

**Programação Web**

**Prof. Ms. Vinícius Magnus - @vinimagnus**

**Atividade Semipresencial**

Faça uma pesquisa relatando quais são os Frameworks de desenvolvimento mais utilizados no PHP, considere também algumas características como aplicabilidade (exemplo um focados em API), facilidade de aprendizagem, maior suporte de documentação e comunidade, tecnologias e padrões mais modernos.

Relate uma pequena descrição de cada uma e escolha um para proceder alguns testes iniciais(instalação e configuração), após relatar como foram estes testes.

**CodeIgniter**

O CodeIgniter é um framework PHP que utiliza arquitetura Model View Controller. Em termos leigos, isso quer dizer que o framework utiliza componentes distintos para lidar com tarefas de desenvolvimento específicas.

Este método é popular entre desenvolvedores porque permite a criação de aplicações altamente escaláveis sem ocupar muito espaço.

Se você é iniciante com frameworks PHP, o CodeIgniter pode ser uma boa opção por ser relativamente fácil de aprender. A plataforma ainda possui um ótimo manual, o que facilita o aprendizado.

O CodeIgniter também oferece uma performance sólida, o que o torna perfeito para criação de aplicações leves para servidores modestos.

**Características**

* Framework leve e focado em desempenho.
* Fácil aprendizado graças à simplicidade do framework e manual detalhado.
* Crie apps escaláveis com a arquitetura baseada em MVC do framework.

Existem muitas vantagens de usar o CodeIgniter. Mas não podemos considerar este o melhor framework porque tem seus pontos negativos.

Por exemplo, suas atualizações são um tanto irregulares, o que não o torna a melhor opção para quem procura um padrão máximo de segurança.

**Laravel**

Quando se fala de framework PHP, um dos nomes mais populares é o Laravel. Esse framework é conhecido por sua sintaxe elegante, que é fácil de aprender e prazeroso de trabalhar.

Com o Laravel, você começa a trabalhar nos seus projetos rapidamente. Você também consegue economizar muito tempo e trabalho com acesso a funções como autenticação de usuário, gerenciamento de seção, e cache.

No geral, o Laravel possui todas as ferramentas que você precisa para construir uma aplicação PHP moderna.

Se você está interessado nos motivos técnicos que tornam o Laravel uma boa opção, é fácil descobrir. Primeiro que a essência do Laravel é sólida em relação a performance. E você pode estender as funcionalidades do framework com extensões.

O Laravel também faz integração com serviços de terceiros, como o Amazon Web Services (AWS), que permite o desenvolvimento de aplicações altamente escaláveis.

Para tarefas em longo prazo, você pode programar para serem executadas de forma assíncrona, o que melhora ainda mais o desempenho.

**Características**

* Um framework que se orgulha de sua sintaxe elegante.
* Extensão de funcionalidades com complementos.
* Funções nativas para trabalhar com gerenciamento de usuários, cache e mais.
* Integração com plataformas externas, como o AWS.
* Execução de tarefas assíncronas para melhor performance.

Por último, o Laravel tem uma comunidade ativa. O que significa que obter ajuda não é uma tarefa difícil. Se essa é a sua primeira vez usando um framework, o Laravel se torna uma ótima opção.

**Symfony**

Se houvesse uma competição pelo título de melhor framework PHP, o Symfony estaria na disputa com o CodeIgniter e Laravel em termos de popularidade. Quando você começa a conhecer melhor o framework, logo descobre o motivo.

Além de ter um nome legal, o Symfony é altamente flexível. Ele possui um pacote e sistema de componentes que permitem a escolha exata das funções PHP que você precisa, ou simplesmente pode-se usar o framework completo.

Para garantir que sua aplicação será executada da maneira que deve ser, o framework possui uma funcionalidade de testes. Mas baseado na nossa experiência, as aplicações construídas com o Symfony não possuem a melhor performance.

Além de que o framework possui um processo de aprendizado mais complexo do que as opções que vimos até agora.

**Características**

* Framework altamente flexível que permite a escolha de componentes individuais.
* Vantagem de ter uma funcionalidade de testes nativa.
* Excelente manual para aprendizado.

O Symfony pode ser uma boa opção se você se importa com modularidade. Você pode utilizar apenas os componentes que deseja ao invés da biblioteca completa. Isso faz com que este seja o melhor framework, se você precisa apenas de uma ajudinha.

**Zend**

Muitas pessoas referem-se ao Zend como um framework ‘cola’, que é uma maneira de se referir à sua natureza baseada em componentes.

O Zend é um framework orientado a objetos, baseado em MVC que permite que você carregue apenas componentes que você deseja como bibliotecas individuais.

Em outras palavras, o Zend permite que você foque apenas nos componentes e funções que você deseja e ignore todo o resto.

Graças a essa abordagem e à natureza de orientação ao objeto do framework, você deve poder reutilizar muitos dos códigos que já escreveu, o que sempre é uma boa notícia.

E mais, é simples integrar a plataformas com bibliotecas externas para estender ainda mais suas funcionalidades.

**Características**

* Framework PHP orientado ao objeto com arquitetura MVC.
* Reutilização de seus códigos graças ao design da plataforma.
* Fácil integração com bibliotecas externas.
* Utilização apenas de componentes desejados.

Mas o Zend possui alguns pontos negativos. Por exemplo, a plataforma é difícil de aprender. Na nossa experiência, se você está saindo do Laravel, por exemplo, para o Zend, é bastante desafiador e até intimidante se for sua primeira vez trabalhando com frameworks.

Ainda assim, o Zend é um ótimo framework se você tem experiência com programação e gosta de códigos orientados ao objeto.