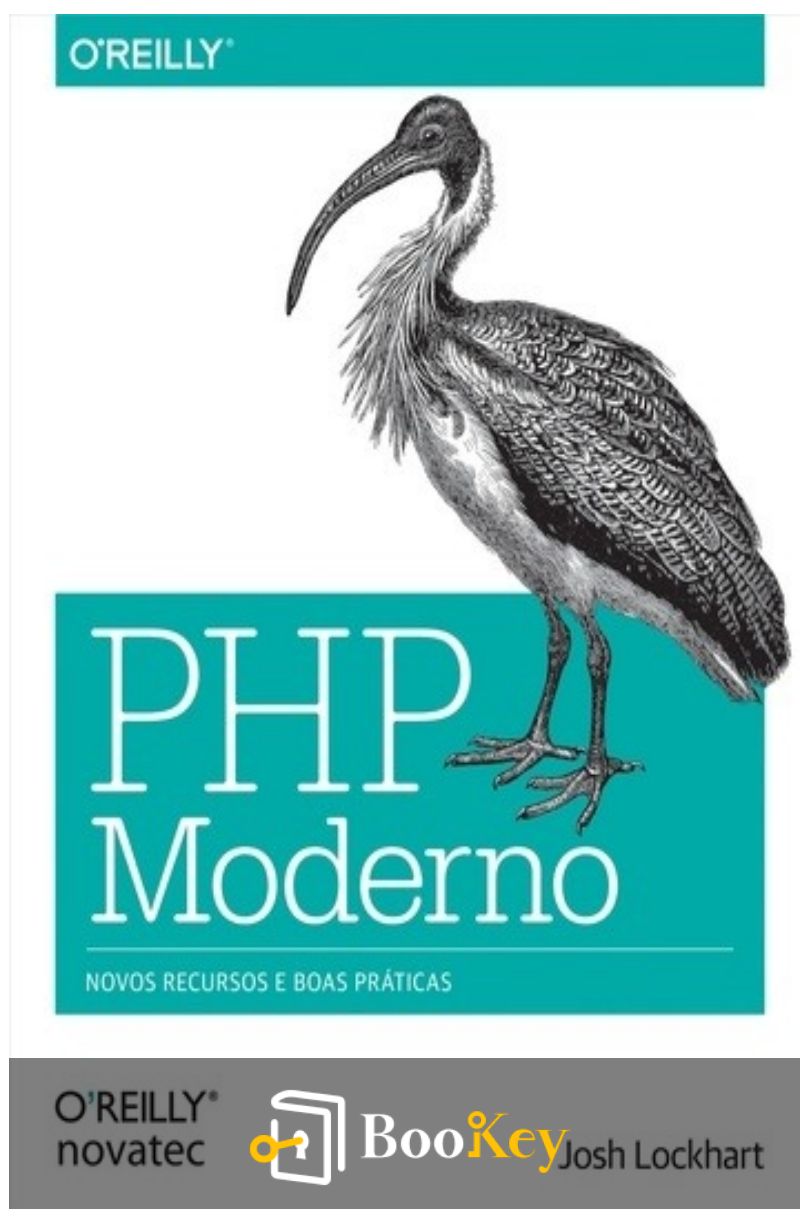


# PHP Moderno PDF

Josh Lockhart



Mais Livros Grátis



Escaneie para baixar



Ouça-o

# PHP Moderno

Aprimore suas habilidades em PHP com práticas e técnicas modernas.

Escrito por Bookey

[Saiba mais sobre o resumo de PHP Moderno](#)

[Ouvir PHP Moderno Audiolivro](#)

Mais Livros Grátis



Escaneie para baixar



[Ouçá-o](#)

## Sobre o livro

Descubra a evolução vibrante do PHP em "PHP Moderno", de Josh Lockhart, onde você explorará a transformação da linguagem em uma ferramenta robusta e madura, completa com programação orientada a objetos, namespaces e uma infinidade de bibliotecas reutilizáveis. Com base em sua experiência como criador do PHP The Right Way, Lockhart apresenta um guia prático que destaca os recursos modernos em ação. Este livro é voltado para aqueles que têm um conhecimento básico de PHP e estão ansiosos para aprofundar seu entendimento e aplicar as melhores práticas em arquitetura de aplicações, gerenciamento de bancos de dados, segurança, testes, depuração e implantação. Eleve suas habilidades em PHP e abraçe o futuro do desenvolvimento com este recurso essencial.

**Mais Livros Grátis**



Escaneie para baixar



**Ouça-o**

## Sobre o autor

Josh Lockhart é uma figura proeminente na comunidade PHP, conhecido por sua experiência e defesa das práticas modernas de desenvolvimento web. Como engenheiro de software e educador, ele se dedicou a promover o PHP como uma linguagem de programação robusta e versátil, contribuindo de forma significativa através de seu trabalho em frameworks e bibliotecas populares. Lockhart também é reconhecido por seu estilo de ensino envolvente, tendo compartilhado seu conhecimento em diversas conferências e plataformas online. Além de suas contribuições técnicas, ele é apaixonado por fomentar uma cultura de colaboração e inovação dentro da comunidade de desenvolvedores, tornando-se uma voz respeitada e um mentor tanto para desenvolvedores PHP aspirantes quanto para os mais experientes.

**Mais Livros Grátis**



Escaneie para baixar



**Ouça-o**



Ad



Escanear para baixar



# Experimente o aplicativo Bookey para ler mais de 1000 resumos dos melhores livros do mundo

Desbloqueie **1000+** títulos, **80+** tópicos

Novos títulos adicionados toda semana

Product & Brand

 Liderança & Colaboração

 Gerenciamento de Tempo

 Relacionamento & Comunicação

 Estratégia de Negócios

 Criatividade

 Memórias

 Conheça a Si Mesmo

 Psicologia

Empreendedorismo

 História Mundial

 Comunicação entre Pais e Filhos

 Autocuidado

 Mente

## Visões dos melhores livros do mundo

amento  
pos

Os 7 Hábitos das  
Pessoas Altamente  
Eficazes



Mini Hábitos



Hábitos Atômicos



O Clube das 5  
da Manhã



Como Fazer Amigos  
e Influenciar  
Pessoas



Com  
Não



Teste gratuito com Bookey



# Lista de conteúdo do resumo

Capítulo 1 : I. Recursos da Linguagem

Capítulo 2 : 1. O Novo PHP

Capítulo 3 : 2. Recursos

Capítulo 4 : II. Boas Práticas

Capítulo 5 : 3. Padrões

Capítulo 6 : 4. Componentes

Capítulo 7 : 5. Boas Práticas

Capítulo 8 : III. Implantação, Testes e Ajustes

Capítulo 9 : 6. Hospedagem

Capítulo 10 : 7. Provisionamento

Capítulo 11 : 8. Ajustando

Capítulo 12 : 9. Implantação

Capítulo 13 : 10. Testes

Capítulo 14 : 11. Profiling

Capítulo 15 : 12. HHVM e Hack

**Mais Livros Grátis**



Escaneie para baixar



Ouça-o

Capítulo 16 : 13. Comunidade

Capítulo 17 : A. Instalando PHP

Capítulo 18 : B. Ambientes de Desenvolvimento Local

**Mais Livros Grátis**

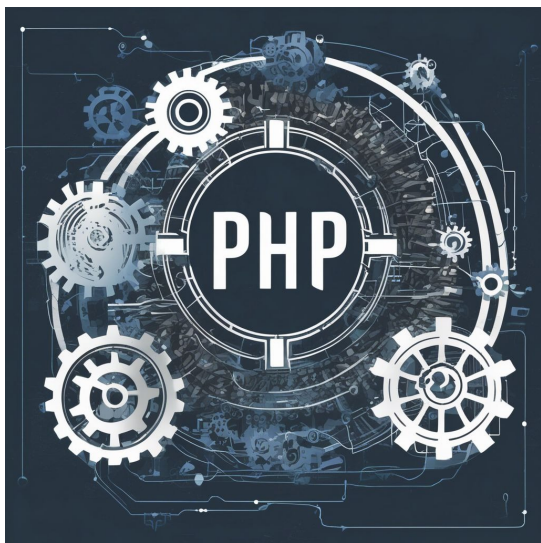


Escaneie para baixar



**Ouça-o**

# Capítulo 1 Resumo : I. Recursos da Linguagem



## Capítulo 1: O Novo PHP

### Visão Geral do PHP Moderno

O PHP está passando por uma transformação, evoluindo de uma linguagem de script tradicional para uma mais moderna, integrando recursos como namespaces, traits, closures e cache de opcode embutido. A mudança no ecossistema PHP é marcada por uma transição de grandes frameworks para componentes menores e especializados, facilitada pelo gerenciador de dependências Composer, que promove a

**Mais Livros Grátis**



Escaneie para baixar



**Ouça-o**



interoperabilidade entre os componentes PHP.

## Contexto Histórico

Criado originalmente por Rasmus Lerdorf em meados da década de 1990, o PHP começou como "Ferramentas de Página Pessoal", um conjunto de scripts CGI. Ao longo dos anos, passou por revisões significativas, tornando-se eventualmente o PHP 3, que introduziu uma sintaxe aprimorada e recursos orientados a objetos. O PHP 3 foi notavelmente bem-sucedido, alcançando 10% de participação no mercado de servidores web até o final de 1998.

## Práticas Atuais em PHP

As práticas de desenvolvimento em PHP moderno enfatizam o controle de versão com ferramentas como o Git, bem como o uso de ambientes de desenvolvimento locais por meio de virtualização. A adoção do Composer para gerenciamento de dependências e das PSRs para padrões de código agilizou o processo de desenvolvimento. As práticas de testes e implantação também avançaram, com ferramentas como PHPUnit e o gerenciador de processos FastCGI do PHP melhorando a qualidade e o desempenho.

**Mais Livros Grátis**



Escaneie para baixar



**Ouça-o**

## Especificação e Compatibilidade

A introdução de uma especificação oficial do PHP em 2014 marcou um marco significativo, fornecendo um guia para os desenvolvedores e garantindo compatibilidade entre diferentes mecanismos PHP, como o Zend Engine original e a Máquina Virtual HipHop do Facebook.

## Desenvolvimentos Futuros

A competição entre o Zend Engine e o HHVM levou a melhorias rápidas no desempenho do PHP. A linguagem de programação Hack, construída sobre o PHP, oferece tipagem estática e novas estruturas de dados, mantendo a compatibilidade com versões anteriores. As melhorias no Zend Engine e o surgimento do PHP 7 sinalizam um futuro vibrante para o desenvolvimento em PHP.

## Considerações Finais

A comunidade PHP está passando por uma fase inovadora, e a transição para práticas modernas é tanto empolgante quanto essencial para desenvolvedores novos e existentes. O livro

**Mais Livros Grátis**



Escaneie para baixar



**Ouça-o**

tem como objetivo navegar por esses novos conceitos e servir como um roteiro para abraçar as práticas do PHP moderno.

**Mais Livros Grátis**



Escaneie para baixar



**Ouça-o**

## Pensamento crítico

**Ponto chave:** A evolução do PHP em direção a práticas modernas é caracterizada pela sua transição de grandes frameworks para componentes especializados.

