/5	24/5 a 29/5	31/5 a 4/6	7/6 a 12/6	14/6 a 18/6	21/6 a 25/6	28/6 a 3/7	5/7 a 9/7	26/7 a 31/7	2/8 a 6/8	9/8 a 13/8	16/8 a 21/8	23/8 a 27/8	30/8 a 3/9	6/9 a 11/9	13/9 a 18/9
	UCs	CH	Docente	Módulo 1				Módulo 2					Módulo 3				Módulo 4					finalização e conselhos de classe
1	Teste de Software	40	MARCOS RODRIGO MOMO/	6	6	6	6	4		4	4	4										
2	Análise de Sistemas II	80	ROGÉRIO ANTONIO SCHMITT/	8	8	8	8	10	8	10	10	10										
3	Práticas em Desenvolvimento de Sistemas I	80	TAMER CAVALCANTE/										10	10	10	10	10	10	10	10		
4	Sistemas Operacionais	40	ANDREU CARMINATI/														10	10	10	10		
5	Metodologia de Pesquisa	40	LEONIDAS de MELLO Jr./	10				0	0	0	0	0		10	10	10	0	0	0	0	0	
6	Programação para Internet II	80	MARCOS RODRIGO MOMO/		10	10	10	10	10	10	10	10										
7	Gerência de Projetos	40	THIAGO PAES/										8	4	4	4	4	4	4	4	4	
	TOTAL	400	TOTAL	24	24	24	24	24	18	24	24	24	18	24	24	24	24	24	24	24	4	

INSTITUTO FEDERAL
Santa Catarina
Câmpus Gaspar

Configuração do ambiente para desenvolvimento Windows

- IDE Visual Studio Code
- Pacote XAMPP – Apache, MySql, php, e phpMyAdmin
- Composer gerenciador de pacotes do PHP
  - composer –version (para checar a instalação)
- NPM faz parte do pacote node
  - npm –v (para checar a instalação)
- Fazer download do Laravel installer using Composer: -através deste installer podemos criar os projetos e artefatos no Laravel de forma mais simples
  - composer global require "laravel/installer"
  - Ver https://laravel.com/docs/5.6/installation

INSTITUTO FEDERAL
Santa Catarina
Câmpus Gaspar


UTILIZANDO FRAMEWORKS
LOGIN
E-MAIL
OAUTH
APLICAÇÃO
API HTTP
BANCO DE DADOS
LOG
FRAMEWORK LARAVEL

Uso de Framework

- Vantagem
  - O desenvolvedor se preocupa apenas com o desenvolvimento da sua aplicação
  - Muitos recursos (genéricos) já estão implementados e testados
  - Oferecem credibilidade quanto à sua continuidade, suporte, documentação etc.
- Desvantagem
  - O modelo de arquitetura do sistema é fortemente acoplada
  - O sistema fica dependente do framework e suas tecnologias adjacentes

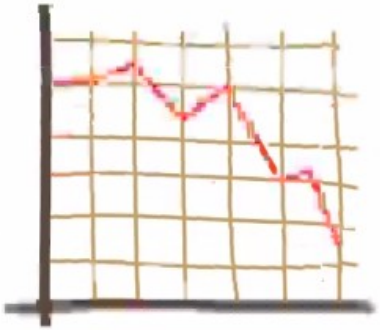
LARAVEL

Implementação rápida e código limpo

Facilita a manutenção e a codificação em times

Framework sedimentado com mais de 7 anos de mercado

Lider de mercado e milhares de sistemas desenvolvidos



TIME TO MARKET

Framework

- É uma estrutura conceitual e tecnológica de suporte definido com módulos de software concretos, que servem de base para a organização e desenvolvimento de software
- Compreende de um conjunto de classes implementadas em uma linguagem de programação específica, usadas para auxiliar o desenvolvimento de software
- Provem funcionalidades genéricas, ou seja, atua onde há funcionalidades em comum a várias aplicações
  - Por exemplo: Interfaces gráficas, persistência, transação, segurança, autenticação etc…

