



UNIVERSIDADE  
Estadual de LONDRINA

ANDERSON YOSHIO MAKINO

**COMPARAÇÃO DE FRAMEWORKS DE DESENVOLVIMENTO EM PHP PARA  
SISTEMAS WEB**

LONDRINA - PR

2022



ANDERSON YOSHIO MAKINO

**COMPARAÇÃO DE FRAMEWORKS DE DESENVOLVIMENTO EM PHP PARA  
SISTEMAS WEB**

Trabalho de Conclusão de Curso apresentado ao Curso de Bacharelado em Ciência da Computação do Departamento de Computação da Universidade Estadual de Londrina, como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Jandira Guenka Palma

**LONDRINA - PR**

**2022**

A ficha catalográfica deverá ser solicitada na biblioteca central  
(obrigatório apenas para Dissertação de Mestrado)

**B222d** Sobrenome, Nome do autor.  
Gerência de redes – protocolo SNMP / nome e sobrenome do aluno. – Londrina, 2010.  
98 f. : il.  
Orientador: nome sobrenome do orientador.  
Dissertação (Mestrado em ) Universidade Estadual de Londrina, Centro de  
Ciências Exatas, Programa de Pós-Graduação em xxxxx, 2010.  
Inclui bibliografia.  
I. Gerencia de redes – Teses. 2. SNMP. 3 NMS – Teses. 3. Assunto 3 – Teses. I.  
Sobrenome, Nome do orientador. II. Universidade Estadual de Londrina. Centro de  
Ciências Exatas. III. Título.

(ATENCAO PARA MONOGRAFIA DE ESPECIALIZAÇÃO NÃO PRECISA)

CDU 641:579

ANDERSON YOSHIO MAKINO

**COMPARAÇÃO DE FRAMEWORKS DE DESENVOLVIMENTO EM PHP PARA  
SISTEMAS WEB**

Trabalho de Conclusão de Curso apresentado ao Curso de Bacharelado em Ciência da Computação do Departamento de Computação da Universidade Estadual de Londrina, como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação.

**BANCA EXAMINADORA**

---

Prof. Dr. Jandira Guenka Palma  
Universidade Estadual de Londrina

---

Prof(a). Dr(a). Segundo Membro da Banca  
Universidade Segundo Membro da Banca

---

Prof. Dr. Terceiro Membro da Banca  
Universidade Terceiro Membro da Banca

---

Prof. Dr. Quarto Membro da Banca  
Universidade Quarto Membro da Banca

Londrina-PR, \_\_\_\_ de \_\_\_\_\_ de 2022.

*Este trabalho é dedicado a todos que me ajudaram ao longo desta caminhada.*

## **AGRADECIMENTOS**

Agradeço a Universidade Estadual de Londrina pela oportunidade de fazer o curso de Ciência da Computação. Agradeço aos professores que fizeram parte da minha formação. Agradeço à professora orientadora Dr. Jandira Guenka Palma por me guiar nesta etapa final do curso. Agradeço aos meus familiares e amigos, pelo apoio e incentivo durante os anos de graduação.

*“Não vos amoldeis às estruturas deste mundo, mas transformai-vos pela renovação da mente, a fim de distinguir qual é a vontade de Deus: o que é bom, o que Lhe é agradável, o que é perfeito. (Bíblia Sagrada, Romanos 12, 2)*



MAKINO, ANDERSON. **Comparação de frameworks de desenvolvimento em PHP para sistemas Web**. 45p. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Universidade Estadual de Londrina, Londrina-PR, 2022.

## RESUMO

As frameworks voltadas ao desenvolvimento de sistemas Web estão presentes no cotidiano dos desenvolvedores devido à crescente necessidade de sistemas Web complexos e bem elaborados, além das facilidades que essas ferramentas trazem ao desenvolvedor. No entanto, devido à existência de diversas frameworks, as empresas e os desenvolvedores encontram dificuldades em identificar a framework que melhor atende às necessidades de seu projeto. Assim, este trabalho busca identificar quais os mais indicados para a produção de um sistema Web utilizando a linguagem PHP (PHP: *Hypertext Preprocessor*), considerando que o PHP é uma linguagem amplamente empregada para desenvolvimento Web. Realizou a identificação das características que são mais relevantes em uma framework por meio de uma pesquisa bibliográfica e depois estudou os diversos aspectos apresentados em cada framework selecionado. Como resultado, foram identificadas as principais características das frameworks para desenvolvimento de sistemas web em PHP, tais como documentação, popularidade e suporte, performance, sustentabilidade, pacotes e componentes e segurança. A partir do estudo realizado, foi feita a tabulação dessas características de cada framework e sua comparação. Essa tabulação e as explicações circunstanciais são ferramentas importantes para auxiliar empresas e desenvolvedores na escolha da framework mais adequada para o desenvolvimento de sistemas web em PHP de acordo com as suas necessidades.

**Palavras-chave:** Framework. Web. PHP. Comparação.

MAKINO, ANDERSON. **Comparison of PHP development frameworks for web systems**. 45p. Final Project (Bachelor of Science in Computer Science) – State University of Londrina, Londrina-PR, 2022.

### ABSTRACT

Frameworks aimed at web development are part of developers' daily lives due to the growing need for complex and well-crafted web systems, as well as the conveniences these tools bring to developers. However, due to the existence of various frameworks, companies and developers face difficulties in identifying the framework that best meets the needs of their project. Thus, this work aims to identify the most suitable frameworks for developing a web system using the PHP (PHP: Hypertext Preprocessor) language, considering that PHP is widely employed for web development. The identification of the most relevant characteristics in a framework was carried out through a bibliographic research, followed by studying the various aspects presented in each selected framework. As a result, the main characteristics of frameworks for PHP web development were identified, such as documentation, popularity and support, performance, sustainability, packages and components, and security. Based on the conducted study, the tabulation of these framework characteristics and their comparison was performed. This tabulation and the circumstantial explanations are important tools to assist companies and developers in choosing the most suitable framework for web development in PHP according to their needs.

**Keywords:** Framework. Web. PHP. Comparison.

## **LISTA DE ILUSTRAÇÕES**

Figura 1 – Benchmark de JSON serialization.	33
Figura 2 – Benchmark de Single query.	34
Figura 3 – Benchmark de Multiple queries.	34
Figura 4 – Benchmark de Fortunes.	35

**LISTA DE TABELAS**

Tabela 1 – Tabela da proposta das frameworks.	29
Tabela 2 – Tabela da documentação das frameworks.	30
Tabela 3 – Tabela da popularidade e suporte das frameworks.	32
Tabela 4 – Tabela da sustentabilidade das frameworks.	36
Tabela 5 – Tabela de pacotes e componentes das frameworks.	37
Tabela 6 – Tabela da segurança das frameworks.	39

**LISTA DE ABREVIATURAS E SIGLAS**

Web	Associação Brasileira de Normas Técnicas
PHP	PHP: Hypertext Preprocessor
CGI	Common Gateway Interface
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
AJAX	Asynchronous JavaScript And XML
SPA	Single Page Application
ORM	Object-Relational Mapping
MVC	Model View Controller
MIT	Massachusetts Institute of Technology
BCIT	British Columbia Institute of Technology
HMVC	Hierarchical Model View Controller
JSON	JavaScript Object Notation
CSRF	Cross-Site Request Forgery
SQL	Structured Query Language
XSS	Cross-Site Scripting

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>15</b>
<b>2</b>	<b>REVISÃO BIBLIOGRÁFICA</b>	<b>17</b>
2.1	Linguagem PHP	17
2.2	Frameworks de desenvolvimento Web	18
2.2.1	A sua importância no desenvolvimento de softwares	18
2.2.2	Frameworks Front-end e Back-end	18
2.3	Principais FRAMEWORKS PHP	19
2.3.1	Laravel	20
2.3.2	Symfony	20
2.3.3	CakePHP	21
2.3.4	CodeIgniter	21
2.3.5	FuelPHP	22
2.4	Principais Características das Frameworks	23
<b>3</b>	<b>Metodologia do Processo Comparativo de Frameworks</b>	<b>24</b>
3.1	Identificação das principais frameworks	24
3.2	Seleção das características a serem estudadas	24
3.3	Pesquisas sobre a característica de cada framework	26
3.4	Levantamento das características	26

3.5	Comparação das características	27
<b>4</b>	<b>Comparação de frameworks de desenvolvimento para sistemas Web</b>	<b>28</b>
4.1	Proposta da framework	28
4.2	Documentação	29
4.3	Popularidade e Suporte	31
4.4	Performance	32
4.5	Sustentabilidade	35
4.6	Pacotes e componentes	36
4.7	Segurança	38
4.8	Considerações finais	40
<b>5</b>	<b>Conclusão</b>	<b>42</b>
	<b>REFERÊNCIAS</b>	<b>43</b>

# 1 INTRODUÇÃO

Com o grande avanço da internet, surgiram novas formas de comunicação com um alcance maior e mais rápido do que os serviços convencionais. Dentre eles, estão os sistemas web, que têm uma grande capacidade de manipular e armazenar dados e possibilitam o acesso a partir de aparelhos conectados à internet. Com o aumento da importância desses serviços, surgiu a necessidade de sistemas mais bem elaborados e complexos. Para atender a essas demandas, foram criadas diversas ferramentas que visam possibilitar o processo de criação desses produtos com alta qualidade e de forma mais simples. Uma dessas principais ferramentas são as frameworks de desenvolvimento. Essas são soluções de alto nível que envolvem a reutilização de peças de software que contêm soluções genéricas para um certo problema<sup>[1]</sup>.