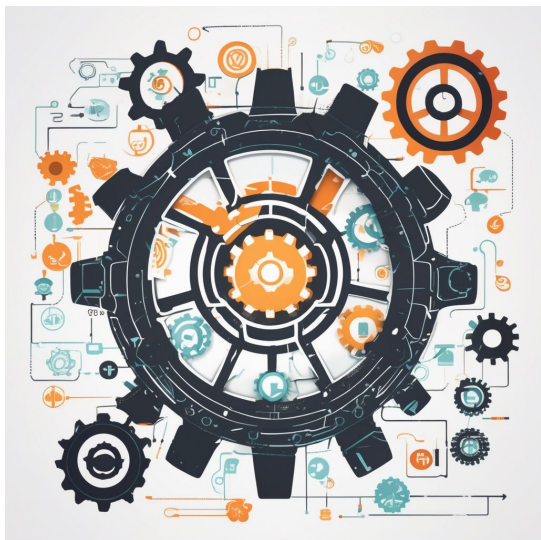
**Interpretação crítica:** Enquanto o autor apresenta um ponto de vista convincente sobre a transformação em curso dentro do ecossistema PHP, os céticos podem argumentar que essa mudança em direção a componentes menores e especializados pode não ser universalmente benéfica. A dependência do Composer e dos PSRs, embora melhore a interoperabilidade, levanta questões sobre a capacidade dos desenvolvedores individuais de navegar com sucesso nesse cenário cada vez mais modular. Além disso, os defensores de grandes frameworks podem afirmar que eles fornecem a estrutura e a estabilidade essenciais para projetos maiores, contrapondo-se à ideia de que componentes menores são o futuro. Críticos podem citar estudos sobre arquitetura de software e designs baseados em componentes (como os encontrados em "Designing Data-Intensive Applications" de Martin Kleppmann) para respaldar sua perspectiva de que os benefícios de



grandes frameworks podem superar as abordagens mais novas.



# Capítulo 2 Resumo : 1. O Novo PHP



Funcionalidade	Descrição
Visão Geral	PHP Moderno possui muitas funcionalidades que aprimoram sua capacidade como linguagem de script para aplicativos web e ferramentas de CLI, com foco em namespaces.
Namespaces	Organiza o código em uma hierarquia virtual, prevenindo colisões de nomes em projetos grandes. Declarados no início dos arquivos PHP.
Importação e Alias	PHP permite a importação e o uso de aliases para namespaces, resultando em um código mais limpo e gerenciável com o uso da palavra-chave <code>use</code> .
Código para uma Interface	Permite que o código dependa de comportamentos definidos por uma interface, tornando a integração mais suave.
Traits	Permitem a reutilização de código sem herança clássica, possibilitando a mistura de funcionalidades em várias classes.
Generators	Proporcionam iteração eficiente em termos de memória sobre conjuntos de dados ao gerar valores sob demanda desde o PHP 5.5.0.
Closures	Funções anônimas que encapsulam estado, úteis para callbacks e criadas com a palavra-chave <code>use</code> .
Zend OPcache	Armazena em cache o bytecode pré-compilado no PHP 5.5.0, melhorando o desempenho ao reduzir a sobrecarga.
Servidor HTTP Integrado	Permite o teste local de aplicações PHP sem software adicional de servidor web, iniciado via linha de comando.
Conclusão	Os recursos do PHP Moderno oferecem vastas melhorias para a criação de aplicações eficientes e mantíveis.

## Capítulo 2: Recursos

Mais Livros Grátis



Escaneie para baixar



Ouça-o

## Visão Geral

O PHP Moderno possui muitos novos recursos empolgantes que aprimoram sua capacidade como uma linguagem de script versátil para aplicações web e ferramentas de linha de comando. Este capítulo apresenta esses recursos, categorizando-os como essenciais ou úteis, com um foco especial em namespaces como um pilar do desenvolvimento moderno em PHP.

## Namespaces

Namespaces, introduzidos no PHP 5.3.0, permitem que os desenvolvedores organizem o código em uma hierarquia virtual, semelhante a sistemas de arquivos. Cada componente e framework PHP pode residir sob um namespace de fornecedor único, evitando conflitos com outros componentes que utilizam nomes de classe comuns.

-

## Importância

: Eles previnem colisões de nomes em grandes projetos com várias dependências, permitindo que múltiplas bibliotecas

**Mais Livros Grátis**



Escaneie para baixar



**Ouça-o**

coexistam pacificamente.

-

## **Declaração**

: Um namespace é declarado no topo de um arquivo PHP logo após a tag `<?php`.

-

## **Subnamespaces**

: Subnamespaces fornecem uma organização adicional dentro do namespace principal para uma melhor gestão do código.

## **Importação e Alias**

O PHP permite a importação e a criação de alias para namespaces, classes, interfaces, funções e constantes, a fim de tornar o código mais claro e gerenciável. Isso significa que você pode usar nomes mais curtos em vez de namespaces totalmente qualificados.

-

## **Forma de Importação**

: Use a palavra-chave `use` para incluir namespaces no topo dos seus arquivos PHP.

-

## **Alias**

: Você pode criar um alias para classes importadas para

**Mais Livros Grátis**



Escaneie para baixar



**Ouça-o**

reduzir a digitação e melhorar a legibilidade.

## Código para uma Interface

O uso de interfaces permite uma arquitetura flexível. Em vez de depender de implementações de classe específicas, seu código pode depender dos comportamentos definidos por uma interface, facilitando a integração com várias implementações.

-

### Exemplo

: Uma classe `DocumentStore` pode utilizar diferentes estratégias de busca de documentos que todas seguem uma interface comum `Documentable`.

## Traits

Introduzidos no PHP 5.4.0, traits permitem a reutilização de código sem as limitações da herança clássica. Eles possibilitam a mesclagem de funcionalidades em várias classes sem exigir um ancestral comum.

-

### Uso

: Traits são ideais para compartilhar métodos entre classes

Mais Livros Grátis



Escaneie para baixar



Ouça-o

não relacionadas enquanto mantêm as hierarquias de herança limpas.

## Geradores

Geradores são recursos poderosos desde o PHP 5.5.0, proporcionando iteração simples e eficiente em termos de memória sobre conjuntos de dados. Os geradores produzem valores sob demanda, em vez de pré-calcular e armazená-los em arrays.

-

## Eficiência de Memória

: Ideal para situações em que a memória é limitada, como a leitura de arquivos grandes.

## Closures

Closures, ou funções anônimas, foram introduzidas no PHP 5.3.0. Elas podem encapsular estado e são particularmente úteis para callbacks.

-

## Criação

: Uma closure é uma função sem nome que pode ser atribuída

Mais Livros Grátis



Escaneie para baixar



Ouça-o



a uma variável e invocada. Ela pode manter estado usando a palavra-chave ``use``.

## **Zend OPcache**

A partir do PHP 5.5.0, o Zend OPcache está integrado ao PHP para armazenar em cache bytecode pré-compilado, melhorando significativamente o desempenho das aplicações ao reduzir a sobrecarga do parsing e da compilação de scripts PHP a cada requisição.

-

### **Configuração**

: É importante habilitar e configurar corretamente o Zend OPcache no arquivo ``php.ini``.

## **Servidor HTTP embutido**

O PHP agora inclui um servidor web embutido para desenvolvimento local, permitindo testes fáceis de aplicações PHP sem a necessidade de software adicional de servidor web.

-

### **Uso**

: Inicie o servidor via linha de comando e aponte para o

**Mais Livros Grátis**



Escaneie para baixar



**Ouça-o**

diretório do seu projeto.

## Conclusão

Com uma variedade de novos recursos poderosos, o PHP Moderno oferece vastas melhorias e conveniências para o desenvolvimento. Compreender e utilizar esses recursos é fundamental para criar aplicações eficientes e manuteníveis. O próximo capítulo se aprofundará na adesão aos padrões da comunidade ao usar esses recursos.

**Mais Livros Grátis**



Escaneie para baixar



**Ouça-o**

## Exemplo

**Ponto chave:** Os namespaces ajudam a evitar colisões de nomes no desenvolvimento em PHP Moderno.

**Exemplo:** Imagine que você está trabalhando em um grande projeto com vários desenvolvedores, todos contribuindo com diferentes bibliotecas. Você se depara com uma classe chamada 'User' nas bibliotecas 'Blog' e 'ECommerce'. Ao usar namespaces, você pode definir essas classes como '\Blog\User' e '\ECommerce\User', organizando efetivamente seu código e evitando conflitos. Isso permite que você trabalhe sem problemas, sem se preocupar com qual classe 'User' você está se referindo, garantindo que sua aplicação permaneça robusta e fácil de manter.

Mais Livros Grátis



Escaneie para baixar



Ouça-o

# Capítulo 3 Resumo : 2. Recursos

Recurso	Descrição
Recursos do PHP Moderno	Melhorias para desenvolvedores que estão atualizando de versões mais antigas ou migrando de outras linguagens, aumentando a capacidade e a eficiência.
Namespaces	Organiza o código em uma hierarquia, reduzindo conflitos e auxiliando o autoload desde o PHP 5.3.0.
Uso de Namespaces	Permite codificação colaborativa sem interferências; previne conflitos de nomenclatura em grandes projetos.
Importação e Alias	Reduz a necessidade de nomes totalmente qualificados, melhorando a clareza e a manutenibilidade.
Interfaces	Atuam como contratos entre objetos, promovendo código flexível e melhor integração com bibliotecas.
Traits	Permitem reutilização de código entre classes sem ancestrais comuns, evitando duplicação e mantendo a estrutura.
Geradores	Criam iteradores que produzem valores sob demanda, melhorando a eficiência de memória.
Closures	Encapsulam comportamento e estado em funções para um código mais limpo e legível.
Melhorias de Performance	Zend OPcache armazena em cache o bytecode pré-compilado para melhorias significativas de desempenho.
Servidor HTTP Integrado	Facilita pré-visualizações de desenvolvimento local de aplicações, mas não é para uso em produção.
Conclusão	Incentivo para os desenvolvedores utilizarem esses recursos modernos para aumentar a produtividade e o desempenho.

## Resumo do Capítulo 3

### Introdução aos Recursos do PHP Moderno

O PHP Moderno inclui muitos novos recursos que são benéficos para os desenvolvedores, especialmente aqueles que estão atualizando de versões mais antigas ou migrando

Mais Livros Grátis



Escaneie para baixar



Ouça-o

de outras linguagens. Essas melhorias aumentam a capacidade e a eficiência do PHP para aplicações e ferramentas web. Entre os recursos mais notáveis estão namespaces, autoloading, geradores, closures, traits e otimizações de desempenho através de mecanismos de cache, como o Zend OPcache.

## Namespaces

Namespaces, introduzidos no PHP 5.3.0, permitem que os desenvolvedores organizem o código em uma hierarquia estruturada, reduzindo conflitos de nomes de classes e melhorando os processos de autoloading. Eles são essenciais para gerenciar códigos de múltiplos autores e garantir identificadores únicos para classes, funções e constantes. A declaração de namespaces é feita no início dos arquivos PHP, ajudando a evitar colisões de nomes em grandes projetos.

**Instalar o aplicativo Bookey para desbloquear  
texto completo e áudio**

**Mais Livros Grátis**



Escaneie para baixar



**Ouça-o**





Escanear para baixar



# Por que o Bookey é um aplicativo indispensável para amantes de livros



## Conteúdo de 30min

Quanto mais profunda e clara for a interpretação que fornecemos, melhor será sua compreensão de cada título.



## Clipes de Ideias de 3min

Impulsione seu progresso.



## Questionário

Verifique se você dominou o que acabou de aprender.



## E mais

Várias fontes, Caminhos em andamento, Coleções...

Teste gratuito com Bookey



# Capítulo 4 Resumo : II. Boas Práticas

Seção	Conteúdo
Parte II	Boas Práticas
Capítulo 3	Padrões
Tipos de Framework	Inclui frameworks macro (Symfony, Laravel), micro frameworks (Silex, Slim) e frameworks legados (CodeIgniter).
Introdução ao PHP-FIG	Formado em 2009 para melhorar a interoperabilidade entre frameworks PHP através de padrões voluntários (PSRs).
Foco na Interoperabilidade	1. Interfaces, 2. Autoloading, 3. Estilo de codificação.
Interfaces	Permitem que frameworks trabalhem com dependências de terceiros sem conhecimento interno.
Autoloading	Padroniza o carregamento de classes em tempo de execução entre frameworks.
Estilo de Codificação	Promove estilos comuns para melhor legibilidade e redução da curva de aprendizado.
Visão Geral do PSR	Problemas comuns abordados pelos PSRs ajudam a unificar práticas em PHP.
PSR-1	Diretrizes básicas de estilo de código incluindo tags PHP padrão, codificação UTF-8, funcionalidade de arquivo único e convenções de nomenclatura.
PSR-2	Estilo de código rigoroso para colaboração: indentação de quatro espaços, quebras de linha Unix, limites de comprimento de linha e nomenclatura consistente.
PSR-3	Define uma interface de logger padrão para frameworks utilizarem componentes de logging de terceiros.
PSR-4	Padroniza estratégias de autoloading, simplificando o carregamento de classes ao mapear namespaces para diretórios do sistema de arquivos.
Resumo	As recomendações do PHP-FIG melhoram o ambiente de desenvolvimento PHP através de padronização e eficiência.