INSTITUTO FEDERAL
Santa Catarina
Câmpus Gaspar

Razões para utilizar um Framework

- Evitar escrever código repetitivo
  - Partes comuns de projetos distintos
  - Exemplo: acesso banco de dados, validação de formulários, segurança
- Utilizar boas práticas
  - Os frameworks geralmente estão baseados em padrões de desenvolvimento
  - Por exemplo: modelo MVC (model, view, controller)
  - Estes padrões nos ajudam a separar os dados e a lógica do negócio da interface

Razões para utilizar um Framework

- Possibilitar desenvolver tecnologias avançadas (atuais)
  - Comunidade sempre em desenvolvimento (ativa)
  - Fazer sozinho poderia levar mais tempo
- Desenvolvimento mais rápido
  - Viabiliza desenvolver rápido, limpo, seguro, modular, suporte

Qual Framework utilizar

- Devemos ter claro quais são as características e necessidades no nosso projeto para escolher qual tecnologia utilizar:
  - Desenvolvimento web
  - Tem suporte a comunidade
  - Tem documentação
  - Simplicidade e potência
  - Arquitetura MVC
  - Reutilização
  - Segurança
  - Etc..
- Aplicação, tamanho, quantidade de requisitos grau de severidade (segurança, tempo)

V • D • E	Framework para aplicações Web [Esconder]
ASP.NET	ASP.NET Dynamic Data • ASP.NET MVC • ASP.NET Web Forms • BFC • DotNetNuke • MonoRail • OpenRasta • Umbraco
ColdFusion	CFWheels • ColdBox Platform • ColdSpring • Fusebox • Mach-II • Model-Glue
Common Lisp	CL-HTTP • UnCommon Web • Weblocks
C++	CppCMS • Wt
Haskell	Happstack • Yesod • Snap
Java	AppFuse • Flexive • Grails • GWT • ICEfaces • ItsNat • JavaServer Faces • Jspx • Juzu • Makumba • OpenXava • Play • Reasonable Server Faces • Remote Application Platform • RIFE • Seam • Spring • Stripes • Struts • Tapestry • Vaadin • WebWork • Wicket • WaveMaker • ZK
JavaScript	Ample SDK • AngularJS • Backbone.js • Chaplin.js • Closure • Dojo Toolkit • Ember.js • Ext JS • jQuery • Meteor • Prototype • React • Rico • script.aculo.us • Sencha Touch • SproutCore • Wakanda
Perl	Catalyst • Dancer • Mason • Maypole • Mojolicious • WebGUI
PHP	AppFlower • CakePHP • CodeIgniter • Drupal • eZ Publish • Fat-Free • FuelPHP • Horde • Joomla! • Kohana • Laravel • Lithium • Midgard • MODX • Nette Framework • Phalcon • PRADO • Qcodo • Seagull • SilverStripe • Symfony • TYPO3 • WordPress • Xaraya • XOOPS • Yii • Zend Framework
Python	BlueBream • CherryPy • Django • Flask • Grok • Nevow • Pyjamas • Pylons • Pyramid • Quixote • TACTIC • Tornado • TurboGears • web2py • Webware • Zope 2
Ruby	Camping • Hobo • Merb • Padrino • Ramaze • RailsBricks • Ruby on Rails • Sinatra • Hanami
Scala	Lift • Play • Scalatra
Smalltalk	AIDA/Web • Seaside
Outras Linguagens	Application Express (PL/SQL) • Grails (Groovy) • Kepler (Lua) • OpenACS (Tcl) • SproutCore (JavaScript/Ruby) • SymbolicWeb (Clojure) • Yaws (Erlang)

Introdução ao Framework Laravel

- Laravel é um Framework PHP utilizado para o desenvolvimento web
- Utiliza a arquitetura MVC
- Possibilita desenvolver aplicações seguras e performáticas de forma rápida, com código limpo e simples
- Incentiva o uso de boas práticas de programação
  - Exemplo: padrão PSR-2 como guia para estilo de escrita do código