Existem diversas frameworks de desenvolvimento criadas em linguagens de programação. Devido ao grande número de frameworks de desenvolvimento disponíveis, o desenvolvedor pode encontrar dificuldades para escolher uma dentre as disponíveis, o que pode ocasionar em uma escolha de ferramenta que não atende muito bem as necessidades do desenvolvedor, tornando o projeto mais difícil e pode ocasionar na queda da qualidade do produto final.

Esse projeto propõe o estudo de frameworks de desenvolvimento Web para identificar quais as frameworks mais indicadas para atender as necessidades do desenvolvedor de sistemas. Dentre as linguagens base das frameworks, o estudo é limitado aos frameworks criados na linguagem PHP, mediante ao fato do PHP ser a linguagem mais usada nos sistemas Web<sup>[2]</sup> e alvo de estudo para aplicação do proponente.

Diante desse problema, este projeto tem como objetivo realizar uma pesquisa bibliográfica para conduzir um estudo analítico de várias frameworks de desenvolvimento web multipropósito baseadas na linguagem PHP. O objetivo principal é demonstrar as características dessas frameworks para a comparação das ferramentas, utilizando gráficos e tabelas como apoio visual direto. Os dados coletados permitem a obtenção de conclusões sobre a melhor utilização de cada framework, levando em consideração as necessidades e o contexto do projeto e do desenvolvedor.

Esse trabalho foi dividido em três etapas principais. Primeiramente, foi realizada uma revisão bibliográfica para identificar as principais frameworks de desenvolvimento web. Em



seguida, no capítulo 3, foi proposto um formato de comparação dos frameworks, que foi detalhado e discutido. No capítulo 4, ocorreu a organização dos dados coletados, com a tabulação e análise dos mesmos. A partir dessa análise, foram fornecidas indicações de frameworks adequadas para diferentes tipos de necessidades do desenvolvedor e do projeto. Por fim, o trabalho foi concluído com um breve capítulo de Conclusões.

## 2 REVISÃO BIBLIOGRÁFICA

As próximas subseções explicam conceitos importantes para o entendimento e execução deste trabalho, foi abordado a linguagem PHP que foi escolhida por ser uma linguagem de programação *open-source* com várias vantagens: suporte a vários bancos de dados diferentes, é uma linguagem multiplataforma, ou seja, funciona em vários sistemas operacionais diferentes e a maior parte dos sites na Web são programados em PHP<sup>[2]</sup>.

Para realização desta revisão, foram estudados os principais documentos que sintetizam as publicações sobre framework, PHP e outros conceitos apresentados neste trabalho, por meio de bases de dados como Google Scholar, IEEE, Scopus e outras fontes menores, utilizando as seguintes palavras-chave: Framework, PHP, Web, *Review*, *Comparison*. Foram encontrados diversos documentos que, apesar de antigos, ainda possuem dados relevantes, bem como documentos recentes que auxiliaram na coleta de dados como por exemplo o artigo dos autores Raul Santelices e Miguel Nussbaum publicado em 2001<sup>[5]</sup>, e o livro do autor Wojciech Banczer publicado em 2015<sup>[10]</sup>. Para coleta de dados mais atualizada das ferramentas, foram utilizadas as documentações e repositórios oficiais de cada ferramenta consultados no site da instituição e no seu respectivo repositório do github.

Foram apresentados conceitos sobre as frameworks de desenvolvimento Web, ferramentas que auxiliam no desenvolvimento de projetos por meio das funcionalidades que elas fornecem. As frameworks mencionadas (Laravel, Symfony, CakePHP, CodeIgniter e FuelPHP) são baseadas em PHP e foram selecionadas por serem frameworks multipropósito promissoras e populares<sup>[6]</sup>.

Por final foram apresentadas as principais características das frameworks de desenvolvimento Web que serão usadas para comparar as ferramentas entre elas.

### 2.1 Linguagem PHP

PHP (originalmente era uma abreviação para *Personal Home Page*, mas atualmente é um acrônimo para *PHP: Hypertext Preprocessor*) foi inicialmente um conjunto de binários CGI (*Common Gateway Interface*) escrito na linguagem de programação C e desenvolvido por Rasmus Lerdorf sendo lançado em 1995. Em 1997 foi feito o desenvolvimento da

ferramenta para que ela evoluísse de uma ferramenta CGI, para uma linguagem de programação muito mais eficiente e poderosa. Assim em 1998 surge o PHP 3.0 que foi a primeira versão que se assemelha ao PHP atual<sup>[3]</sup>.

Com o passar do tempo, a linguagem recebeu diversas atualizações com melhorias e novas funções, com a versão 8.2 sendo a última disponível no ano de 2022. O PHP group oferece suporte para outras versões até 3 anos depois de seu lançamento<sup>[4]</sup>.

De acordo com a w3techs o PHP é usado em mais de 70% de todos os *websites* como linguagem de programação *server-side*<sup>[2]</sup>.

## 2.2 Frameworks de desenvolvimento Web

Uma framework de desenvolvimento é uma solução de alto nível para a reutilização de pedaços de software, uma forma de reutilização simples baseada em bibliotecas que permite partilhar funções comuns e lógica genérica de uma aplicação de domínio. Também assegura um melhor nível de qualidade do produto final, uma vez que uma parte importante da aplicação já se encontra dentro da estrutura e, portanto, já foi testada<sup>[5]</sup>. Uma framework de desenvolvimento Web é uma framework de desenvolvimento cujas soluções estão voltadas para os sistemas Web.

As frameworks de desenvolvimento podem ser baseadas em uma ou mais linguagens e podem trabalhar juntas para oferecer soluções mais completas.

### 2.2.1 A sua importância no desenvolvimento de softwares

As aplicações tecnológicas crescem em possibilidades e complexidade, criando uma dificuldade de lidar com todos os processos e funções necessários para a sua criação. Para que o desenvolvedor não tenha que se preocupar com todos os detalhes e funções genéricas, são utilizados frameworks que irão facilitar a produção do software e oferecer funções que foram testadas e otimizadas, trazendo um nível de confiabilidade do código<sup>[23]</sup>.

### 2.2.2 Frameworks Front-end e Back-end

Os sistemas Web podem ser divididos em duas partes, o *front-end* e o *back-end*. A parte em que o usuário pode ver e interagir (menus, botões, formulários) é chamada de

*front-end*. O *front-end* normalmente é projetado usando HTML(*HyperText Markup Language*), CSS(*Cascading Style Sheets*) e Javascript. Já a parte em que o sistema realiza as operações de conexão com o banco de dados, servidor e o código que interage com eles é chamada de *back-end*<sup>[24]</sup>.

As frameworks *front-end* podem ser baseadas em apenas uma linguagem, ou pode ser composto por mais de uma, além disso, elas podem trabalhar em conjunto no mesmo sistema. Algumas se preocupam apenas em entregar uma interface dinâmica e responsivo como o bootstrap, outras frameworks se preocupam em trazer páginas mais dinâmicas como o jquery que facilita a implementação de requisições AJAX (*Asynchronous JavaScript And XML*) e outras funções do javascript. Além disso, existem frameworks SPA(*Single Page Application*) como react, vue, angular, entre outros, que se tornaram extremamente populares<sup>[24]</sup>.

Já as frameworks *back-end*, diferente das frameworks *front-end*, podem ser encontradas em diversas linguagens de programação como PHP, Java, Python, Javascript e diversas outras e podem variar em vários aspectos entre si, como performance, segurança, escalabilidade, curva de aprendizado, comunidade, entre outros.

## 2.3 Principais Frameworks PHP

Diversas frameworks PHP surgiram na última década para facilitar o processo de desenvolvimento de sistemas Web. Frameworks como Laravel, Symfony, CakePHP, CodeIgniter e FuelPHP são muito usadas e são as mais promissoras de acordo com o sitepoint.com<sup>[6]</sup>.

Este trabalho não apenas cobrirá as duas frameworks mais populares (Symfony e Laravel)<sup>[25]</sup>, mas também outras frameworks que, apesar de serem menos usadas, como o CodeIgniter e o CakePHP, ainda são bem populares<sup>[26]</sup>. O trabalho também incluirá o FuelPHP que tem uma popularidade menor, mas se destaca pelo uso do padrão de projeto HMVC.