## Parte II: Boas Práticas

### Capítulo 3: Padrões

Mais Livros Grátis



Escaneie para baixar



Ouça-o

O ecossistema PHP inclui uma variedade de frameworks—frameworks macro como Symfony e Laravel, micro frameworks como Silex e Slim, e frameworks legados como CodeIgniter. Frameworks mais antigos desenvolvidos de forma isolada resultam em comunicação ineficiente e criatividade limitada para os desenvolvedores. Felizmente, a comunidade PHP fez uma transição para um ecossistema mais distribuído e interoperável com a ajuda do Grupo de Interoperabilidade de Frameworks PHP (PHP-FIG).

## PHP-FIG em Ação

Em 2009, desenvolvedores do php|tek iniciaram discussões para aprimorar a interoperabilidade entre frameworks PHP, levando à criação do PHP-FIG. Este grupo tem como objetivo estabelecer padrões voluntários, conhecidos como Recomendações de Padrões PHP (PSRs), para facilitar uma melhor comunicação e compartilhamento de código entre frameworks. Recomendações finais são frequentemente adotadas por frameworks PHP populares, incentivando os desenvolvedores a interagir com o PHP-FIG para feedback e melhorias.

## Interoperabilidade de Frameworks

**Mais Livros Grátis**



Escaneie para baixar



**Ouça-o**

O PHP-FIG foca em três aspectos críticos para a interoperabilidade: interfaces, autoloading e estilo de codificação:

-

### **Interfaces:**

Permitem que frameworks assumam que certos métodos são implementados por dependências de terceiros sem precisar conhecer seu funcionamento interno.

-

### **Autoloading:**

Processo simplificado para carregar automaticamente classes em tempo de execução, afastando-se da necessidade de cada framework implementar seu método de carregamento único.

-

### **Estilo:**

Um estilo de codificação comum reduz a curva de aprendizado para colaboradores e promove uma melhor legibilidade do código.

## **O Que É um PSR?**

As Recomendações de Padrões PHP (PSRs) abordam problemas comuns entre frameworks PHP. PSR-1, PSR-2,

**Mais Livros Grátis**



Escaneie para baixar



**Ouça-o**

PSR-3 e PSR-4 são recomendações principais que focam em estilo de código, interfaces de logging e padrões de autoloading. Elas ajudam a unificar e simplificar práticas de código na comunidade PHP.

## **PSR-1: Estilo de Código Básico**

O PSR-1 estabelece diretrizes fundamentais de codificação que incluem:

- Uso de tags PHP padrão (<?php e <?=?).
- Codificação UTF-8 sem BOM.
- Funcionalidade em um único arquivo—ou definir símbolos ou realizar uma ação.
- Suporte para padrões de autoloading do PSR-4.
- Manter nomes de classes em CamelCase, constantes em maiúsculas e nomes de métodos em camelCase.

## **PSR-2: Estilo de Código Rigoroso**

Além do PSR-1, o PSR-2 oferece diretrizes mais rigorosas para uma melhor colaboração entre colaboradores diversos:

- Indentação com quatro espaços.
- Uso de quebras de linha do Unix.
- Limitar o comprimento da linha.





- Consistência na nomeação, visibilidade, e estruturas de controle, assim como na colocação de chaves.

### **PSR-3: Interface de Logger**

O PSR-3 define uma interface padrão para logging, especificando métodos comuns que os logs devem implementar, permitindo que frameworks deleguem o logging a componentes de terceiros como o Monolog.

### **PSR-4: Autoloaders**

O PSR-4 introduz estratégias de autoloading padronizadas, garantindo uma integração contínua entre componentes PHP. Ele mapeia namespaces para diretórios do sistema de arquivos, simplificando o carregamento de componentes sem a necessidade de autoloaders individuais para cada componente.

Em resumo, as recomendações do PHP-FIG são essenciais para promover um ambiente de desenvolvimento PHP mais eficiente, interoperável e padronizado, estabelecendo assim as bases para melhorias futuras no ecossistema PHP.

**Mais Livros Grátis**



Escaneie para baixar



**Ouça-o**

## Exemplo

**Ponto chave:** Adote as Recomendações de Padrões PHP (PSRs) para uma melhor colaboração.

**Exemplo:** Ao seguir as PSRs, você permite que seus projetos se comuniquem de forma perfeita com diferentes frameworks, tornando a codificação menos difícil e mais agradável.

Mais Livros Grátis



Escaneie para baixar



Ouça-o

# Capítulo 5 Resumo : 3. Padrões

Seção	Resumo
Introdução aos Frameworks e Componentes PHP	O ecossistema PHP possui vários frameworks (macro, micro, legado). Frameworks mais antigos muitas vezes carecem de interoperabilidade, limitando a integração de componentes entre diferentes frameworks.
Grupo de Interoperabilidade de Frameworks PHP (PHP-FIG)	Estabelecido em 2009 para aprimorar a comunicação e a interoperabilidade entre frameworks PHP, criando Recomendações de Padrões PHP (PSRs) para compartilhamento de código e funcionalidade.
Padrões Chave para Interoperabilidade	<p>Interfaces: Permitem a interação com componentes externos sem entender suas implementações.</p> <p>Autoloading: O PSR-4 padroniza o autoloading para gerenciar múltiplos componentes.</p> <p>Estilo de Código: PSRs (1 e 2) garantem um estilo de código consistente para legibilidade e contribuição.</p>
Visão Geral dos PSRs	<p>PSR-1: Diretrizes básicas de estilo de código.</p> <p>PSR-2: Estilo de codificação rígido para consistência.</p> <p>PSR-3: Interface de logger para métodos de registro.</p> <p>PSR-4: Padrão de autoloading baseado em namespaces.</p>
Criando Componentes PHP	PHP Moderno incentiva a construção de aplicações a partir de componentes especializados e interoperáveis que se concentram em problemas específicos, promovendo velocidade e reduzindo a redundância de código.
Características de Bons Componentes PHP	<p>Focados, resolvendo problemas específicos com eficácia.</p> <p>Pequenos, mantendo código mínimo.</p> <p>Cooperativos, evitando poluição de namespace.</p> <p>Bem-testados, mantendo uma cobertura abrangente.</p> <p>Bem-documentados, fornecendo instruções e documentação claras.</p>
Usando o Composer para Gestão de Dependências	O Composer simplifica a integração de componentes PHP e a gestão de dependências, permitindo inclusão fácil e autoloading em projetos.
Construindo e Distribuindo Componentes PHP	Desenvolva componentes com estrutura clara e convenções de nomenclatura únicas, documente de forma abrangente e envie para o Packagist para acessibilidade.
Conclusão	A transição para componentes PHP sinaliza uma evolução nas práticas, permitindo aplicações eficientes por meio dos PSRs e ferramentas como Composer, enquanto promove o engajamento e a colaboração da comunidade.



# **Resumo do Capítulo 5: Padrões e Interoperabilidade do PHP Moderno**

## **Introdução aos Frameworks e Componentes PHP**

O ecossistema PHP conta com uma ampla variedade de frameworks, incluindo frameworks macro como Symfony e Laravel, micro frameworks como Silex e Slim, e frameworks legados como CodeIgniter. Contudo, frameworks mais antigos frequentemente carecem de interoperabilidade, o que limita a capacidade dos desenvolvedores de utilizar componentes de outros frameworks sem uma extensa integração personalizada.

## **Grupo de Interoperabilidade de Frameworks PHP (PHP-FIG)**

Para solucionar essas ineficiências, o Grupo de Interoperabilidade de Frameworks PHP (PHP-FIG) foi estabelecido em 2009. Sua missão é aprimorar a comunicação e a interoperabilidade entre frameworks PHP através do desenvolvimento e recomendação de padrões,

**Mais Livros Grátis**



Escaneie para baixar



**Ouça-o**

conhecidos como Recomendações de Padrões PHP (PSRs). Esses PSRs servem como diretrizes para facilitar o compartilhamento de código e funcionalidade entre diferentes frameworks.

## **Padrões Chave para Interoperabilidade**

1.

### **Interfaces**

: Interfaces definidas pelo PSR permitem que frameworks PHP interajam suavemente com componentes externos sem a necessidade de entender suas implementações subjacentes. Isso promove o compartilhamento de componentes especializados, como log de atividades ou manipulação de requisições HTTP.

2.

### **Autoloading**

: O PSR-4 introduz um mecanismo de autoloading padronizado, facilitando a gestão e o carregamento de classes a partir de múltiplos componentes PHP sem depender de autoloaders específicos de cada framework.

3.

### **Estilo de Código**

: Os PSRs delineiam convenções de estilo de código

**Mais Livros Grátis**



Escaneie para baixar



**Ouça-o**

consistentes (PSR-1 e PSR-2) que garantem legibilidade e familiaridade, reduzindo as barreiras para novos colaboradores e mantendo projetos de equipe coesos.

## Visão Geral dos PSRs

-

### PSR-1

: Diretrizes básicas de estilo de código (tags PHP, codificação e princípios básicos de codificação).

-

### PSR-2

: Diretrizes rigorosas de estilo de codificação para garantir consistência entre projetos.

-

### PSR-3

: Interface de log que define métodos padrão para funcionalidade de log.

-

### PSR-4

: Padrão de autoloading para carregar automaticamente arquivos de classes com base em namespaces.

## Criando Componentes PHP

Mais Livros Grátis



Escaneie para baixar



Ouça-o

O PHP Moderno enfatiza a construção de aplicações a partir de componentes especializados e interoperáveis, em vez de depender exclusivamente de grandes frameworks. Os componentes se concentram em resolver problemas específicos, como enviar requisições HTTP ou analisar arquivos CSV. Essa abordagem modular permite um desenvolvimento mais ágil e menos codificação redundante.

## **Características de Bons Componentes PHP**

-

### **Focado**

: Resolve bem um único problema específico.

-

### **Pequeno**

: Mantém o código o mais minimalista possível para realizar sua função.

-

### **Cooperativo**

: Interage bem com outros componentes e evita poluição de namespace.

-

### **Bem testado**

**Mais Livros Grátis**



Escaneie para baixar



**Ouça-o**



: Mantém testes abrangentes e cobertura adequada.

-

### **Bem documentado**

: Fornece instruções claras e documentação em linha para facilitar o uso e a colaboração.

## **Usando o Composer para Gerenciamento de Dependências**

O Composer é uma ferramenta crucial para gerenciar componentes PHP e suas dependências. Ele permite que os desenvolvedores incluam e carreguem componentes facilmente nos projetos, simplificando enormemente a integração de várias bibliotecas.

## **Construindo e Distribuindo Componentes PHP**

Para criar um componente PHP:

- Desenvolva seu componente com uma estrutura clara (por exemplo, ``src/``, ``tests/``, ``composer.json``, etc.).
- Use uma convenção de nomenclatura única de fornecedor/pacote para evitar conflitos no ecossistema.
- Documente seu componente de forma abrangente e inclua um arquivo README para os usuários.



- Submeta o componente ao Packagist para fácil acessibilidade.