INSTITUTO FEDERAL
Santa Catarina
Câmpus Gaspar

Introdução ao Framework Laravel

- Para a criação de interface gráfica, o Laravel utiliza uma Engine de template chamada Blade
- Oferece uma gama de ferramentas que ajudam a criar interfaces bonitas e funcionais de forma rápida e evitar a duplicação de código
- Para se comunicar com banco de dados utiliza uma implementação simples do ActiveRecord chamada de Eloquent ORM, que é uma ferramenta que traz várias funcionalidades para facilitar a inserção, atualização, busca e exclusão de registros
- Com configuração simples e pouco código podemos configurar a conexão com Banco de Dados e trabalhar com ele

INSTITUTO FEDERAL
Santa Catarina
Câmpus Gaspar

Criando um projeto Visual Studio Code

- Criar uma pasta com o seu nome
  - Criar uma aplicação laravel chamada aula1

    laravel new aula1
  - Criar web server para rodar a aplicação criada

    php artisan serve
  - Acessar aplicação

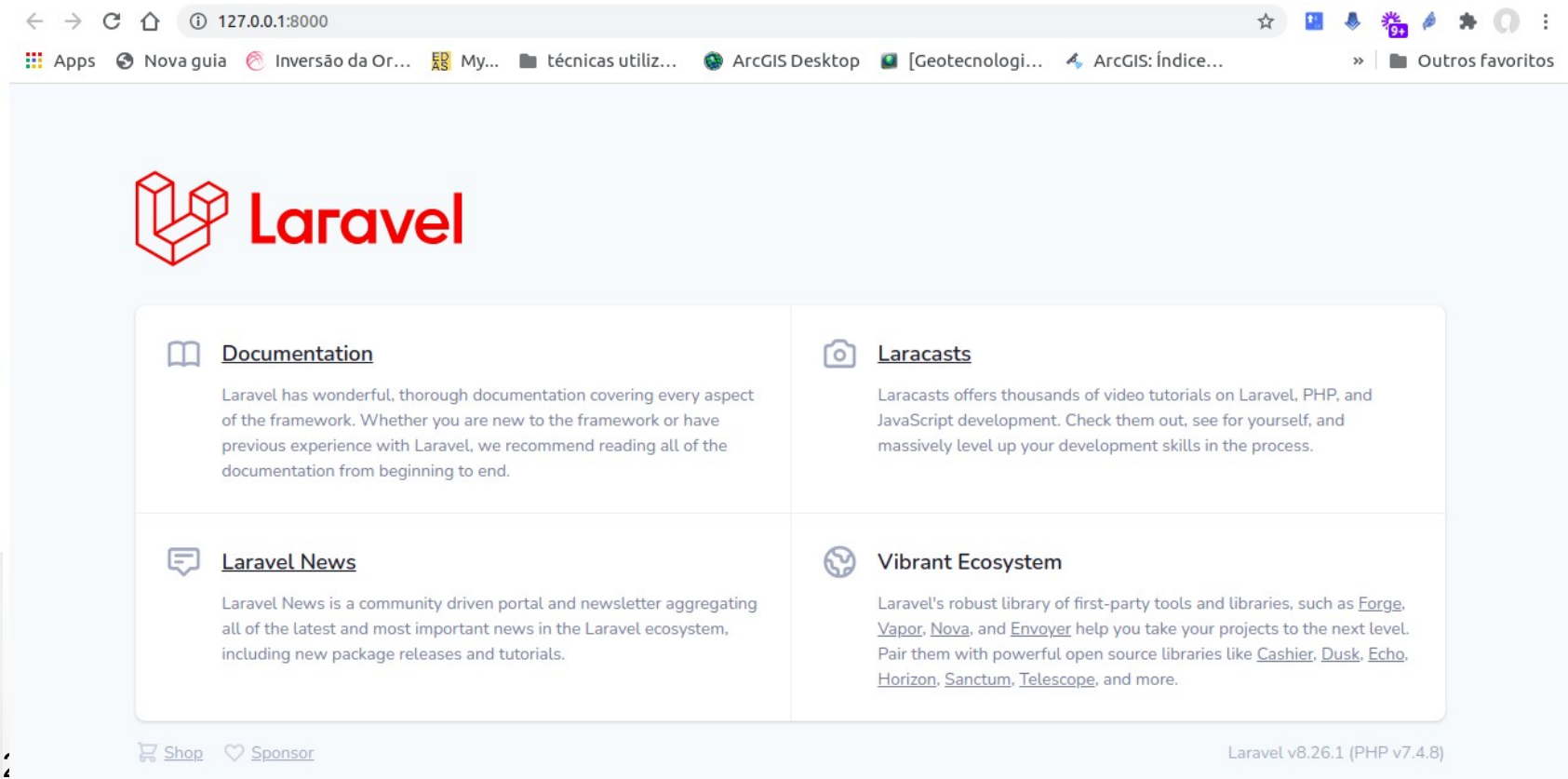
    localhost:8000

Estrutura de pastas

- Após criar o projeto, várias pastas são criadas:
  - app: aqui fica o código principal da aplicação, seus modelos e controllers
  - config: nessa pasta fica toda a configuração, como, banco de dados, e-mails, entre outros