As características das frameworks foram pautadas principalmente de fontes direto da organização responsável pelo desenvolvimento/fornecimento da ferramenta.

### 2.3.1 Laravel

Laravel, a framework PHP para web artesãos, é uma framework PHP *open-source* criada em 2011 por Taylor Otwell. A frase usada pelo seu criador para definir a framework é

“Laravel é uma estrutura de aplicação web com uma sintaxe expressiva e elegante. Acreditamos que o desenvolvimento deve ser uma experiência agradável e criativa para ser verdadeiramente gratificante. Laravel tenta tirar a dor do desenvolvimento, facilitando as tarefas comuns utilizadas na maioria dos projetos web.”<sup>[7]</sup>.

Em 2009 a versão PHP 5.3 tinha sido lançada e oferecia suporte a orientação ao objeto e outros recursos, mas muitas frameworks direcionaram seus esforços em manter suporte a versões mais antigas do PHP ao invés de aprimorarem as suas ferramentas. Grandes frameworks da época consistiam em Symfony, Zend, Kohana, Lithium e CodeIgniter, sendo o CodeIgniter provavelmente o mais conhecido da época. Taylor se deparando com esse cenário decide criar a sua própria ferramenta, que inicialmente tinha o intuito de fornecer uma alternativa ao CodeIgniter que, de acordo com o mesmo, não fornecia recursos essenciais, como autenticação e autorização de usuários<sup>[8]</sup>.

Na sua primeira versão a ferramenta já oferecia suporte a ORM (*Object-Relational Mapping*), mecanismo de *routing*, *cache*, sessões e outras funcionalidades, mas ainda não era uma framework MVC (*Model View Controller*) por não ter controllers, mas em menos de seis meses a sua segunda versão foi lançada com a adição desse recurso. Ao decorrer de suas versões a framework recebeu mais funcionalidades e até foi reescrita, tornando-se a ferramenta que conhecemos hoje<sup>[8]</sup>.

### 2.3.2 Symfony

“Symfony é um conjunto de Componentes PHP, uma Estrutura de Aplicação Web, uma Filosofia, e uma Comunidade - todos trabalhando juntos em harmonia”<sup>[9]</sup>. Criado em 2005 por Fabien Potencier, symfony é uma framework PHP *open-source* com a proposta de agilizar a criação e manutenção de aplicações Web robustas e bem customizável. A filosofia da ferramenta é “Abraçando e promovendo o profissionalismo, melhores práticas, padronização e interoperabilidade de aplicações.”<sup>[9]</sup>.

A primeira versão estável foi lançada em 2007 e teve suporte durante três anos, um ano depois foi lançada a versão 1.1 que sucedeu a primeira versão, mas não era compatível com a mesma, tornando difícil a migração do projeto. Posteriormente foram lançadas novas

versões que eram compatíveis com a 1.1, mas não havia grandes mudanças em sua estrutura, o que facilitava a migração de versões, mas não trazia muitas mudanças. Em 2011 foi então lançada a versão 2.0, tendo o seu código-fonte totalmente reescrito. Essa nova versão não era perfeita, mas parecia muito promissora<sup>[10]</sup>. A partir da versão 2.2 foi-se adotado um novo processo de lançamento de versões da framework <sup>[11]</sup>.

A ferramenta é patrocinada pela SensioLabs que se responsabiliza em criar produtos *open-source* como o twig (uma biblioteca utilizada pelo symfony) e outros produtos.

O Symfony conta com um grande número de componentes, tais como pacotes de segurança, pacote ORM, testes, componentes para twig, entre outros. Vários desses componentes também são aproveitados por outras frameworks como, por exemplo, o Laravel<sup>[12]</sup>.

### 2.3.3 CakePHP

CakePHP é uma framework PHP *open-source* com o objetivo de oferecer uma estrutura onde usuários de todos os níveis podem desenvolver aplicações Web robustas com agilidade sem perda de flexibilidade<sup>[13]</sup>.

Em 2005 Michal Tatarynowicz criou uma framework em PHP básica inspirada pela framework Ruby on Rails<sup>[14]</sup>. O desenvolvedor acreditando que sua ferramenta tinha potencial, a publicou usando a licença MIT (*Massachusetts Institute of Technology*) e a disponibilizou para a comunidade. A partir disso foi-se criando um grupo que se responsabiliza em gerir e atualizar a ferramenta conhecida como “*The Bakers*”, tendo como co-fundador Larry Masters.

Assim como as frameworks citadas acima, o cakePHP é uma framework MVC com suporte a ORM, ferramentas de segurança e outros pacotes, mas o seu diferencial se encontra no módulo de auto detecção de configurações. Por meio desse módulo a ferramenta tira os diversos arquivos de configuração do caminho do desenvolvedor e torna a produção mais simples<sup>[15]</sup>.

### 2.3.4 CodeIgniter

“CodeIgniter é uma framework PHP *open-source* cujo objetivo é possibilitar um desenvolvimento de projeto muito mais rápido do que se o desenvolvedor apenas utilizasse a linguagem pura, através de bibliotecas para funções genéricas, uma interface simples e

estrutura lógica para acessar essas bibliotecas. CodeIgniter oferece as ferramentas que o desenvolvedor necessita sem entrar em seu caminho”<sup>[16]</sup>.

Criada em 2006 pela Ellislab, uma empresa de desenvolvimento de software, foi a responsável pela ferramenta até 2014. A empresa afirmou que não tinha mais recursos para manter a ferramenta da forma que era merecido, confiou o CodeIgniter a *British Columbia Institute of Technology* (BCIT)<sup>[17]</sup>. A BCIT manteve a posse até 2019. Atualmente a framework se encontra aos cuidados da CodeIgniter Foundation, a responsável pelo CodeIgniter 4, a mais recente versão da ferramenta<sup>[18]</sup>.

O CodeIgniter é uma ferramenta compacta com uma forte segurança, desempenho melhor do que a maioria de seus competidores, quase nenhuma configuração e prefere soluções simples ao invés das complexas. Enquanto a ferramenta disponibiliza suporte a recursos importantes como suporte a ORM, autenticação, entre outros, ela se mantém leve por carregar apenas os módulos que o desenvolvedor usa, deixando outros recursos inutilizados de lado. A ferramenta também não prende o desenvolvedor nos padrões de projeto, permitindo ele evitar o uso dos mesmo, como por exemplo o MVC, que apesar de ser encorajado, não é forçado<sup>[19]</sup>.

### 2.3.5 FuelPHP

“FuelPHP é uma framework PHP *open-source* que se destaca na multidão das frameworks entregando as coisas boas das grandes frameworks enquanto retira as partes ruins delas”<sup>[20]</sup>.

Criado em 2011 por um grupo de desenvolvedores, a ferramenta atualmente se encontra em sua versão 1.8 e foi criado um repositório no GitHub para o desenvolvimento da versão 2.0.

FuelPHP é a framework mais nova dentre às cinco citadas, mas é uma ferramenta com novos conceitos e paradigmas como o HMVC (*Hierarchical Model View Controller*) ao invés do MVC. O HMVC oferece melhor organização de código, maior modularidade, mais extensibilidade e encoraja o reuso de código<sup>[6]</sup>.

Apesar de ser uma framework jovem se comparada com a maioria de seus concorrentes, ela possui diversos recursos importantes como módulos de segurança, suporte a ORM, autenticação e vários outros. Alguns pontos negativos da ferramenta é que a comunidade é pequena e atualmente a ferramenta não é compatível com as versões mais recentes do PHP<sup>[20]</sup>.

## 2.4 Principais Características das Frameworks

Para se avaliar as frameworks citadas acima é necessário definir quais características serão importantes serem analisadas.

De acordo com o artigo “*selecting a php framework for a web application project - the method and case study*” (traduzido como “Selecionando uma Framework PHP para um Projeto de Aplicação Web - O Método e Estudo de Caso”) <sup>[21]</sup>, existem diversos fatores a serem considerados na escolha da framework, como a sua ideia principal, popularidade, sustentabilidade, suporte, entre outros.

O artigo “*How to choose a web development framework: Analyzing best practices on the adoption of web frameworks*” (traduzido como “Como escolher um framework de desenvolvimento web: Analisando as melhores práticas na adoção de frameworks web”) <sup>[27]</sup>, de igual modo traz diversos fatores técnicos e não técnicos a serem considerados na escolha de uma framework, tais como desempenho, facilidade de manutenção, suporte a outras ferramentas, segurança, estabilidade, popularidade e documentação, entre outros.

Por fim, o artigo “*PHP Frameworks Usability in Web Application Development*” (traduzido como “Usabilidade das frameworks PHP no desenvolvimento de aplicações web”) <sup>[28]</sup> apresenta oito fatores para analisar uma framework: documentação e suporte técnico, ferramentas e suporte para código-fonte gerado, técnicas de banco de dados, *cache*, integração/compatibilidade com outros sistemas, concisão do código-fonte e eficiência da framework.