## Conclusão

A mudança em direção a componentes PHP marca uma evolução significativa nas práticas de desenvolvimento PHP. Ao aproveitar os PSRs e ferramentas como o Composer, os desenvolvedores podem criar aplicações flexíveis, manuteníveis e eficientes, minimizando a necessidade de reinventar problemas já resolvidos. Participe da comunidade PHP contribuindo com componentes e melhorando bibliotecas existentes, promovendo uma melhor colaboração e compartilhamento de conhecimento no desenvolvimento em PHP.

**Mais Livros Grátis**



Escaneie para baixar



**Ouça-o**

## Exemplo

**Ponto chave:** A importância da interoperabilidade em frameworks PHP.

**Exemplo:** Imagine que você está desenvolvendo uma nova aplicação web usando Laravel, mas deseja integrar um pacote de logging que foi originalmente criado para Symfony. Graças às Recomendações de Padrões PHP (PSRs), especialmente a PSR-3 que define uma interface de logger, você pode usar esse componente de logging sem se preocupar com os detalhes de sua implementação. Essa interoperabilidade permite que você se concentre em desenvolver ótimas funcionalidades para sua aplicação, em vez de passar horas escrevendo código de integração personalizado.

Mais Livros Grátis



Escaneie para baixar



Ouça-o

## Pensamento crítico

**Ponto chave:** A mudança para um ecossistema de componentes PHP modular representa um avanço significativo.

**Interpretação crítica:** Lockhart enfatiza que o desenvolvimento de PHP moderno foca cada vez mais no uso de componentes interoperáveis, em vez de grandes frameworks, promovendo eficiência e práticas de codificação colaborativa. No entanto, pode-se questionar se essa mudança realmente beneficia todos os desenvolvedores; de fato, embora essa abordagem modular permita soluções especializadas e um desenvolvimento mais ágil, sua dependência de uma variedade de componentes pode complicar projetos, especialmente para aqueles menos familiarizados com a gestão de dependências através de ferramentas como o Composer. Críticos podem argumentar que essa modularidade poderia levar à fragmentação dentro do ecossistema PHP, tornando mais difícil para novos desenvolvedores navegar e contribuir de forma eficaz. A importância de integrar tanto frameworks quanto componentes individuais deve ser reconhecida, como evidenciado por fontes como 'The Pragmatic



Programmer' de Hunt e Thomas, que discutem o equilíbrio entre modularização e coesão do sistema.

# Capítulo 6 Resumo : 4. Componentes

## Capítulo 6: Componentes

### Visão Geral dos Componentes

O PHP Moderno enfatiza a construção de aplicações usando componentes especializados e interoperáveis, em vez de depender exclusivamente de frameworks monolíticos como Laravel ou Symfony. Os desenvolvedores são incentivados a aproveitar componentes existentes para resolver problemas específicos de forma eficiente, sem reinventar a roda.

### Benefícios do Uso de Componentes

-

#### **Eficiência**

: Existem inúmeros componentes pré-construídos disponíveis. Por exemplo, por que criar uma biblioteca para requisições HTTP quando o Guzzle já existe?

-

#### **Integração**

**Mais Livros Grátis**



Escaneie para baixar



**Ouça-o**

: Componentes especializados facilitam a combinação de várias bibliotecas, uma vez que seguem interfaces comuns.

-

## **Flexibilidade**

: Os desenvolvedores podem trocar componentes livremente dependendo das necessidades do projeto, ao contrário da estrutura rígida frequentemente imposta por um framework.

## **Definição de Componentes**

Um componente é uma coleção de código (classes, interfaces, traits) projetada para resolver um problema específico dentro de uma aplicação PHP. Características de bons componentes incluem:

-

### **Foco preciso**

: Projetado para lidar excepcionalmente bem com uma única tarefa.

**Instalar o aplicativo Bookey para desbloquear texto completo e áudio**

**Mais Livros Grátis**



Escaneie para baixar



**Ouça-o**



Ad



Escanear para baixar



App Store  
Escolha dos Editores



22k avaliações de 5 estrelas

## Feedback Positivo

Afonso Silva

...cada resumo de livro não só  
..., mas também tornam o  
...divertido e envolvente. O  
...tizou a leitura para mim.

**Fantástico!**



Estou maravilhado com a variedade de livros e idiomas  
que o Bookey suporta. Não é apenas um aplicativo, é  
um portal para o conhecimento global. Além disso,  
ganhar pontos para caridade é um grande bônus!

Brígida Santos

F



O  
só  
o  
O

na Oliveira

...correr as  
...ém me dá  
...omprar a  
...ar!

**Adoro!**



Usar o Bookey ajudou-me a cultivar um hábito de  
leitura sem sobrecarregar minha agenda. O design do  
aplicativo e suas funcionalidades são amigáveis,  
tornando o crescimento intelectual acessível a todos.

Duarte Costa

**Economiza tempo!**



O Bookey é o meu apli  
crescimento intelectual  
perspicazes e lindame  
um mundo de conheci

**Aplicativo incrível!**



Eu amo audiolivros, mas nem sempre tenho tempo para  
ouvir o livro inteiro! O Bookey permite-me obter um resumo  
dos destaques do livro que me interessa!!! Que ótimo  
conceito!!! Altamente recomendado!

Estevão Pereira

**Aplicativo lindo**



Este aplicativo é um salva-vidas para  
de livros com agendas lotadas. Os re  
precisos, e os mapas mentais ajudar  
o que aprendi. Altamente recomend

Teste gratuito com Bookey



# Capítulo 7 Resumo : 5. Boas Práticas

## Resumo do Capítulo 7: Boas Práticas em PHP Moderno

### Visão Geral

Este capítulo descreve um conjunto de boas práticas práticas para a criação de aplicações em PHP, enfatizando a importância de usar as ferramentas adequadas para melhorar o desempenho, a segurança e a estabilidade da aplicação. Reflete sobre a evolução do PHP e a necessidade de adaptar as práticas de acordo com as versões atuais do PHP.

### Sanitização, Validação e Escape

- Nunca confie em entradas de fontes externas não controladas (por exemplo, `$_GET`, `$_POST`, `$_COOKIE`).

- 

### Sanitize Input

: Proteja sua aplicação escapando ou removendo caracteres inseguros usando funções como ``htmlentities()`.

Mais Livros Grátis



Escaneie para baixar



Ouça-o

-

## **Validar Dados**

: Certifique-se de que os dados de entrada atendem aos formatos especificados (por exemplo, validando e-mails ou números de telefone) usando funções como ``filter_var()``.

-

## **Escape Output**

: Use métodos de escape apropriados para evitar a renderização de código malicioso ao exibir a saída.

## **Gerenciamento de Senhas**

-

### **Nunca Conheça as Senhas dos Usuários**

: Armazene apenas senhas hash para prevenir vazamentos de dados.

-

### **Nunca Restrinja as Senhas dos Usuários**

: Permita que os usuários criem senhas seguras sem restrições desnecessárias.

-

### **Nunca Envie Senhas dos Usuários por E-mail**

: Use links tokenizados para recuperação de senhas em vez de enviar credenciais em texto simples.

**Mais Livros Grátis**



Escaneie para baixar



**Ouça-o**

-

## **Hash as Senhas dos Usuários**

: Utilize algoritmos fortes como bcrypt para hash de senhas a fim de aumentar a segurança. O PHP 5.5+ tem funções integradas para simplificar esse processo.

## **Interação com o Banco de Dados usando PDO**

-

### **Use PDO para Interfaces de Banco de Dados**

: A extensão PDO (PHP Data Objects) permite uma interação flexível com diversos sistemas de banco de dados usando uma interface consistente.

-

### **Instruções Preparadas**

: Utilize instruções preparadas para prevenir injeção de SQL, vinculando com segurança a entrada do usuário aos comandos SQL.

-

### **Transações**

: Agrupe múltiplas operações de banco de dados dentro de uma transação para garantir atomicidade e integridade dos dados.

**Mais Livros Grátis**



Escaneie para baixar



**Ouçá-o**

## Manuseio de Data e Hora

- Utilize as classes `DateTime`, `DateInterval` e `DateTimeZone` para manipulação precisa de data e hora, evitando armadilhas comuns relacionadas a fusos horários e formatos.

## Tratamento de Erros e Exceções

- Empregue exceções como uma forma moderna de lidar com erros, permitindo que os desenvolvedores antecipem e resolvam problemas de forma elegante.
- Defina manipuladores globais de exceções e erros para registrar e gerenciar erros inesperados de forma eficaz.
- Considere usar ferramentas como Monolog para logging robusto em ambientes de produção, garantindo que o tratamento de erros não exponha dados do usuário.

## Streams e Filtros Personalizados

- Aproveite streams para um manuseio eficiente de dados, permitindo a leitura e escrita a partir de várias fontes de forma consistente.
- Crie e registre filtros de stream personalizados para modificar dados em tempo real, aprimorando a manipulação





dos dados do stream.

## Notas Finais

Seguindo essas boas práticas, você pode garantir que suas aplicações em PHP sejam seguras, sustentáveis e amigáveis ao usuário, adaptando-se ao cenário em constante evolução do desenvolvimento web.

**Mais Livros Grátis**



Escaneie para baixar



**Ouça-o**

## Pensamento crítico

**Ponto chave:** Sanitização, Validação e Escape

**Interpretação crítica:** A afirmação de que os desenvolvedores devem sanitizar, validar e escapar rigorosamente as entradas do usuário é fundamental para a segurança das aplicações. No entanto, embora essas práticas de Josh Lockhart em 'PHP Moderno' ofereçam uma estrutura robusta, é vital lembrar que nenhum método é infalível. A eficácia dessas técnicas pode variar dependendo das habilidades de implementação do desenvolvedor e da natureza dinâmica das ameaças à segurança. É benéfico considerar visões contrastantes, como as encontradas no guia de testes da OWASP, que sugere uma abordagem em constante evolução para a segurança de aplicações web, indicando que confiar apenas em métodos pré-definidos pode levar à complacência.

Mais Livros Grátis



Escaneie para baixar



Ouça-o



# Capítulo 8 Resumo : III. Implantação, Testes e Ajustes

## Capítulo 8: Hospedagem de Aplicações PHP

### Introdução

Hospedar uma aplicação PHP é essencial para que os usuários possam acessá-la. Existem quatro métodos principais de hospedagem: servidores compartilhados, servidores privados virtuais (VPS), servidores dedicados e plataformas como serviço (PaaS). Cada opção possui benefícios distintos, adaptados às diferentes necessidades e orçamentos dos usuários.

### Servidor Compartilhado

-

#### Custo

: R\$1–10/mês

-

Mais Livros Grátis



Escaneie para baixar



Ouça-o

## Visão Geral

: Servidores compartilhados são a opção menos cara, mas não são amigáveis para desenvolvedores. Múltiplos usuários compartilham os recursos do servidor, o que pode levar a problemas de desempenho.

-

## Limitações

: Personalização mínima, falta de acesso SSH e vulnerabilidade a scripts mal projetados de outros usuários afetam sua aplicação.

-

## Recomendação

: Adequado para projetos muito simples com baixas exigências, mas geralmente não é aconselhável para sites empresariais ou aplicações populares.

## Servidor Privado Virtual (VPS)

-

### Custo

: R\$10–100/mês

-

## Visão Geral

: Um VPS funciona como um servidor dedicado, mas é

Mais Livros Grátis



Escaneie para baixar



Ouça-o

virtual. Ele aloca seus próprios recursos, como memória, CPU e largura de banda.

-

## **Vantagens**

: Oferece acesso root SSH, a capacidade de instalar qualquer software necessário e recursos mais significativos do que servidores compartilhados. Adequado para a maioria das aplicações que necessitam de confiabilidade e flexibilidade.

-

## **Recomendação**

: Ideal para desenvolvedores que precisam de controle e recursos; pode ser escalado conforme a demanda.

## **Servidor Dedicado**

-

## **Custo**

: Centenas de dólares/mês

-

## **Visão Geral**

: Um servidor físico totalmente configurável gerenciado pela empresa de hospedagem.