  - public: geralmente aqui fica seu arquivo index.php, as imagens, css e o js
  - vendor: possui o source code do laravel, plugins e dependências. Tudo que for usado de terceiros, como, frameworks e bibliotecas devem ficar aqui
  - Descrição completa de todas, acessar a documentação:
  - https://laravel.com/docs/5.4/structure

Editando página inicial

- Pasta resources/views “welcome.blade.php”



INSTITUTO FEDERAL
Santa Catarina
Câmpus Gaspar

Rotas no Laravel

- Rotas é o caminho que inserimos na barra do navegador
  - Por exemplo: localhost:8000/
    - A “/” neste caso é o nome da rota.
  - Na pasta routes/web.php, eu configuro a rota chamada “/”

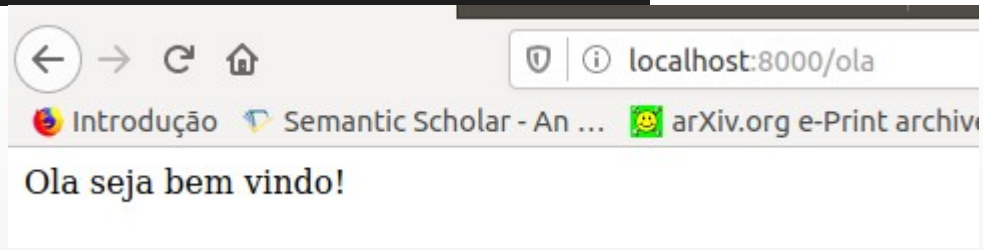
```
/*Já tem um rota cadastrada chamada '/' */
Route::get('/', function () {
    return view('welcome');
});
```

  - Neste caso, a call back da função vai carregar a view chamada “welcome” o arquivo Blade da pasta resources/welcome.blade.php

Configurando rotas

- Para configurar uma nova rota, utilizo a classe de configuração de rotas, chamada routes.
- A partir da classe route, posso utilizar métodos http estático tais como: get, post, put, delete, option etc…
- Por exemplo:

```
/*Configurando uma nova rota */
Route::get('/ola', function(){
    return "Ola seja bem vindo!";
});
```