### **3 METODOLOGIA DO PROCESSO COMPARATIVO DE FRAMEWORKS**

Para realizar o processo comparativo das frameworks de desenvolvimento Web selecionadas, foram realizadas as seguintes etapas:

1. Identificação das principais frameworks
2. Seleção das características a serem estudadas
3. Pesquisas sobre a característica de cada framework
4. Levantamento das características
5. Comparação das características

#### **3.1 Identificação das principais frameworks**

De acordo com o tópico 2.3 as 5 frameworks selecionadas são muito populares, de multipropósito e oferecem um amplo leque de soluções para os diversos problemas que podem surgir no desenvolvimento do sistema. São elas: Laravel, Symfony, CakePHP, CodeIgniter e FuelPHP.

#### **3.2 Seleção das características a serem estudadas**

A lista das características selecionadas foi elaborada a partir dos artigos de comparação relatados no tópico 2.4. Para compor essa lista, foram consideradas as características presentes em todos os artigos.

As características selecionadas para a comparação das frameworks são:

- Proposta de framework

A proposta da framework PHP é uma das primeiras coisas a se considerar ao comparar diferentes opções. Cada framework tem sua própria abordagem para o desenvolvimento de aplicativos web. Algumas se concentram em fornecer uma ampla gama de recursos e funcionalidades, enquanto outras são mais leves e focadas em desenvolvimento mais rápido. É importante encontrar uma framework que se encaixe bem nos requisitos do projeto.

- Documentação

A documentação da framework é essencial para entender como ela funciona e como usá-la corretamente. Uma documentação bem escrita e organizada pode economizar muito tempo e esforço durante o desenvolvimento. Além disso, uma boa documentação pode ser um indicador da qualidade geral da framework e do compromisso da comunidade em torno dela.

- Popularidade e suporte

A popularidade e o suporte também são fatores importantes a serem considerados. Frameworks populares tendem a ter uma grande comunidade de desenvolvedores ativos, o que pode levar a um grande número de recursos e documentação disponíveis. Além disso, é mais provável que essas frameworks recebam atualizações e correções de segurança regulares. Por outro lado, algumas frameworks menos conhecidas podem oferecer recursos exclusivos e serem mais adaptáveis para projetos específicos.

- Performance

O desempenho é um fator crucial a ser considerado, especialmente para aplicativos web em larga escala. A velocidade de carregamento e a eficiência do código podem afetar significativamente a experiência do usuário. É importante escolher uma framework que possa oferecer desempenho confiável e escalável.

- Sustentabilidade

A sustentabilidade é importante porque uma framework que não é mais suportada ou atualizada pode apresentar problemas de segurança e desempenho. Ao comparar diferentes frameworks, é importante considerar o quão ativo é o desenvolvimento, quantas atualizações recentes foram lançadas e qual é a política de suporte a longo prazo.

- Pacotes e componentes

Os pacotes e componentes disponíveis para uma framework também podem ser importantes, pois eles podem economizar tempo e esforço no desenvolvimento de recursos e funcionalidades específicas. É importante avaliar o quão bem esses pacotes e componentes funcionam e se eles estão atualizados e bem mantidos.

- Segurança

Finalmente, a segurança é um fator crítico a ser considerado, especialmente para aplicativos web que processam informações confidenciais do usuário. É

importante escolher uma framework que tenha recursos de segurança robustos e regulares atualizações de segurança.

Ao comparar diferentes frameworks PHP, é importante avaliar cada uma dessas características para determinar qual é a melhor escolha para um projeto específico. A escolha da framework certa pode ter um grande impacto na eficiência e sucesso do desenvolvimento de um projeto de aplicativo web.

### 3.3 Pesquisas sobre a característica de cada framework

Foi realizado uma pesquisa das características de cada ferramenta, utilizando os sites oficiais dos frameworks, seus repositórios oficiais e o *benchmark* do ano de 2021 da TechEmpower<sup>[22]</sup> para coletar as últimas atualizações sobre essas ferramentas.

As fontes de cada framework, são:

- Laravel: <https://laravel.com/> & <https://github.com/laravel>
- Symfony: <https://symfony.com/> & <https://github.com/symfony/>
- CodeIgniter: <https://codeigniter.com/> & <https://forum.codeigniter.com/> & <https://github.com/bcit-ci/CodeIgniter>
- FuelPHP: <https://fuelphp.com/> & <https://github.com/fuel/>
- CakePHP: <https://cakephp.org/> & <https://github.com/cakephp/>

### 3.4 Levantamento das características

A partir da pesquisa, foi realizada uma breve descrição de cada característica de cada ferramenta.

Posteriormente, foi realizado uma comparação dessas características para em quais casos/circunstâncias que certa framework pode se destacar em relação às outras.

### **3.5 Comparação das características**

Após a coleta de dados de cada característica foi comparado aspectos que diferem as ferramentas e ressaltou-se as características mais importantes de cada uma.

Cada característica da framework é acompanhada por tabela(s) e/ou gráfico(s) que demonstram a característica, No final de cada subcapítulo de comparação das características das frameworks selecionadas, foram criadas tabelas e gráficos para visualização e comparação dos dados coletados. Esses recursos visuais foram utilizados para apresentar de forma clara e objetiva as informações levantadas sobre cada característica. Foi realizada a tabulação da característica para todas as frameworks.

## 4 COMPARAÇÃO DE FRAMEWORKS DE DESENVOLVIMENTO PARA SISTEMAS WEB

A comparação foi realizada conforme disposto no processo da metodologia. Inicialmente fez a pesquisas sobre a característica de cada framework e o levantamento das características. Para sintetizar e facilitar a comparação de cada características foi realizada a tabulação da característica para todas as frameworks nos tópicos:

- Proposta de framework
- Documentação
- Popularidade e suporte
- Performance
- Sustentabilidade
- Pacotes e componentes
- Segurança

### 4.1 Proposta da framework

Ao comparar a proposta de frameworks Laravel, Symfony, CakePHP, CodeIgniter e FuelPHP, é possível observar que todas compartilham o objetivo comum de tornar o desenvolvimento em PHP mais eficiente e produtivo. No entanto, cada framework tem suas particularidades.

O Laravel, por exemplo, tem como proposta fornecer uma experiência de desenvolvimento mais agradável e expressiva, com um foco especial em uma sintaxe elegante e intuitiva.

O Symfony é conhecido por sua flexibilidade e escalabilidade, permitindo que os desenvolvedores criem aplicações de grande porte de maneira organizada e estruturada.

O CakePHP é uma framework que visa acelerar o processo de desenvolvimento, oferecendo uma estrutura de arquivos bem definida e pronta para uso.

O CodeIgniter tem uma proposta similar, focando em simplicidade e facilidade de uso para desenvolvedores iniciantes.

Por fim, o FuelPHP tem como proposta oferecer uma estrutura robusta e modular, com suporte para recursos avançados de programação e uma sintaxe elegante.

A Tabela 1 demonstra a proposta de cada framework.

Tabela 1 - Tabela da proposta das frameworks.

Framework	Proposta
Laravel	Experiência de desenvolvimento agradável
Symfony	Flexibilidade e escalabilidade
CakePHP	Aceleração do processo de desenvolvimento
CodeIgniter	Simplicidade e facilidade de uso
FuelPHP	Estrutura robusta e modular

Fonte: elaborado pelo autor (2023)

## 4.2 Documentação

Os frameworks Laravel, Symfony, CakePHP, CodeIgniter e FuelPHP são muito populares no desenvolvimento web em PHP, e todos eles oferecem uma ampla documentação para os desenvolvedores que desejam usá-los.

O Laravel tem uma documentação bem organizada e fácil de seguir. A documentação é dividida em seções, incluindo uma introdução ao framework, instalação, configuração, uso de banco de dados, autenticação e autorização, rotas, middlewares e muitos outros tópicos. Além disso, a documentação é escrita em inglês e conta com uma comunidade ativa de desenvolvedores que ajudam a mantê-la atualizada e a responder dúvidas.

O Symfony possui uma documentação igualmente bem estruturada, porém com um enfoque mais técnico. A documentação está disponível em inglês e também é dividida em várias seções, incluindo fundamentos, arquitetura, componentes, criação de aplicativos web e muitos outros tópicos. Além disso, a documentação conta com uma forte ênfase em boas práticas de programação e design de software.

O CakePHP tem uma documentação que segue a mesma estrutura do Laravel e Symfony. A documentação está disponível em inglês, português, espanhol, japonês e francês e abrange tópicos como instalação, configuração, rotas, controllers, views, models, testes, autenticação e muitos outros tópicos. A documentação também possui uma forte ênfase em boas práticas de programação e design de software.