-

## **Desempenho**

**Mais Livros Grátis**



Escaneie para baixar



**Ouça-o**

: Proporciona o mais alto desempenho para aplicações exigentes. Requer um esforço significativo para instalação e manutenção.

-

## **Tipos**

: Pode ser não gerenciado ou gerenciado, o que afeta o nível de suporte fornecido.

-

## **Recomendação**

: Vale a pena considerar quando um VPS se torna muito caro devido à demanda por recursos.

## **Plataforma como Serviço (PaaS)**

-

## **Custo**

: R\$10–100/mês

-

## **Visão Geral**

: O PaaS facilita a implantação rápida sem gestão de servidores necessária; ideal para desenvolvedores que preferem facilidade de uso.

-

## **Fornecedores**

**Mais Livros Grátis**



Escaneie para baixar



**Ouça-o**

: Exemplos incluem Heroku, AWS Elastic Beanstalk e Google App Engine.

-

## **Vantagens**

: Oferece recursos escaláveis e opções de gerenciamento amigáveis ao usuário.

-

## **Recomendação**

: Melhor para desenvolvedores que buscam conveniência e redução da sobrecarga de gerenciamento.

## **Escolhendo um Plano de Hospedagem**

- Avalie cuidadosamente as necessidades e escolha com base nos requisitos atuais. Os planos podem ser escalados conforme necessário.

- Para pequenas aplicações, soluções PaaS são as mais eficientes, enquanto o VPS é melhor para aqueles que precisam de mais controle. Faça upgrade para servidores dedicados à medida que sua aplicação cresce.

## **Conclusão**

Independentemente da solução de hospedagem selecionada,

**Mais Livros Grátis**



Escaneie para baixar



**Ouça-o**

certifique-se de que suporte a versão mais recente do PHP e as extensões necessárias para o desempenho ideal da sua aplicação.

**Mais Livros Grátis**



Escaneie para baixar



Ouça-o



# Capítulo 9 Resumo : 6. Hospedagem

## Capítulo 9: Hospedagem

### Introdução à Hospedagem de Aplicações PHP

Para tornar sua aplicação PHP acessível aos usuários, é necessário hospedá-la em um servidor. Existem quatro opções principais de hospedagem: servidores compartilhados, servidores privados virtuais (VPS), servidores dedicados e plataformas como serviço (PaaS). Cada opção varia em benefícios, adequação e custo.

### Servidor Compartilhado

-

#### Visão Geral

: A opção mais acessível, custando entre R\$ 1–10/mês, onde os recursos são compartilhados entre vários usuários.

-

#### Limitações

: Não é amigável para desenvolvedores devido a restrições de



personalização, possíveis problemas de desempenho devido ao compartilhamento de recursos, falta de acesso SSH remoto e capacidades de automação limitadas.

-

## **Uso**

: Adequado para necessidades pequenas e modestas, mas não recomendado para sites de negócios ou aplicações populares.

## **Servidor Privado Virtual (VPS)**

-

## **Visão Geral**

: Funciona como um servidor independente com recursos dedicados, como memória, CPU e largura de banda, custando entre R\$ 10–100/mês.

-

## **Vantagens**

: Fornece acesso root SSH. permite personalização de

**Instalar o aplicativo Bookey para desbloquear  
texto completo e áudio**

**Mais Livros Grátis**



Escaneie para baixar



**Ouça-o**



# Ler, Compartilhar, Empoderar

Conclua Seu Desafio de Leitura, Doe Livros para Crianças Africanas.

## O Conceito



Esta atividade de doação de livros está sendo realizada em conjunto com a Books For Africa. Lançamos este projeto porque compartilhamos a mesma crença que a BFA: Para muitas crianças na África, o presente de livros é verdadeiramente um presente de esperança.

## A Regra



Ganhe 100 pontos



Resgate um livro



Doe para a África

Seu aprendizado não traz apenas conhecimento, mas também permite que você ganhe pontos para causas beneficentes! Para cada 100 pontos ganhos, um livro será doado para a África.

Teste gratuito com Bookee





# Capítulo 10 Resumo : 7.

## Provisionamento

### Resumo do Capítulo 10: Provisionamento de um Servidor PHP

#### Introdução ao Provisionamento de Servidores

Provisionar um servidor para sua aplicação PHP envolve configurar e gerenciar recursos do servidor de acordo com as necessidades específicas da sua aplicação. Para alguns, usar uma Plataforma como Serviço (PaaS) simplifica esse processo, enquanto outros precisam configurar Servidores Privados Virtuais (VPS) ou servidores dedicados por conta própria.

#### Configuração Inicial do Servidor

1.

#### Escolhendo um Provedor de VPS

: Provedores recomendados incluem o Linode pela

**Mais Livros Grátis**



Escaneie para baixar



**Ouça-o**

confiabilidade, embora possa não ser a opção mais barata.

2.

### **Acesso ao Servidor**

: Acesse seu servidor através do SSH utilizando um comando como ``ssh root@<seu-ip-do-servidor>``.

3.

### **Atualizações de Software**

: Atualize o software do sistema operacional do seu servidor usando ``apt-get`` para Ubuntu ou ``yum`` para CentOS.

## **Práticas de Segurança**

1.

### **Criar um Usuário Não Root**

: Estabeleça um usuário não root (por exemplo, ``deploy``) para limitar o acesso e aumentar a segurança.

- Para Ubuntu: ``adduser deploy`` seguido da concessão de acesso ``sudo``.

- Para CentOS: ``adduser deploy``, defina uma senha e adicione ao grupo wheel para privilégios ``sudo``.

2.

### **Autenticação por Par de Chaves SSH**

: Recomenda-se fortalecer a segurança por meio de chaves



SSH em vez de senhas. Gere uma chave SSH e armazene a chave pública no servidor, garantindo que as permissões estejam configuradas corretamente.

3.

## **Desativar Autenticação por Senha**

: Fortalecer ainda mais seu servidor envolve desativar o login root e a autenticação por senha na configuração do SSH.

## **Configurando o PHP-FPM**

- O PHP-FPM (Gerenciador de Processos FastCGI) é necessário para gerenciar os processos PHP.
- Instale o PHP-FPM e configure-o, focando em limites de memória, pools de processos e otimizações de desempenho.

## **Configuração do Nginx**

1.

### **Instalação**

: Use o gerenciador de pacotes para instalar o Nginx.

- Ubuntu: ``sudo apt-get install nginx``
- CentOS: ``sudo yum install nginx``.

2.



## **Configurar Hosts Virtuais**

: Configure um host virtual para sua aplicação PHP criando um arquivo de configuração que detalha o nome do servidor, arquivos de log, raiz do documento e direcionando requisições PHP para o PHP-FPM.

## **Automatizando o Provisionamento do Servidor**

- Considere usar ferramentas de automação como Puppet, Chef, Ansible ou SaltStack para facilitar o provisionamento do servidor.

## **Serviços Online para Delegar o Provisionamento do Servidor**

- Serviços como o Forge podem automatizar as tarefas de configuração e gerenciamento do servidor, incluindo configurações de segurança e instalação de software para aplicações PHP.

## **Leitura e Recursos de Aprendizado Adicionais**

- O capítulo enfatiza recursos para aprimorar habilidades de administração de sistemas, recomendando "Servers for

**Mais Livros Grátis**



Escaneie para baixar



**Ouça-o**



Hackers" de Chris Fido para aqueles que apreciam administração de sistemas.

## Conclusão

Após seguir estas etapas, seu servidor deve estar bem configurado, seguro e pronto para executar aplicações PHP de forma eficaz. O próximo capítulo abordará estratégias para otimizar o desempenho.

**Mais Livros Grátis**



Escaneie para baixar



**Ouça-o**

# Capítulo 11 Resumo : 8. Ajustando

## Resumo do Capítulo 11: Ajustando Aplicações PHP

### Introdução

Para otimizar sua aplicação PHP, é fundamental ajustar sua configuração juntamente com o nginx e o PHP-FPM. Uma instalação padrão do PHP carece de eficiência, semelhante a um terno mal ajustado. Um ajuste adequado melhora o desempenho, mas não pode compensar práticas de codificação deficientes.

### Arquivo php.ini

O intérprete PHP é ajustado usando o arquivo de configuração `php.ini`, que reside em diretórios específicos dependendo do ambiente. As principais configurações incluem:

1.

### Gerenciamento de Memória

Mais Livros Grátis



Escaneie para baixar



Ouça-o

- A configuração ``memory_limit`` controla o uso de memória por processo PHP, com padrão de 128M.
- Ajuste conforme o tamanho da aplicação: menor para aplicativos pequenos (64M) e maior para aplicativos que consomem mais memória (512M).

2.

## **Zend OPcache**

- Este cache de opcode embutido melhora o desempenho armazenando código PHP compilado, reduzindo a sobrecarga.
- Configurações recomendadas incluem:
  - ``opcache.memory_consumption``: 64M
  - ``opcache.max_accelerated_files``: 4000
- Ajuste as strings internadas e os tempos de cache durante o desenvolvimento e a produção.

3.

## **Uploads de Arquivos**

- Desative uploads de arquivos se não forem necessários, ou limite o tamanho máximo de upload e o número de uploads simultâneos.

4.

## **Tempo Máximo de Execução**

**Mais Livros Grátis**



Escaneie para baixar



**Ouça-o**

- Defina ``max_execution_time`` mais baixo (por exemplo, para 5 segundos) para tempos de resposta mais rápidos e gerencie tarefas de longa duração por meio de processos em segundo plano ou filas.

5.

## **Gerenciamento de Sessões**

- Utilize armazenamento na memória, como Memcached ou Redis, para melhorar o gerenciamento de sessões e facilitar a escalabilidade.

6.

## **Bufferização de Saída**

- Ative a bufferização de saída para enviar dados de forma mais eficiente e reduzir requisições HTTP.

7.

## **Cache Realpath**

- Aumente o tamanho do cache realpath do PHP para aplicações maiores para melhorar o desempenho da gestão de arquivos.

## **Conclusão**

**Mais Livros Grátis**



Escaneie para baixar



**Ouça-o**

Com as configurações apropriadas definidas, a aplicação PHP está bem ajustada e pronta para implantação. O próximo capítulo abordará estratégias para automatizar efetivamente a implantação de aplicações PHP.

**Mais Livros Grátis**



Escaneie para baixar



Ouça-o

## Exemplo

**Ponto chave:** Otimize as configurações.

**Exemplo:** Ao ajustar o arquivo ``php.ini`` para a sua aplicação PHP, pense em como você ajustaria o motor de um carro para um desempenho máximo. Assim como você não dirigiria um carro de corrida com as configurações do motor inalteradas, otimizar configurações como ``memory_limit`` ou habilitar o Zend OPcache pode melhorar significativamente a velocidade e a eficiência da sua aplicação. Imagine seu site carregando instantaneamente em vez de apresentar lentidão, proporcionando uma experiência de usuário mais fluida que mantém os visitantes engajados e satisfeitos. Cada pequeno ajuste pode levar a melhorias dramáticas, garantindo que sua aplicação PHP funcione como uma máquina bem afinada.

Mais Livros Grátis



Escaneie para baixar



Ouça-o

## Pensamento crítico

**Ponto chave:** Ajustar a Configuração do PHP é Essencial para a Eficácia da Aplicação

**Interpretação crítica:** Josh Lockhart enfatiza que ajustar aplicações PHP por meio de modificações cuidadosas na configuração de arquivos como o php.ini melhora significativamente o desempenho. No entanto, é preciso avaliar criticamente a afirmação de que a otimização de desempenho pode levar à eficácia ótima da aplicação, pois pode deixar de fora questões mais profundas, como a qualidade do código e a arquitetura. Embora a configuração desempenhe um papel, é fundamental reconhecer que práticas de codificação inadequadas podem anular os benefícios da otimização, como apoiado por fontes como 'Code Complete' de Steve McConnell, que destaca a importância da qualidade do código fundamental.