Atividade

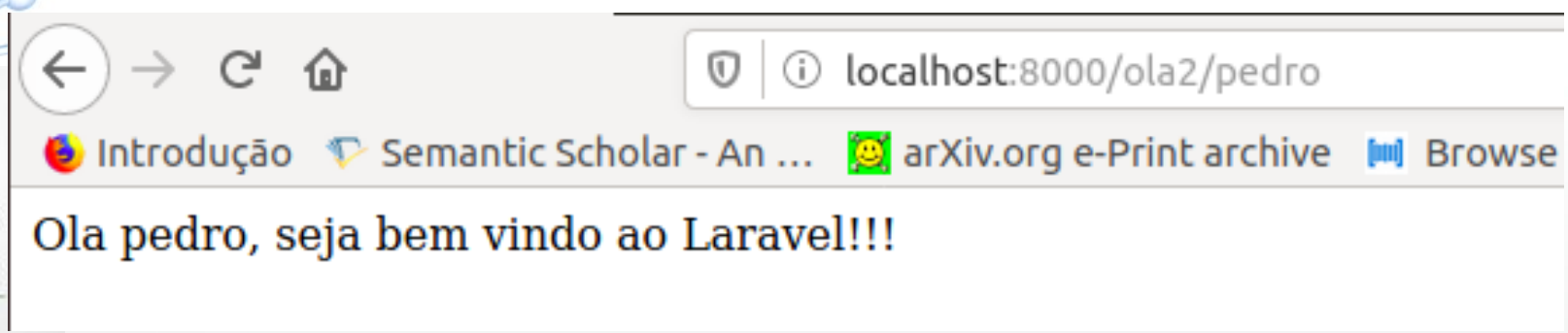
- Crie uma nova rota com a mensagem “Hello World!!!”
- Liste as rotas criadas
  - Listamos as rotas através do comando no terminar

    php artisan list:route

Rotas com parâmetros

- Podemos passar parâmetros às rotas
- Por exemplo:

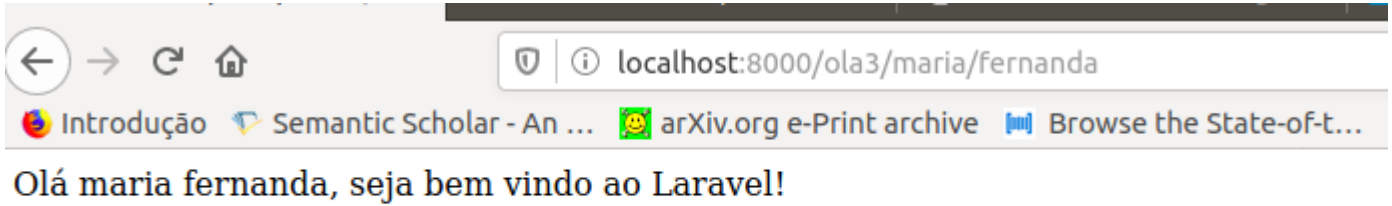
```
/* Criando rota com parâmetros */
Route::get('ola2/{nome}', function($nome){
    echo "Ola ".$nome. ", seja bem vindo ao Laravel!!!";
});
```