O CodeIgniter tem uma documentação que segue uma estrutura um pouco diferente dos frameworks mencionados anteriormente. A documentação está disponível em inglês e conta com seções sobre introdução, instalação, configuração, uso básico, controllers, models, views, libraries e muitos outros tópicos. A documentação é bem escrita e fácil de seguir, mas não é tão abrangente quanto a documentação de outros frameworks.

Por fim, o FuelPHP possui uma documentação que é bem similar ao CodeIgniter, com seções sobre instalação, configuração, rotas, controllers, views, models, validação, autenticação e muitos outros tópicos. A documentação é bem escrita e fácil de seguir, porém não é tão abrangente quanto a documentação de outros frameworks.

Em resumo, os frameworks Laravel, Symfony e CakePHP possuem documentações bem organizadas e abrangentes, com uma forte ênfase em boas práticas de programação e design de software. Já o CodeIgniter e FuelPHP possuem documentações mais simples e menos abrangentes.

A tabela 2 demonstra características da documentação de cada framework.

Tabela 2 - Tabela da documentação das frameworks.

Framework	Idiomas disponíveis na documentação	Estrutura da documentação	Enfoque da documentação	Ênfase em boas práticas
Laravel	Inglês	Organizada em seções	Prático	Sim, enfatiza boas práticas de programação
Symfony	Inglês	Estruturada em várias seções	Técnico	Sim, enfatiza boas práticas de programação e design de software
CakePHP	Inglês, português, espanhol, japonês e francês	Organizada em seções	Prático	Sim, enfatiza boas práticas de programação e design de software
CodeIgniter	Inglês	Estrutura um pouco diferente, mas completa	Prático	Não tão enfatizada quanto em outros frameworks

FuelPHP	Inglês	Estrutura similar ao CodeIgniter	Prático	Não tão enfatizada quanto em outros frameworks
---------	--------	----------------------------------	---------	--

Fonte: elaborado pelo autor (2023)

### 4.3 Popularidade e suporte

Os frameworks PHP Laravel, Symfony, CakePHP, CodeIgniter e FuelPHP possuem suas próprias vantagens e desvantagens em relação à popularidade e suporte, o que pode afetar a escolha de um desenvolvedor ao selecionar um para um projeto específico.

Em termos de popularidade, Laravel é, sem dúvida, o framework mais popular entre os desenvolvedores PHP. De acordo com o relatório de popularidade do GitHub de 2023, o Laravel é o framework PHP mais popular, com mais de 72 mil estrelas e 23 mil forks no repositório do GitHub.

Em segundo lugar em popularidade está o Symfony, que possui mais de 28 mil estrelas e 9 mil forks no GitHub.

Em terceiro lugar em popularidade está o CodeIgniter tem mais de 18 mil estrelas e 7 mil forks no GitHub.

Em quarto lugar em popularidade está o CakePHP possui mais de 8 mil estrelas e 3 mil forks no GitHub.

Por fim, o FuelPHP tem pouco mais de mil estrelas e 280 forks no GitHub.

Quanto ao suporte, todos os frameworks mencionados têm uma comunidade ativa e um bom suporte de documentação. O Laravel tem uma documentação muito detalhada e uma comunidade forte que fornece muitos recursos e ajuda no desenvolvimento. O Symfony também possui uma documentação abrangente e um forte suporte da comunidade, além de um grande número de desenvolvedores que contribuem com novos recursos e plugins. O CakePHP é conhecido por ter uma comunidade muito ativa, com muitos fóruns e grupos de discussão disponíveis para suporte. O CodeIgniter também possui uma comunidade forte e uma documentação clara e concisa, enquanto o FuelPHP tem uma comunidade menor, mas ainda ativa e uma documentação útil e clara.



A tabela 3 demonstra valores de estrelas e forks nos repositórios e características do suporte de cada framework com valores aproximados para baixo.

Tabela 3 - Tabela da popularidade e suporte das frameworks.

Framework	Popularidade (estrelas no GitHub)	Forks no GitHub	Suporte
Laravel	72.000	23.000	Forte comunidade, documentação detalhada
Symfony	28.000	9.000	Forte comunidade, documentação abrangente
CodeIgniter	18.000	7.000	Comunidade forte, documentação clara
CakePHP	8.000	3.000	Comunidade ativa, muitos fóruns e grupos
FuelPHP	1.000	280	Comunidade menor, documentação útil e clara

Fonte: elaborado pelo autor (2023)

#### 4.4 Performance

Performance em um framework diz respeito à rapidez e eficiência com que ele processa e responde às solicitações do usuário. A velocidade de carregamento das páginas, o tempo de resposta das requisições, o consumo de recursos e outros fatores influenciam na performance. Uma boa performance é essencial para garantir a satisfação do usuário e a escalabilidade do sistema. Para comparar as frameworks, utilizamos o benchmark da Techempower de 2021<sup>[22]</sup>, que inclui os seguintes testes:

**Serialização JSON:** cada resposta é uma serialização JSON de um objeto recém-criado que mapeia a chave mensagem para o valor "Hello, World!"

**Consulta única:** cada solicitação busca uma única linha de uma tabela de banco de dados simples, que é serializada como uma resposta JSON

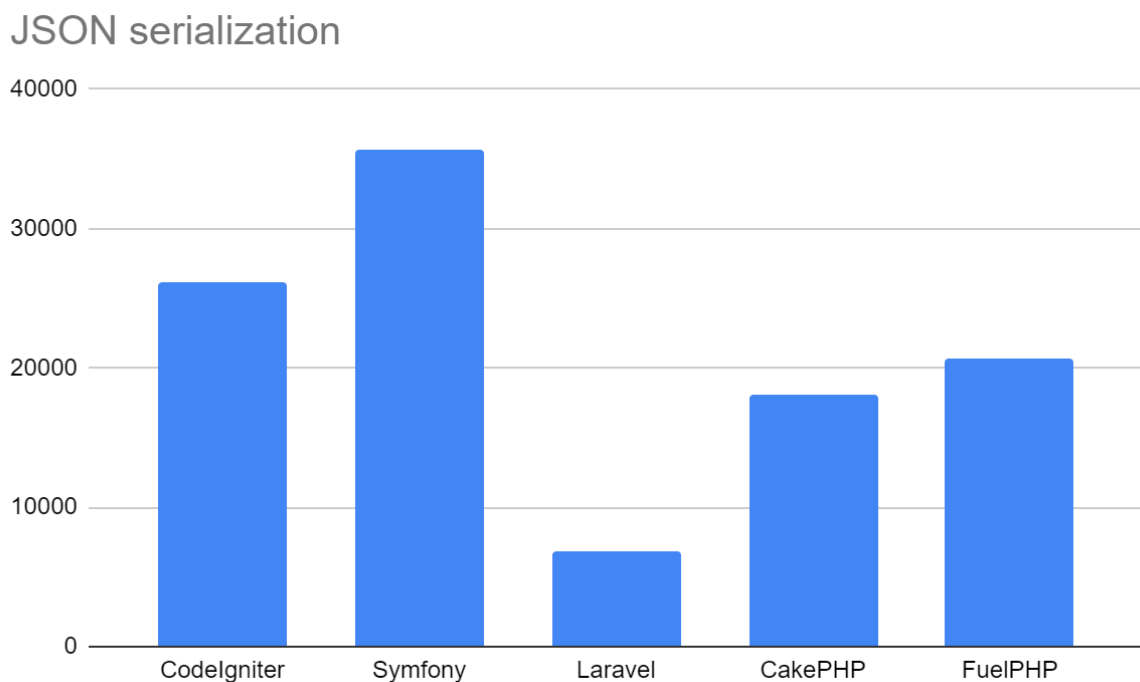
**Múltiplas consultas:** cada solicitação busca várias linhas de uma tabela de banco de dados simples, que são serializadas como uma resposta JSON. O teste é executado várias

vezes, com 1, 5, 10, 15 e 20 consultas por solicitação, e todos os testes são executados com 512 concorrências.

Fortunes: o ORM do framework é usado para buscar todas as linhas de uma tabela de banco de dados contendo um número desconhecido de mensagens de fortune cookie Unix (a tabela tem 12 linhas, mas o código não pode ter conhecimento do tamanho da tabela). Uma mensagem de fortune cookie adicional é inserida na lista em tempo de execução e, em seguida, a lista é classificada pelo texto da mensagem. Por fim, a lista é entregue ao cliente usando um modelo HTML no lado do servidor. O texto da mensagem deve ser considerado não confiável e devidamente escapado, e as mensagens de fortune cookie UTF-8 devem ser renderizadas corretamente.