Mais Livros Grátis



Escaneie para baixar



Ouça-o



# Capítulo 12 Resumo : 9. Implantação

## Resumo do Capítulo 12: Implantação

### Introdução

Este capítulo discute estratégias modernas de implantação para aplicações PHP, distantes dos métodos tradicionais como FTP, em direção a processos automáticos mais confiáveis e eficientes.

### Controle de Versão

-

#### Importância do Controle de Versão

: O uso de controle de versão, especialmente o Git, é fundamental para rastrear mudanças, marcar lançamentos e facilitar a capacidade de reversão.

-

#### Automatizando a Implantação

: A implantação automatizada deve ser simples, previsível e reversível para incentivar atualizações regulares em



produção.

## **Processo de Implantação**

-

### **Simplicidade**

: O processo de implantação deve consistir em um comando simples que seja fácil de executar.

-

### **Previsibilidade**

: É essencial que o processo de implantação se comporte de maneira consistente, sem resultados imprevisíveis.

-

### **Reversibilidade**

: Em caso de problemas com uma nova versão, reverter para a versão anterior deve ser possível com um comando simples.

**Instalar o aplicativo Bookey para desbloquear  
texto completo e áudio**

**Mais Livros Grátis**



Escaneie para baixar



**Ouça-o**



# As melhores ideias do mundo desbloqueiam seu potencial

Essai gratuit avec Bookey



Escanear para baixar



# Capítulo 13 Resumo : 10. Testes

## Capítulo 10 - Testes

### Visão Geral dos Testes em PHP

Os testes são um componente essencial, mas muitas vezes negligenciado, no desenvolvimento de aplicações em PHP. Muitos desenvolvedores os veem como algo que consome tempo e pode ser um fardo, possivelmente devido à falta de familiaridade com as ferramentas disponíveis. Este capítulo tem como objetivo mudar essa percepção, enfatizando que os testes devem ser integrados ao longo de todo o processo de desenvolvimento.

### Por Que Testamos?

Os testes garantem que as aplicações PHP se comportem como esperado e ajudam a aliviar o medo de colocar o código em produção, proporcionando confiança em sua funcionalidade. Testes bem implementados previnem bugs futuros e tempo de inatividade desnecessário, economizando

**Mais Livros Grátis**



Escaneie para baixar



**Ouça-o**



tempo e dinheiro a longo prazo.

## Quando Testamos?

-

### **Antes do Desenvolvimento:**

Configure as ferramentas de teste como dependências vitais para facilitar testes mais simples durante o processo de codificação.

-

### **Durante o Desenvolvimento:**

Escreva e execute testes enquanto constrói funcionalidades para detectar problemas precocemente.

-

### **Após o Desenvolvimento:**

Continue a modificar os testes conforme a aplicação evolui, garantindo que permaneçam relevantes e eficazes.

## O Que Testamos?

Os testes se concentram nos menores componentes de uma aplicação:

-

### **Testes de Unidade:**

Mais Livros Grátis



Escaneie para baixar



Ouça-o

Verificam a funcionalidade de classes, métodos e funções individuais em isolamento.

-

### **Testes Funcionais:**

Avaliam a aplicação como um todo para garantir que funcione em conjunto como esperado.

### **Como Testamos?**

Existem várias metodologias para testes em PHP:

-

### **Testes de Unidade:**

Usando o PHPUnit, um popular framework de testes de unidade que adere à arquitetura xUnit. Conhecer o PHPUnit é crucial para quem trabalha com componentes PHP.

-

### **Desenvolvimento Orientado a Testes (TDD):**

Escrever testes antes do código real da aplicação, seguido pelo desenvolvimento iterativo de funcionalidades até que os testes passem.

-

### **Desenvolvimento Orientado a Comportamento (BDD):**

Escrever testes em um formato de linguagem natural, sendo

**Mais Livros Grátis**



Escaneie para baixar



**Ouça-o**

o SpecBDD focado em detalhes de implementação e o StoryBDD na lógica de negócios.

## Configurando o PHPUnit

-

### Estrutura de Diretórios:

Organize os projetos com um diretório ``src/`` para o código-fonte e um diretório ``tests/`` para os testes.

- Para instalar o PHPUnit, use o Composer com: ``composer require --dev phpunit/phpunit``.

- Para análise de cobertura de código, o profiler do Xdebug pode ser instalado para gerar relatórios sobre quais partes do código estão sendo testadas.

## Escrevendo Testes com PHPUnit

1. Crie um caso de teste estendendo

``PHPUnit_Framework_TestCase``.

2. Defina métodos de teste com nomes que começam com ``test``.

3. Utilize asserções como ``assertEquals()`` e

``assertAttributeEquals()`` para validar os resultados esperados em relação aos resultados reais.





## Executando Testes

Execute a suíte de testes usando o comando:

```
```bash
vendor/bin/phpunit -c phpunit.xml
```
```

Os resultados indicarão asserções e testes que passaram ou falharam, junto com métricas de desempenho.

## Cobertura de Código

Gere um relatório de cobertura de código para avaliar quanto do código está sendo testado:

```
```bash
vendor/bin/phpunit -c phpunit.xml --coverage-html coverage
```
```

Analise o relatório para áreas que precisam de mais testes, mas reconheça que 100% de cobertura não é realista ou necessário; fornece diretrizes para melhorias.

## Testes Contínuos com Travis CI

Automatize os testes com serviços como o Travis CI, que

**Mais Livros Grátis**



Escaneie para baixar



**Ouça-o**

podem executar testes a cada nova alteração de código, garantindo que a aplicação permaneça estável em várias versões do PHP. Um exemplo de configuração usando YAML especifica as versões do PHP e os comandos para instalação de dependências e execução de testes.

## Conclusão

Este capítulo fornece uma base para integrar testes nas práticas de desenvolvimento em PHP. Ao entender o porquê, o quando e o como dos testes — juntamente com passos práticos para implementá-los — os desenvolvedores podem aumentar significativamente a confiabilidade e confiança em suas aplicações.

**Mais Livros Grátis**



Escaneie para baixar



Ouça-o

## Pensamento crítico

**Ponto chave:** Importância dos Testes no Desenvolvimento

**Interpretação crítica:** O capítulo enfatiza que os testes não são apenas uma etapa adicional, mas uma parte crítica para garantir a qualidade e a confiança em aplicações PHP. No entanto, é importante considerar que a defesa do capítulo por priorizar os testes pode negligenciar cenários únicos onde o desenvolvimento rápido ou a falta de recursos podem dificultar a implementação de testes aprofundados. Experiências variadas no desenvolvimento (por exemplo, metodologias ágeis ou projetos com prazos apertados) podem desafiar a noção de que os testes devem sempre ser integrais, sugerindo que uma abordagem mais sutil é necessária. Fontes como "The Pragmatic Programmer" de Andrew Hunt e David Thomas discutem como a flexibilidade nas práticas pode, às vezes, gerar melhores resultados em determinadas circunstâncias.

Mais Livros Grátis



Escaneie para baixar



Ouça-o

# Capítulo 14 Resumo : 11. Profiling

## Capítulo 14: Profiling

Profiling é uma técnica essencial para analisar o desempenho de aplicações e identificar gargalos dentro do código PHP. É particularmente útil quando uma aplicação apresenta comportamento lento, pois permite que os desenvolvedores rastreiem a pilha de chamadas PHP, monitorem quais funções ou métodos são invocados e avaliem sua duração de execução, uso de memória e CPU ao longo do ciclo de vida da aplicação.

### Quando Usar um Profiler

O profiling deve ser realizado apenas quando os problemas de desempenho são difíceis de diagnosticar. Problemas de desempenho comuns podem ser detectados usando ferramentas de benchmarking como Apache Bench ou Siege, que simulam a interação do usuário com a aplicação. Se os problemas de desempenho ainda não forem evidentes, deve-se utilizar um profiler para investigar mais a fundo.

**Mais Livros Grátis**



Escaneie para baixar



**Ouça-o**

## Tipos de Profilers

Existem duas categorias de profilers: aqueles adequados para desenvolvimento e aqueles que podem ser usados em produção. O Xdebug é um profiler de desenvolvimento bem conhecido que exige recursos significativos para operar e gera uma saída que não é legível por humanos, a qual pode ser visualizada usando ferramentas como KCacheGrind. Em contraste, o XHProf, desenvolvido pelo Facebook, pode funcionar tanto em ambientes de desenvolvimento quanto em produção e gera dados que podem ser visualizados através do XHGUI.

## Usando o Xdebug

Para usar o Xdebug de forma eficaz, é necessário configurá-lo no arquivo php.ini. As principais configurações incluem habilitar o profiling sob demanda, especificar o diretório de saída para os resultados e controlar quando o profiling ocorre usando parâmetros de consulta.

## Usando o XHProf

O HProf também pode operar de maneira semelhante ao

**Mais Livros Grátis**



Escaneie para baixar



**Ouça-o**

Xdebug, mas utiliza uma carga mais leve nos recursos do sistema e é mais adequado para ambientes com carga de usuários mais intensa. Para instalar e configurar corretamente o XHProf e sua ferramenta de visualização, XHGUI, algumas dependências específicas devem ser atendidas, incluindo suporte ao Composer e ao PHP MongoDB.

## Outros Profilers

-

### **New Relic Profiler:**

Um serviço pago que fornece monitoramento de desempenho em tempo real e análises.

-

### **Blackfire Profiler:**

Uma ferramenta mais recente que oferece capacidades de visualização únicas para descobrir gargalos em aplicações.

## Leitura Adicional

Para aprofundar sua compreensão sobre profiling em PHP, vários recursos online estão disponíveis, abordando ferramentas como XHProf e XHGUI, além de práticas recomendadas em análise de desempenho.

**Mais Livros Grátis**



Escaneie para baixar



**Ouça-o**

## O Que Vem a Seguir?

O capítulo também alude a desenvolvimentos futuros no ecossistema PHP, incluindo desempenho aprimorado com PHP 7, HHVM e outras iniciativas que visam evoluir as capacidades e o desempenho do PHP como uma linguagem.

**Mais Livros Grátis**



Escaneie para baixar



Ouça-o



# Capítulo 15 Resumo : 12. HHVM e Hack

## Capítulo 15: HHVM e Hack

### Visão Geral do HHVM e Hack

O Facebook causou um impacto significativo na comunidade PHP com duas iniciativas: HHVM (Hip Hop Virtual Machine) e Hack, que é uma extensão da linguagem PHP. O HHVM, lançado em 2013, melhora o desempenho por meio de um compilador just-in-time, acelerando significativamente sites como Wordpress e MediaWiki. O Hack introduz tipagem estática e novas estruturas de dados, mantendo a compatibilidade com versões anteriores do PHP.

### A Necessidade do HHVM

Desde 1994, o interpretador principal do PHP era o Zend Engine, mas à medida que o Facebook crescia, isso se tornou um gargalo para o desempenho. Em resposta, o Facebook desenvolveu o HPHPC, um compilador de PHP para C++ que aumentou o desempenho, mas tinha limitações. O HHVM foi

**Mais Livros Grátis**



Escaneie para baixar



**Ouça-o**

desenvolvido como uma solução híbrida que otimiza a execução sem os longos tempos de compilação, permitindo que os scripts PHP sejam executados "just in time."

## **Aprimoramentos de Desempenho**

O HHVM converte scripts PHP em bytecode e utiliza um compilador JIT, superando o HPHPC a partir do final de 2012. O HHVM visa compatibilidade quase total com frameworks PHP, e embora ainda existam casos particulares a resolver, muitas organizações usam o HHVM em produção.