INSTITUTO FEDERAL
Santa Catarina
Câmpus Gaspar

Atividade

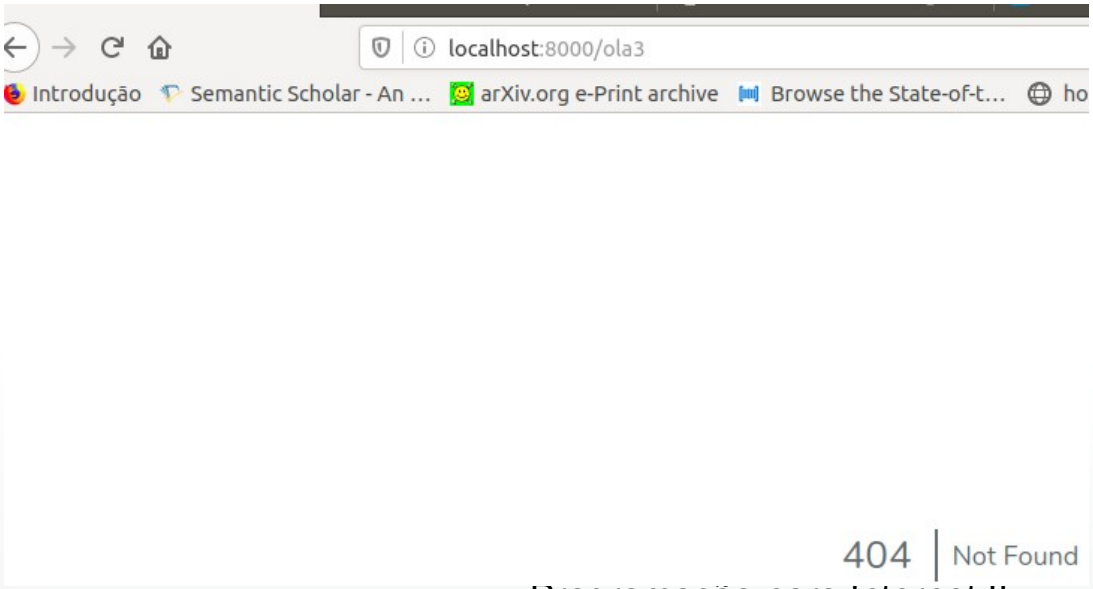
- Criar uma nova rota com dois parâmetros
- Por exemplo, com a saída



- Verifique as rotas criadas
  - php artisan route:list

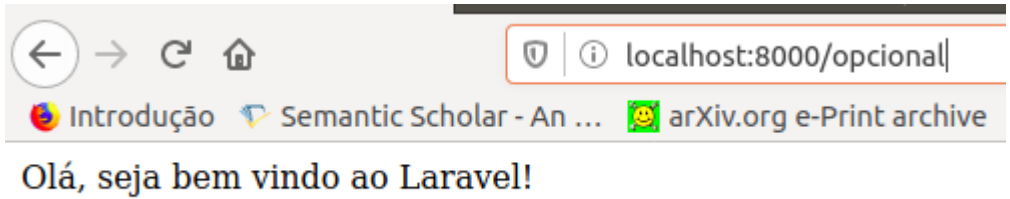
Rotas com parâmetros

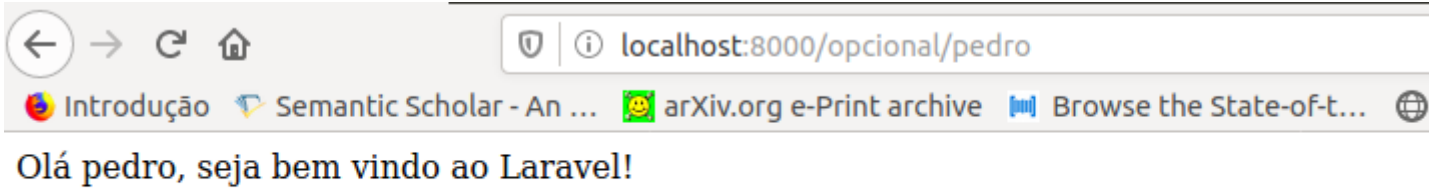
- Caso os parâmetros não sejam informados pelo usuário, o que vai acontecer.



Rotas com parâmetro opcional

```php
/*Criando rota com parâmetro opcional */
Route::get('opcional/{nome?}', function($nome=null){
    if (isset ($nome))
        echo "Olá $nome, seja bem vindo ao Laravel!";
    else
        echo "Olá, seja bem vindo ao Laravel!";
});
```





Atividade

- Crie um projeto Laravel:
  - Altere a página inicial
  - Crie rotas que receba parâmetros obrigatórios
  - Crie rotas que recebe parâmetros opcionais, caso o usuário não informar o parâmetro, deve ser informado uma mensagem na tela “usuário não informou parâmetro”

Referências

- Laravel. https://laravel.com/docs/6.x

- Estrutura de pastas. https://laravel.com/docs/5.4/structure