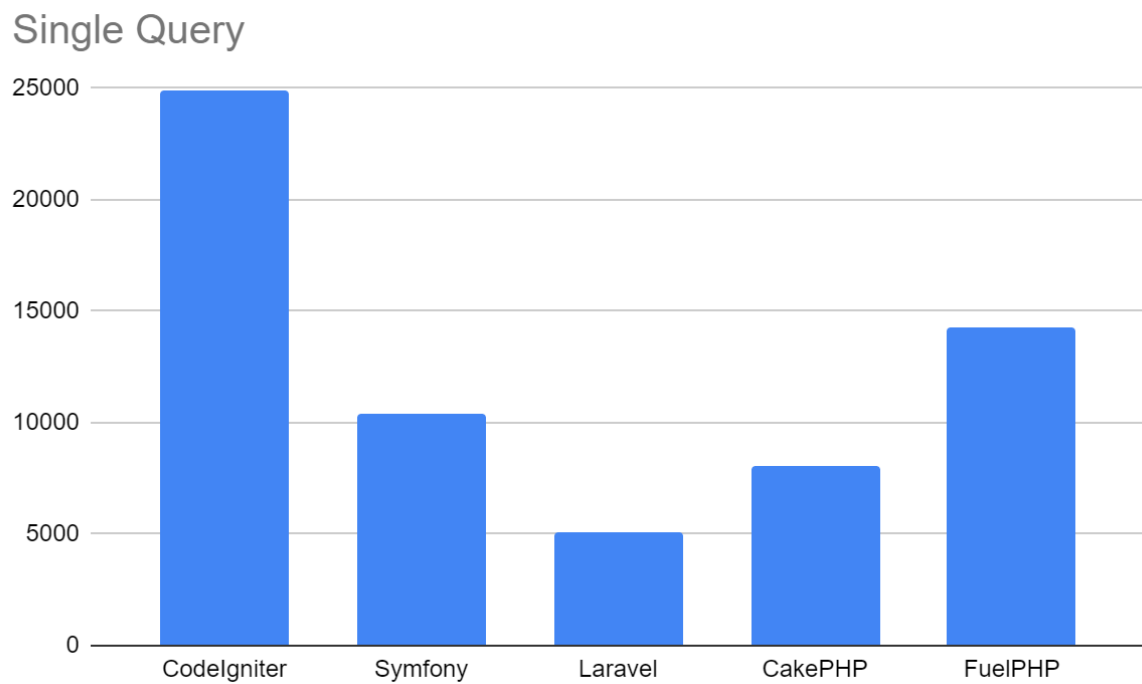
Na figura 1 a figura 4 foram tabulados as características dos testes de benchmark descritos no título do gráfico.

Figura 1 - Benchmark de JSON serialization.



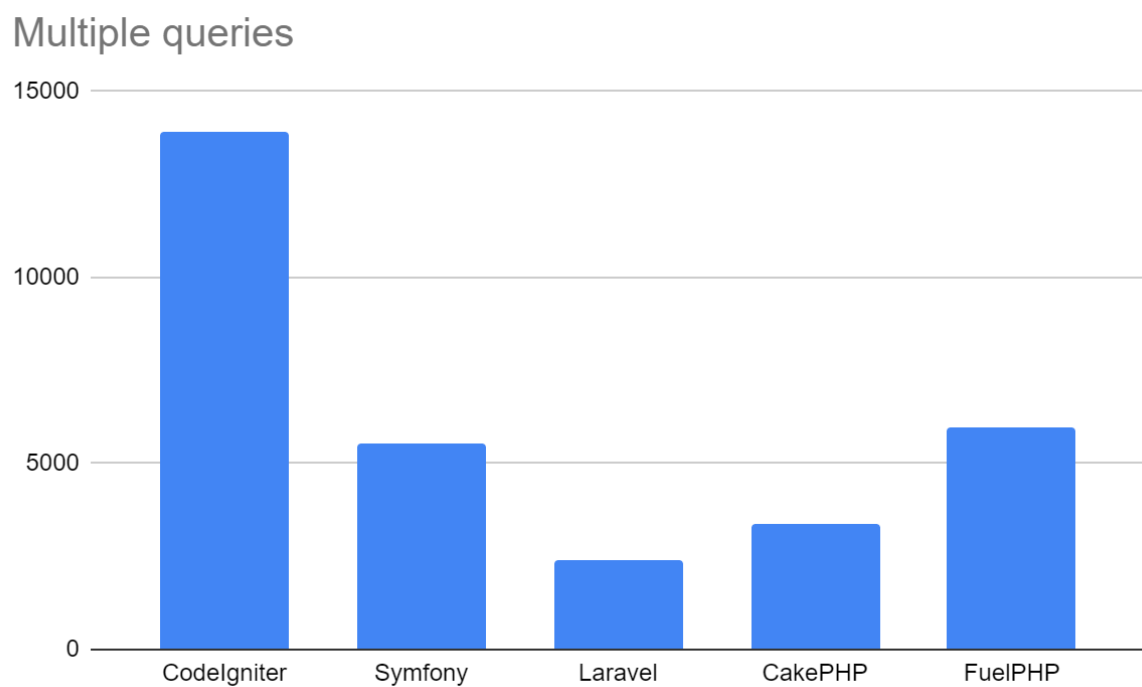
Fonte: elaborado pelo autor (2023)

Figura 2 - Benchmark de Single query.



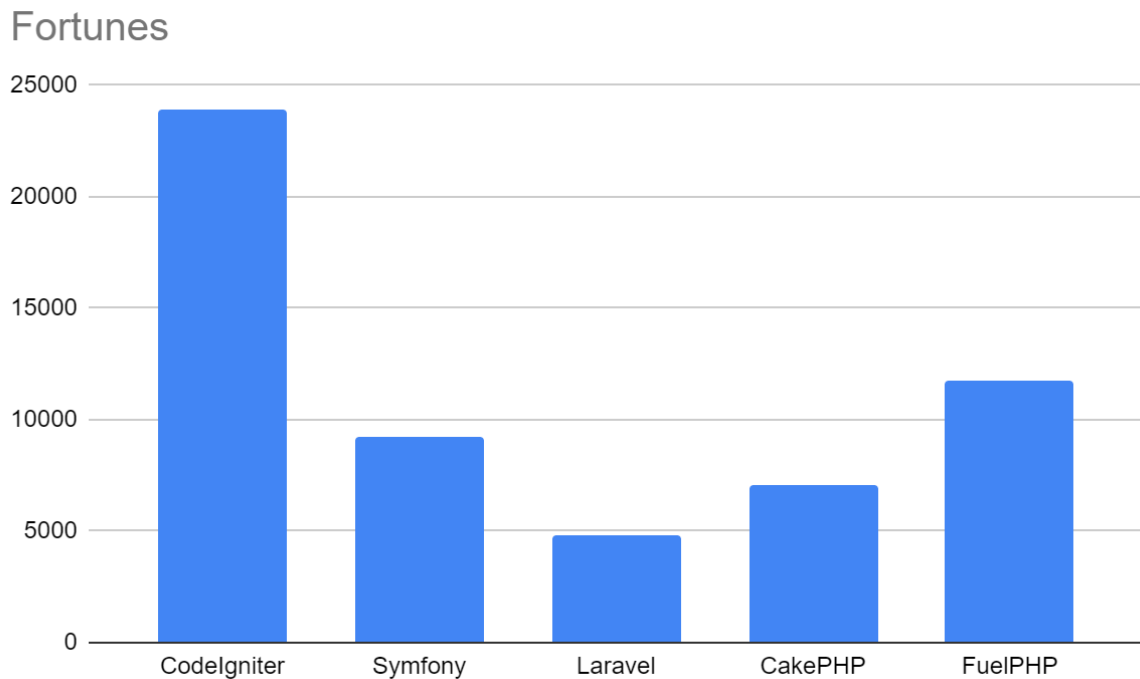
Fonte: elaborado pelo autor (2023)

Figura 3 - Benchmark de Multiple queries.



Fonte: elaborado pelo autor (2023)

Figura 4 - Benchmark de Fortunes.



Fonte: elaborado pelo autor (2023)

## 4.5 Sustentabilidade

Uma framework sustentável é aquela que é bem documentada, possui uma comunidade ativa e uma base sólida de usuários e desenvolvedores. Ela deve ser capaz de lidar com atualizações de tecnologia e manter-se atualizada com as mudanças nas necessidades dos usuários e das empresas ao longo do tempo.

Em termos de suporte, documentação e comunidade, Symfony é provavelmente a escolha mais robusta. Com mais de uma década de desenvolvimento ativo e suporte de grandes empresas como SensioLabs, o Symfony tem uma base sólida de usuários e uma vasta documentação online. Além disso, o Symfony conta com um grande número de pacotes e bibliotecas que podem ser utilizados em conjunto com o framework. Por conta disso, muitos desenvolvedores consideram o Symfony um framework sustentável para longo prazo.

Laravel, por sua vez, é um dos frameworks mais populares atualmente, principalmente por sua simplicidade e facilidade de uso. O Laravel possui uma comunidade ativa e uma documentação bem elaborada, o que o torna uma escolha sustentável para projetos de

pequeno a médio porte. No entanto, como o Laravel é relativamente novo em comparação com outros frameworks, pode haver incertezas sobre sua longevidade.