## **O HHVM É o Certo Para Você?**

O HHVM não é uma solução universal e é melhor para desenvolvedores que já otimizaram ainda mais suas aplicações. Antes de considerar o HHVM, as melhorias de desempenho da aplicação devem focar na otimização de

**Instalar o aplicativo Bookey para desbloquear  
texto completo e áudio**

**Mais Livros Grátis**



Escaneie para baixar



**Ouça-o**

Ad



Escanear para baixar



# Experimente o aplicativo Bookey para ler mais de 1000 resumos dos melhores livros do mundo

Desbloqueie **1000+** títulos, **80+** tópicos

Novos títulos adicionados toda semana

Product & Brand

 Liderança & Colaboração

 Gerenciamento de Tempo

 Relacionamento & Comunicação

 Estratégia de Negócios

 Criatividade

 Memórias

 Conheça a Si Mesmo

 Psicologia

Empreendedorismo

 História Mundial

 Comunicação entre Pais e Filhos

 Autocuidado

 Mente

## Visões dos melhores livros do mundo

amento  
pos

Os 7 Hábitos das  
Pessoas Altamente  
Eficazes



Mini Hábitos



Hábitos Atômicos



O Clube das 5  
da Manhã



Como Fazer Amigos  
e Influenciar  
Pessoas



Com  
Não



Teste gratuito com Bookey



# Capítulo 16 Resumo : 13. Comunidade

## Capítulo 16: Comunidade

### Introdução

A comunidade PHP é um recurso vital para os desenvolvedores, oferecendo um ambiente diversificado e acolhedor para aprender e compartilhar experiências. Interagir com outros desenvolvedores pode aprimorar suas habilidades e ampliar sua rede de contatos.

### PUG Local

-

### Participe de Grupos de Usuários de PHP (PUG)

#### Locais

: Encontre e junte-se ao seu Grupo de Usuários de PHP local em [php.ug](http://php.ug). É uma excelente forma de se conectar com desenvolvedores de PHP da sua região.

-

### Inicie um PUG

Mais Livros Grátis



Escaneie para baixar



Ouça-o



: Se não houver nenhum nas proximidades, considere iniciar um ou participe do NomadPHP, um grupo online com palestrantes mensais e discussões sobre diversos tópicos de PHP.

## Conferências

-

### Participe de Conferências PHP

: Diversas conferências de PHP ocorrem anualmente, facilitando o networking com líderes da indústria e atualizações sobre novas funcionalidades. Uma lista de conferências pode ser encontrada em [\[php.net/conferences\]\(http://php.net/conferences/\)](http://php.net/conferences/).

## Mentoria

-

### Busque ou Torne-se um Mentor

: Iniciantes podem encontrar mentores em [\[phpmentoring.org\]\(http://phpmentoring.org\)](http://phpmentoring.org). Desenvolvedores experientes são incentivados a mentorear novatos, elevando o nível de habilidade geral na comunidade.

Mais Livros Grátis



Escaneie para baixar



Ouça-o

## Mantenha-se Atualizado

-

### Recursos

: Fique informado sobre os desenvolvimentos em PHP através de diversos recursos:

-

### Sites

: [php.net](http://php.net), [php-fig.org](http://php-fig.org), [phptherightway.com](http://phptherightway.com)

-

### Listas de Discussão

: [php.net/mailling-lists.php](http://php.net/mailling-lists.php)

-

### Contas no Twitter

: @official\_php, @phpc

-

### Podcasts

: Voices of the ElePHPant, [looselycoupled.info](http://looselycoupled.info), outros mencionados acima.

-

### Contas de Humor

: @phpbard e @phpdrama.

## Conclusão

Mais Livros Grátis



Escaneie para baixar



Ouça-o



Participe ativamente da comunidade PHP através de pontos locais, conferências e programas de mentoria. Mantenha-se atualizado com recursos para garantir crescimento e conexão contínuos em sua jornada com PHP.

**Mais Livros Grátis**



Escaneie para baixar



Ouçá-o

# Capítulo 17 Resumo : A. Instalando PHP

## Resumo do Capítulo 17: Instalando PHP

### Introdução

Este capítulo foca na instalação do PHP em diferentes sistemas operacionais, especialmente Linux, OS X e Windows. Ele descreve os passos para utilizar gerenciadores de pacotes, usar repositórios de terceiros e instalações personalizadas.

### Instalação do PHP no Linux

-

#### Visão Geral

: Usando gerenciadores de pacotes como ``aptitude`` (Ubuntu) e ``yum`` (CentOS/RHEL) para instalar o PHP, especificamente para uso em linha de comando.

-

#### Gerenciadores de Pacotes

: Cada distribuição Linux possui seu próprio gerenciador de



pacotes. Para Ubuntu, use ``aptitude``; para CentOS, use ``yum``.

-

## **Consideração de Versão**

: Os repositórios padrões podem oferecer versões desatualizadas do PHP, portanto, repositórios de terceiros devem ser adicionados.

**\*Passos de Instalação no Ubuntu\*:**

1.

### **Adicionar Dependências de Software**

: Instale ``python-software-properties`` para permitir o uso do PPA.

2.

### **Adicionar PPA**

: Use ``sudo add-apt-repository ppa:ondrej/php5-5.6`` para expandir a seleção de software.

3.

### **Atualizar e Instalar PHP**

: Atualize as listas de pacotes com ``sudo apt-get update`` e instale o pacote PHP CLI junto com as extensões necessárias.

**\*Passos de Instalação no CentOS\*:**

1.

### **Adicionar Repositório EPEL**

**Mais Livros Grátis**



Escaneie para baixar



**Ouçá-o**

: Use comandos RPM para instalar os repositórios EPEL e remi.

2.

## **Instalar PHP**

: Similar ao Ubuntu, utilize `yum` para instalar o PHP com as extensões relevantes.

## **Instalação do PHP no OS X**

-

### **Pré-Instalação do PHP**

: O OS X inclui PHP, mas provavelmente não a versão mais recente. Métodos recomendados incluem MAMP e Homebrew.

**\*Instalação do MAMP\*:**

- Fornece uma interface gráfica fácil de usar. Os usuários podem baixar um instalador de pacote, iniciar os servidores a partir do aplicativo e acessar arquivos no diretório `htdocs`.

**\*Instalação do Homebrew\*:**

1.

## **Instalar Ferramentas de Linha de Comando do XCode**

: Necessário para compilar pacotes.

2.

**Mais Livros Grátis**



Escaneie para baixar



**Ouça-o**

## **Instalar Homebrew**

: Use um comando Ruby no terminal.

3.

## **Permissões de Diretório**

: Garanta que sua conta de usuário possua o diretório  
`/usr/local`.

4.

## **Adicionar PATH ao Ambiente**

: Configure o PATH do sistema para incluir os binários do Homebrew.

5.

## **Adicionar Repositórios de Fórmulas**

: Adicione os repositórios necessários para instalação do PHP via Homebrew.

6.

## **Instalar PHP**

: Pesquise e instale a versão do PHP desejada usando `brew install`.

## **Compilando PHP a Partir do Código Fonte**

-

## **Vantagens**

: Oferece mais controle sobre a instalação e suporta

**Mais Livros Grátis**



Escaneie para baixar



**Ouçá-o**

configurações específicas.

-

## **Processo**

:

1.

### **Baixar Código Fonte**

: Obtenha a versão mais recente do site oficial do PHP.

2.

### **Instalar Dependências**

: Utilize gerenciadores de pacotes para instalar as bibliotecas necessárias.

3.

### **Configurar e Compilar o PHP**

: Use `./configure` com opções específicas, incluindo habilitar FPM, suporte a multibyte, curl, etc.

4.

### **Instalar**

: Finalize executando `make && make install`.

5.

### **Configurar `php.ini`**

: Baixe um arquivo `php.ini` pré-configurado para uso em desenvolvimento local.

## **Instalação do PHP no Windows**

**Mais Livros Grátis**



Escaneie para baixar



**Ouça-o**



-

## **Binaries**

: Baixe os binaries do PHP em PHP.net e configure o `php.ini` para uso na linha de comando.

-

## **WAMP**

: Similar ao MAMP, fornece um ambiente de instalação tudo-em-um para desenvolvimento em PHP no Windows.

-

## **Zend Server**

: Oferece um pacote completo com Apache, PHP, MySQL e ferramentas de depuração.

## **Conclusão**

O capítulo fornece métodos abrangentes de instalação do PHP nos principais sistemas operacionais, garantindo que os usuários possam desenvolver em seu ambiente preferido enquanto entendem as ferramentas disponíveis para gerenciar o PHP de forma eficaz.

**Mais Livros Grátis**



Escaneie para baixar



**Ouça-o**

# Capítulo 18 Resumo : B. Ambientes de Desenvolvimento Local

## Apêndice B: Ambientes de Desenvolvimento Local

Neste capítulo, discutimos a importância de desenvolver aplicações em PHP Moderno em um computador local e as ferramentas para atingir isso de forma eficaz. Usar a pilha de software padrão de um sistema operacional é desaconselhado devido a problemas como software desatualizado ou possíveis sobrescritas durante as atualizações do SO. Em vez disso, recomenda-se configurar um ambiente de desenvolvimento local dentro de uma máquina virtual (VM), garantindo que ele reflita o ambiente de produção para consistência e confiabilidade.

### VirtualBox

O VirtualBox é uma ferramenta de código aberto que possibilita a criação e gerenciamento de máquinas virtuais. Ele fornece um processo de instalação simples e é compatível com sistemas OS X e Windows. É eficiente, embora sua

**Mais Livros Grátis**



Escaneie para baixar



**Ouçá-o**

interface de usuário possa não ser tão polida quanto as opções comerciais.

## Vagrant

Para gerenciar VMs do VirtualBox com facilidade, introduzimos o Vagrant. Esta ferramenta simplifica o processo de criação, início, parada e destruição de VMs através de uma interface de linha de comando. Após a instalação do Vagrant, os usuários podem utilizar diversos comandos essenciais, como ``vagrant init``, ``vagrant up`` e ``vagrant halt``, para gerenciar efetivamente seu ambiente de desenvolvimento.

## Vagrant Boxes

As Vagrant boxes servem como ambientes pré-configurados para construir aplicações. Os usuários podem explorar as

**Instalar o aplicativo Bookey para desbloquear  
texto completo e áudio**

Mais Livros Grátis



Escaneie para baixar



Ouça-o



Escanear para baixar



# Por que o Bookey é um aplicativo indispensável para amantes de livros



## Conteúdo de 30min

Quanto mais profunda e clara for a interpretação que fornecemos, melhor será sua compreensão de cada título.



## Clipes de Ideias de 3min

Impulsione seu progresso.



## Questionário

Verifique se você dominou o que acabou de aprender.



## E mais

Várias fontes, Caminhos em andamento, Coleções...

Teste gratuito com Bookey



# Melhores frases do PHP Moderno por Josh Lockhart com números de página

Ver no site do Bookey e gerar imagens de citações bonitas

## Capítulo 1 | Frases das páginas 75-88

1. A linguagem PHP está vivendo um renascimento.
2. Desenvolvedores PHP dependem menos de frameworks monolíticos e mais de componentes especializados menores.
3. O gerenciador de dependências Composer está revolucionando a forma como construímos aplicações PHP; ele nos emancipa do jardim murado de um framework e nos permite misturar e combinar componentes PHP interoperáveis mais adequados para nossas aplicações PHP personalizadas.
4. O PHP Moderno abrange muitas novas práticas que podem ser desconhecidas para aqueles que são novos no PHP, ou para aqueles que estão atualizando de versões mais antigas do PHP.
5. É um momento empolgante para ser um programador PHP.

Mais Livros Grátis



Escaneie para baixar



Ouça-o



## Capítulo 2 | Frases das páginas 89-214

1. A linguagem PHP Moderna tem muitos recursos novos e empolgantes.
2. Namespaces são uma ferramenta importante que organiza o código PHP em uma hierarquia virtual.
3. Namespaces são importantes porque nos permitem criar código isolado que funciona junto ao código de outros desenvolvedores.
4. Programar para uma interface mudou a minha vida como programador PHP...
5. Traits permitem implementações modulares que podem ser injetadas em classes não relacionadas.
6. Geradores são iteradores simples... que economizam memória ao gerar valores sob demanda.
7. Closures e funções anônimas são... ferramentas extremamente úteis que todo desenvolvedor PHP deve ter no seu kit.
8. Zend OPcache... é importante porque armazena o bytecode PHP pré-compilado.