CakePHP tem uma história de mais de 18 anos, o que o torna uma opção razoavelmente estável. Embora tenha uma comunidade menor do que outros frameworks, o CakePHP ainda tem uma base de usuários dedicada e uma documentação sólida.

CodeIgniter é outro framework de longa data, com uma comunidade dedicada e uma documentação sólida. No entanto, ao contrário de outros frameworks, o CodeIgniter tem um número limitado de atualizações recentes, o que pode sugerir uma possível estagnação no desenvolvimento.

Por fim, FuelPHP é um framework relativamente novo que ainda está em desenvolvimento ativo. Apesar de ter uma comunidade pequena, o FuelPHP tem recursos interessantes, como uma arquitetura modular e flexibilidade em relação a outros frameworks. No entanto, pode haver incertezas sobre sua sustentabilidade a longo prazo, já que ainda não teve tempo suficiente para estabelecer uma base sólida de usuários e desenvolvedores e tem um número ainda mais limitado de atualizações do que o CodeIgniter.

A tabela 4 mostra características de cada framework que contribuem para uma boa sustentabilidade.

Tabela 4 - Tabela da sustentabilidade das frameworks.

Framework	Histórico de desenvolvimento	Comunidade	Documentação	Atualizações frequentes
Laravel	Relativamente novo	Ativa	Elaborada	Sim
Symfony	Mais de uma década	Robusta	Vasta	Sim
CakePHP	Mais de 18 anos	Dedicada	Sólida	Sim
CodeIgniter	Longa data	Dedicada	Sólida	Não
FuelPHP	Relativamente novo	Pequena	Interessante	Não

Fonte: elaborado pelo autor (2023)

## 4.6 Pacotes e componentes

As frameworks Laravel, Symfony, CakePHP, CodeIgniter e FuelPHP compartilham algumas características semelhantes em termos de organização de código, estrutura de diretórios e arquitetura MVC (Model-View-Controller). No entanto, cada uma dessas frameworks tem suas próprias características únicas em relação aos pacotes e componentes que oferecem.

Laravel é uma das frameworks PHP mais populares e modernas atualmente disponíveis. Ele possui uma ampla gama de recursos integrados, como autenticação, roteamento, ORM, cache, filas e agendamento de tarefas. A framework também possui sua própria implementação do container de dependência, chamado de "Service Container", que é usado para gerenciar a instância dos objetos.

Symfony é uma framework PHP poderosa e flexível que oferece uma variedade de pacotes e componentes reutilizáveis. Ele fornece recursos abrangentes para gerenciamento de rotas, autenticação, segurança, cache e gerenciamento de dependências. O Symfony usa o container de serviço para gerenciar a instância dos objetos.

O CakePHP é uma framework PHP que se concentra na simplicidade e facilidade de uso. Apesar de oferecer uma gama menor de recursos, ele possui recursos integrados importantes, tais como validação de formulários, roteamento, autenticação e autorização, cache e ORM. A framework também possui sua própria implementação do container de dependência, chamado de "Dependency Injection Container".

O CodeIgniter é uma framework PHP leve e rápida que é fácil de aprender e usar. Ele oferece recursos integrados para gerenciamento de rotas, banco de dados, formulários, validação e upload de arquivos, mas também é mais limitado do que o Laravel e Symfony. O gerenciamento de pacotes é feito manualmente.

O FuelPHP é uma framework PHP que visa fornecer uma arquitetura HMVC clara e estruturada. Ele possui recursos integrados, como roteamento, ORM, cache, validação de formulários e segurança. O FuelPHP usa o container de serviço para gerenciar a instância dos objetos.

Com exceção do CodeIgniter, que utiliza outro gerenciador de pacotes, todas as frameworks utilizam o composer como padrão. Além disso, a maioria delas segue o padrão PSR (PHP Standard Recommendation) até um certo nível, garantindo a compatibilidade e a interoperabilidade entre diferentes projetos.

A tabela 5 apresenta algumas características sobre os pacotes e componentes de cada framework.

Tabela 5 - Tabela de pacotes e componentes das frameworks.

Framework	Arquitetura	Container de Dependência	Gerenciador de Pacotes
Laravel	MVC	Service Container	Composer
Symfony	MVC	Service Container	Composer
CakePHP	MVC	Dependency Injection Container	Composer
CodeIgniter	MVC	N/A	Manual
FuelPHP	HMVC	Service Container	Composer

Fonte: elaborado pelo autor (2023)

## 4.7 Segurança

A escolha de uma framework segura é fundamental para garantir a proteção dos dados e a confiabilidade de um sistema, evitando vulnerabilidades e ataques cibernéticos. Cada framework tem sua própria abordagem em relação à segurança

Laravel é conhecido por ser uma framework segura, com recursos de segurança incorporados em seu núcleo. A framework tem uma série de recursos para ajudar a proteger contra ataques, incluindo proteção contra CSRF, injeção de SQL e XSS. O Laravel também é atualizado regularmente para corrigir vulnerabilidades de segurança. Além disso, a documentação do Laravel oferece muitos conselhos e diretrizes de segurança para desenvolvedores.

Assim como o Laravel, o Symfony é uma framework segura e tem uma série de recursos de segurança em seu núcleo, incluindo proteção contra CSRF, injeção de SQL e XSS. Além disso, o Symfony tem um sistema de autenticação robusto e recursos de autorização para controlar o acesso a determinadas partes do aplicativo. O Symfony também é atualizado regularmente para corrigir vulnerabilidades de segurança e tem uma comunidade ativa para fornecer suporte e conselhos de segurança.

O CakePHP tem recursos de segurança em seu núcleo, incluindo proteção contra CSRF, injeção de SQL e XSS. Além disso, o CakePHP tem uma série de recursos de

segurança adicionais, como proteção contra ataques de força bruta e detecção de tentativas de login suspeitas. O CakePHP é atualizado regularmente para corrigir vulnerabilidades de segurança e tem uma comunidade ativa para fornecer suporte e conselhos de segurança.

O CodeIgniter é uma framework mais antiga e, apesar de ter algumas medidas de segurança em seu núcleo, não é tão seguro quanto algumas das outras frameworks mencionadas. No entanto, existem recursos de segurança adicionais disponíveis por meio de bibliotecas de terceiros e, como o CodeIgniter é uma framework mais simples, é mais fácil para os desenvolvedores garantir a segurança do aplicativo.

O FuelPHP tem uma série de recursos de segurança em seu núcleo, incluindo proteção contra CSRF, injeção de SQL e XSS. Além disso, o FuelPHP tem um sistema de autenticação integrado e recursos de autorização para controlar o acesso a determinadas partes do aplicativo. O FuelPHP não é atualizado a algum tempo, o que pode indicar que vulnerabilidades de segurança descobertas recentemente não foram corrigidas, mas tem uma comunidade ativa para fornecer suporte e conselhos de segurança.

A tabela 6 apresenta características de segurança de cada framework.

Tabela 6 - Tabela da segurança das frameworks.

Framework	Recursos de Segurança Incorporados	Recursos de Segurança Adicionais	Atualizações Regulares	Comunidade Ativa
Laravel	Proteção contra CSRF, injeção de SQL e XSS	Diretrizes de segurança na documentação	Sim	Sim
Symfony	Proteção contra CSRF, injeção de SQL e XSS, sistema de autenticação e autorização	Recursos de segurança disponíveis em bibliotecas de terceiros	Sim	Sim
CakePHP	Proteção contra CSRF, injeção de SQL e XSS, proteção contra ataques de força bruta e detecção de tentativas de login suspeitas	Recursos de segurança disponíveis em bibliotecas de terceiros	Sim	Sim
CodeIgniter	Algumas medidas de	Recursos de segurança	Não é tão regular quanto	Comunidade ativa disponível



	segurança	disponíveis em bibliotecas de terceiros	outras frameworks	
FuelPHP	Proteção contra CSRF, injeção de SQL e XSS, sistema de autenticação e autorização	Recursos de segurança disponíveis em bibliotecas de terceiros	Não é tão regular quanto outras frameworks	Comunidade ativa disponível, porém menor em número

Fonte: elaborado pelo autor (2023)

## 4.8 Considerações finais

Cada framework demonstrou ter uma proposta diferente, tornando essa característica mais dependente das necessidades do projeto e do desenvolvedor.

Cada framework disponibiliza sua própria documentação e oferece suporte para ajudar todos os usuários a usar as funcionalidades da ferramenta. Dentre as frameworks comparadas pode-se analisar que apenas o CakePHP apresenta uma documentação com outros idiomas além do inglês, o que pode ser um ponto importante caso o desenvolvedor tenha alguma dificuldade com o idioma. A estrutura da documentação das frameworks é um pouco diferente no CodeIgniter e no FuelPHP, mas é completa. Dentre as frameworks, o symfony apresenta uma documentação mais técnica que difere em relação às outras que são mais práticas. Por final o Laravel, Symfony e CakePHP enfatizam as boas práticas de programação, o que pode auxiliar programadores mais inexperientes a seguirem esses padrões.

Laravel e Symfony são as frameworks mais populares e predominantes no mercado, com o CodeIgniter logo atrás. Em seguida, temos o CakePHP, e por fim, o FuelPHP como a menos popular. Quanto ao suporte, todas as ferramentas contam com uma ativa comunidade, documentação abrangente e responsáveis engajados.

A ordem dos frameworks em termos de tempo de resposta para as operações selecionadas no tópico 4.4 é a seguinte: Laravel possui o menor tempo de resposta, seguido por CakePHP, FuelPHP e Symfony, em ordem crescente de tempos de resposta. Por fim o CodeIgniter apresentou os piores resultados.

Apesar das diferenças nas datas de lançamento das 5 frameworks, todas possuem documentação e comunidades bem consolidadas, com exceção do FuelPHP, que possui uma comunidade menor, porém dedicada. Além disso, Laravel, Symfony e CakePHP são

frequentemente atualizados, enquanto CodeIgniter e FuelPHP recebem atualizações com menor frequência.

A disponibilidade de pacotes e componentes na ferramenta desempenha um papel importante na eficiência do programador durante o processo de programação, uma vez que algumas frameworks fornecem recursos essenciais para o seu projeto na forma de pacotes e componentes. Laravel e Symfony destacam-se como as frameworks mais completas nesse aspecto, enquanto o CakePHP, FuelPHP e CodeIgniter são mais limitados. É importante mencionar que o CodeIgniter não utiliza o Composer como gerenciador de pacotes, o que pode ocasionar maior dificuldade no seu gerenciamento.

Por fim, as frameworks possuem uma estrutura de segurança bem definida, com práticas de segurança reforçadas em suas respectivas documentações. No entanto, devido às atualizações menos frequentes, o CodeIgniter e o FuelPHP podem apresentar algumas brechas de segurança.

## 5 CONCLUSÕES

Esse projeto realizou um estudo das frameworks de desenvolvimento Web para identificar quais as frameworks mais indicadas para atender as necessidades do desenvolvedor de sistemas. O estudo foi realizado a partir de uma pesquisa bibliográfica das 5 frameworks de desenvolvimento web selecionadas, comparando as suas características e criando gráficos e tabelas para permitir uma análise comparativa visual e mais clara.

Em geral, Laravel e Symfony são frameworks PHP mais poderosas, com uma ampla gama de pacotes e componentes reutilizáveis, sendo adequadas para projetos maiores e mais complexos. CakePHP e FuelPHP também oferecem recursos integrados suficientes para facilitar o desenvolvimento de aplicativos web, sendo que este último apresenta o padrão HMVC, o que pode ser um ponto importante para alguns projetos, essas frameworks são indicadas para projetos mais simples. Por fim, CodeIgniter é a framework mais simples e leve, mas também é a menos poderosa e menos flexível em termos de recursos e componentes integrados, tendo destaque pela ausência do composer como gerenciador de pacotes e apresentou os piores resultados no benchmark.

Este trabalho teve como objetivo estudar as principais características de cada framework em PHP. Assim, foi apresentada uma síntese dos resultados por meio de tabelas, gráficos e considerações relevantes. Essas informações são instrumentos importantes para auxiliar os desenvolvedores na seleção da framework mais adequada para atender as necessidades do projeto. Em trabalhos futuros, poderia complementar esse estudo com pesquisas de campo, como entrevistas com desenvolvedores e empresas desenvolvedoras de sistemas web com PHP, a fim de identificar as percepções sobre cada característica das frameworks e verificar quais são as frameworks mais utilizadas na região.

## REFERÊNCIAS

- [1] Cuadrado, José Luis López; et al. **Analyzing best practices on Web development frameworks: The lift approach**. The Lift approach. Journal of Systems and Software, 2015.
- [2] w3techs. **Usage statistics of PHP for websites**. Disponível em: <https://w3techs.com/technologies/details/pl-php>. Acesso em: 7 jan. 2023.
- [3] PHP. **História do PHP**. Disponível em: [https://www.php.net/manual/pt\\_BR/history.php.php](https://www.php.net/manual/pt_BR/history.php.php). Acesso em 7 jan. 2023
- [4] PHP. **Supported Versions**. Disponível em: <https://www.php.net/supported-versions.php>. Acesso em 7 jan. 2023
- [5] SANTELICES, Raúl A; NUSSBAUM, Miguel. **A framework for the development of videogames**. Pontificia Universidad Católica de Chile: John Wiley & Sons, Ltd, 2001.
- [6] RIBEIRO, Claudio; **The Best PHP Frameworks for 2023**. Disponível em: <https://www.sitepoint.com/best-php-frameworks/#bestphpframeworkstouse>, Acesso em 10 jan, 2023.
- [7] LARAVEL. **Laravel**. Disponível em: <https://laravel.com/>. Acesso em:12 jan. 2023.
- [8] SURGUY, MAK. **History of Laravel PHP framework, Eloquence emerging**. Disponível em: <https://maxoffsky.com/code-blog/history-of-laravel-php-framework-eloquence-emerging/>. Acesso em: 12 jan. 2023.
- [9] Symfony. **What is Symfony**. Disponível em: <https://symfony.com/what-is-symfony> . Acesso em: 12 jan. 2023.
- [10] BANCER, Wojciech. **Symfony2 Essentials**. Birmingham, Reino Unido: Packt Publishing Ltd, 2015.
- [11] Symfony-docs. **Github**. Disponível em: <https://github.com/symfony/symfony-docs/blob/4cd6dc2825924c9569621bf749f168a7ba2a235d/contributing/community/releases.rst>. Acesso em 13 jan. 2023.
- [12] MAJIDA, Laaziri. **A comparative study of laravel and symfony PHP frameworks**. Abdelmalek Essaâdi University: Abdelmalek Essaâdi University, 2018.
- [13] CakePHP. **Github**. Disponível em: <https://github.com/cakephp/cakephppt>. Acesso em 14 jan. 2023.
- [14] SOLANKI, Nilesh; SHAH, Darshil; SHAH, Aakash. A Survey on different Framework of PHP. International Journal of Latest Technology in Engineering, Management & Applied

Science (IJLTEMAS), 2017.

[15] DĀSA, Rādhārādhya. **Learn CakePHP: With Unit Testing**. California: Apress, 2016.

[16] CodeIgniter User Guide. **CodeIgniter** Disponível em:

[https://codeigniter.com/user\\_guide/intro/index.html](https://codeigniter.com/user_guide/intro/index.html). Acesso em 17 jan. 2023.

[17] WIBAWA, S. C. et al. **Online test application development using framework CodeIgniter**. IOP Publishing Ltd, 2018. In: IOP Conf. Ser.: Mater. Sci. Eng., v. 296, p. 012041.

[18] Forum CodeIgniter. **Forum CodeIgniter**. Disponível em:

<https://forum.codeigniter.com/thread-74677.html>. Acesso em 17 jan. 2023.

[19] CodeIgnite. **CodeIgniter**. Disponível em: <https://codeigniter.com/>. Acesso em 17 jan. 2023.

[20] FuelPHP. **Github**. Disponível em: <https://github.com/fuel/fuel>. Acesso em 19 jan. 2023.

[21] A. Zurkiewicz, M. Miłosz. **SELECTING A PHP FRAMEWORK FOR A WEB APPLICATION PROJECT - THE METHOD AND CASE STUDY, INTED**. Lublin University of Technology, 2015.

[22] TechEmpower. **Web Framework Benchmarks**. Disponível em:

<https://www.techempower.com/benchmarks>, acessado em 02 de abril de 2023.

[23] Laaziri, M., Benmoussa, K., Khouliji, S., & Kerkeb, M. L. **A Comparative study of PHP frameworks performance**. Procedia Computer Science, 2019.

[24] MADURAPPERUMA, I. H. SHAFANA, M. S. SABANI, M. J. A. **State-of-Art Frameworks for Front-end and Back-end Web Development**. Faculty of Technology, South Eastern University of Sri Lanka, Sri Lanka, 2022.

[25] ABUTALEB, Hamza; TAMIMI, Abdelfatah; ALRAWASHDEH, Thamer. **Empirical Study of Most Popular PHP Framework**. In: 2021 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT), Amman, Jordan, 2021.

[26] LAAZIRI, Majida et al. **A comparative study of laravel and symfony PHP frameworks**. International Journal of Interactive Mobile Technologies, 2018.

[27] MOSTRÖ, Mathilda; RYRBERG, Sophie. **How to choose a web development framework: Analyzing best practices on the adoption of web frameworks**. Linköping University | Department of Computer and Information Science, 2022.

[28] HARIS, Norhaidah; YAHAYA, Jamaiah; BAKAR, Juliana Aida Abu. **PHP Frameworks Usability in Web Application Development**. International Journal of Advanced Computer Science and Applications, 2019.