9.O PHP tem um servidor web embutido... para desenvolvimento local.

10.A linguagem PHP Moderna tem muitos recursos poderosos que podem melhorar suas aplicações.

## **Capítulo 3 | Frases das páginas 215-1073**

1.Se há um recurso do PHP Moderno que eu quero que você conheça, são os namespaces.

2.Os namespaces são importantes porque nos permitem criar código isolado que funciona ao lado do código de outros desenvolvedores.

3.Sem namespaces, uma colisão de nomes faz com que o PHP falhe.

4.Com namespaces, seu código e o código de outros desenvolvedores podem usar o mesmo nome de classe, interface, função ou constante, assumindo que seu código está abaixo de um namespace de fornecedor único.

5.Aprender a programar para uma interface mudou minha vida como programador PHP...

6.As interfaces desacoplam nosso código de suas

**Mais Livros Grátis**



Escaneie para baixar



**Ouça-o**

dependências, permitindo que nosso código dependa de qualquer código de terceiros que implemente a interface esperada.

7. Um trait é uma implementação parcial de classe que pode ser misturada em uma ou mais classes PHP existentes.

8. A linguagem PHP usa um modelo de herança clássico... No entanto, o que fazemos se duas classes PHP não relacionadas precisarem exibir comportamentos semelhantes?

9. Geradores computam e fornecem valores de iteração sob demanda.

10. Uma closure é uma função que encapsula seu estado circundante no momento em que é criada.

**Mais Livros Grátis**



Escaneie para baixar



Ouça-o



Baixe o app Bookey para desfrutar

# Mais de 1000 resumos de livros com quizzes

**Teste grátis disponível!**

Escanear para baixar



## Capítulo 4 | Frases das páginas 1074-1122

- 1....o ecossistema PHP Moderno é um verdadeiro caldeirão de código que ajuda nós, desenvolvedores, a construir aplicações incríveis.
- 2.Este ambiente centralizado é adequado se você está satisfeito com as ferramentas do framework. No entanto, e se você usar o framework CodeIgniter, mas quiser escolher uma biblioteca auxiliar do framework Symfony?
- 3.Frameworks criados isoladamente não foram projetados para se comunicar com outros frameworks. Isso é extremamente ineficiente, tanto para os desenvolvedores (a criatividade é limitada pela escolha do framework) quanto para os próprios frameworks (eles reinventam código que já existe em outros lugares).
- 4.A comunidade PHP evoluiu de um modelo de framework centralizado para um ecossistema distribuído de componentes eficientes, interoperáveis e especializados.
- 5.O PHP-FIG cria recomendações que os frameworks PHP podem implementar voluntariamente para melhorar a



comunicação e o compartilhamento com outros frameworks.

6.PSR é um acrônimo para recomendações de padrões PHP.

Se você leu recentemente um blog relacionado a PHP, provavelmente viu os termos PSR-1, PSR-2, PSR-3, e assim por diante.

7.PSR-1 fornece diretrizes simples que são fáceis de implementar com um esforço mínimo.

8.Um estilo de código rigoroso comum permite que os desenvolvedores escrevam código que é facilmente e rapidamente compreendido por outros colaboradores.

## **Capítulo 5 | Frases das páginas 1123-1520**

1.Frameworks criados de forma isolada não foram projetados para se comunicarem com outros frameworks. Isso é extremamente ineficiente, tanto para os desenvolvedores (a criatividade é limitada pela escolha do framework) quanto para os próprios frameworks (eles reinventam código que já existe em outro lugar).

**Mais Livros Grátis**



Escaneie para baixar



**Ouça-o**

- 2.A comunidade PHP evoluiu de um modelo de framework centralizado para um ecossistema distribuído de componentes eficientes, interoperáveis e especializados.
- 3.Eu recomendo fortemente que você se envolva com o PHP-FIG, se apenas para enviar feedback e ajudar a moldar o futuro dos seus frameworks PHP favoritos.
- 4.Um componente PHP é focado e existe apenas para resolver um único problema muito bem. Não é um faz-tudo e mestre de nada; é um mestre em um.
- 5.Use a ferramenta certa para o trabalho. A maioria dos frameworks PHP Moderno são apenas um conjunto de convenções construídas sobre componentes PHP menores.
- 6.Packagist é onde você encontra componentes PHP.  
Composer é como você instala componentes PHP.

## **Capítulo 6 | Frases das páginas 1521-2446**

- 1.Não estamos mais atados a frameworks monolíticos e seus jardins murados. Hoje, escolhemos entre uma vasta e crescente coleção de componentes especializados para criar aplicações





personalizadas.

- 2.É tolice não aproveitar esses componentes para construir melhores aplicações mais rapidamente, em vez de perder tempo reinventando a roda.
- 3.A maioria dos frameworks PHP Modernos é apenas um conjunto de convenções construídas sobre componentes PHP menores.
- 4.Use a ferramenta certa para o trabalho. Se você está trabalhando em um projeto menor que pode ser resolvido com uma coleção precisa de componentes PHP, então use componentes.
- 5.Um componente PHP é um conjunto de código que ajuda a resolver um problema específico na sua aplicação PHP.
- 6.Uma boa documentação torna isso possível. O componente PHP deve ter um arquivo README que diga o que o componente faz, como instalá-lo e como usá-lo.

**Mais Livros Grátis**



Escaneie para baixar



Ouçá-o



Baixe o app Bookey para desfrutar

# Mais de 1000 resumos de livros com quizzes

**Teste grátis disponível!**

Escanear para baixar



## Capítulo 7 | Frases das páginas 2447-5377

1. Fox Mulder está correto—não confie em ninguém.
2. O conselho mais simples que posso te dar é este: saneie a entrada, valide os dados e escape a saída.
3. Nunca saiba as senhas dos seus usuários.
4. Você nunca deve ser capaz de saber as senhas dos seus usuários.
5. Sempre ative o relatório de erros.
6. Nunca restrinja as senhas dos seus usuários.
7. Se você exigir que senhas se encaixem em um padrão específico, você está efetivamente fornecendo um mapa para os criminosos hackearem sua aplicação.
8. Sempre conheça a codificação de caracteres dos seus dados. Armazene os dados com a codificação de caracteres UTF-8. Exiba os dados com a codificação de caracteres UTF-8.
9. Uma transação é tudo ou nada.

## Capítulo 8 | Frases das páginas 5378-5390

1. Você precisa hospedar sua aplicação em um



servidor e torná-la acessível ao seu público-alvo.

2. Escolha apenas o que você precisa, quando você precisa.

3. Grande poder, no entanto, vem com grande responsabilidade.

4. Se sua aplicação se tornar superpopular (centenas de milhares de visitantes por mês) e um VPS se tornar muito caro, você pode considerar atualizar para um servidor dedicado.

5. Recomendo planos de hospedagem PaaS para desenvolvedores que não desejam gerenciar seus próprios servidores.

## **Capítulo 9 | Frases das páginas 5391-5477**

1. Escolha apenas o que você precisa quando precisar. Você pode sempre escalar sua infraestrutura de hospedagem para cima ou para baixo quando necessário.

2. Um grande poder, no entanto, vem com uma grande responsabilidade.

3. Se sua aplicação se tornar superpopular e seu VPS estiver

**Mais Livros Grátis**



Escaneie para baixar



**Ouça-o**

sofrendo sob o peso de milhões de visitantes (parabéns, aliás!), adquira um servidor dedicado.

4. Eu quase sempre prefiro VPSs por seu equilíbrio entre custo, recursos e flexibilidade.

5. Plataformas como serviço (PaaS) são uma maneira rápida de lançar sua aplicação PHP, e—diferente de um servidor privado virtual ou dedicado—você não precisa gerenciar um PaaS.

6. O provisionamento de servidores é um processo longo. Também não é um processo divertido, especialmente se você provisionar muitos servidores manualmente.

7. Se você é responsável por vários servidores, eu encorajo fortemente você a explorar ferramentas de provisionamento, pois elas economizam muito tempo.

**Mais Livros Grátis**



Escaneie para baixar



Ouçá-o





Baixe o app Bookey para desfrutar

# Mais de 1000 resumos de livros com quizzes

**Teste grátis disponível!**

Escanear para baixar





## Capítulo 10 | Frases das páginas 5478-5916

1. Provisionar um servidor é uma arte, não uma ciência. Como você provisiona seu servidor depende inteiramente das necessidades da sua aplicação.
2. No entanto, a administração básica de sistemas é uma habilidade incrivelmente valiosa para desenvolvedores de aplicações, que permite um desenvolvimento de aplicações mais flexível e robusto.
3. A extensão Zend OPcache está integrada ao PHP 5.5.0+.
4. Código ruim ainda é código ruim. Por exemplo, a afinação do PHP não pode resolver consultas SQL mal escritas ou chamadas de API não responsivas.
5. A segurança do servidor é uma tarefa contínua que deve ser constantemente monitorada.
6. Muitos provedores de VPS, como Linode e Digital Ocean, cobram por hora, o que significa que você pode iniciar e brincar com um VPS a praticamente zero custo.
7. Não defina isso para algo muito grande, caso contrário, seu



servidor web (por exemplo, nginx) pode reclamar que a requisição HTTP tem um corpo muito grande ou que está expirando.

8. Costumo achar que os processos PHP consomem entre 5 a 20 MB de memória (seu resultado pode variar).

9. Se você aceitar uploads de arquivos muito grandes, certifique-se de que seu servidor web está configurado adequadamente.

10. Considero a administração de sistemas fascinante. Não quero fazer isso como um trabalho em tempo integral, mas gosto de mexer na linha de comando.

## **Capítulo 11 | Frases das páginas 5917-6043**

1. As instalações padrão do PHP são como um terno médio que você encontra em uma loja de departamentos; servem, mas não servem bem.

2. No entanto, a otimização do PHP é um fruto ao alcance que pode melhorar a eficiência do PHP e o desempenho da aplicação.

3. Não fique muito animado. A otimização do PHP não é uma



cura universal para o desempenho da aplicação. Código ruim ainda é código ruim.

4. Se sua aplicação PHP não tiver recursos suficientes, é prudente descobrir isso antes de levar a aplicação à produção.
5. Um processo de implantação simples é menos assustador, e isso significa que você está mais propenso a enviar código para produção.
6. O Capistrano é uma solução de implantação elegante e simples que torna as implantações de aplicações PHP simples, previsíveis e reversíveis.
7. Torne seu processo de implantação previsível. Um processo previsível é ainda menos assustador porque você sabe exatamente o que ele vai fazer.

## **Capítulo 12 | Frases das páginas 6044-6276**

1. Prefiro versionar meu código com Git, mas outros softwares de controle de versão, como Mercurial, também funcionam. Uso Git porque é o que conheço e ele funciona perfeitamente com



repositórios online populares, como Bitbucket e GitHub.

2. Faça do seu processo de implantação um comando simples de uma linha. Um processo de implantação simples é menos assustador, e isso significa que você tem mais chances de enviar código para a produção.
3. Torne seu processo de implantação previsível. Um processo previsível é ainda menos assustador porque você sabe exatamente o que ele vai fazer. Não deve ter efeitos colaterais inesperados.
4. Se você acidentalmente enviar um código ruim para a produção, deve ser um simples comando de uma linha para reverter para a base de código estável anterior. Esta é sua rede de segurança.
5. Os testes nos ajudam a escrever código que funciona bem na primeira vez. Os testes nos permitem iterar continuamente sem quebrar o código antigo.
6. Instale suas ferramentas de teste antes de desenvolver sua aplicação. Não importa quais ferramentas de teste você



escolha. Instale-as como se fossem uma dependência vital da aplicação.

7.Os testes devem ser uma preocupação em primeiro plano antes do desenvolvimento, durante o desenvolvimento e após o desenvolvimento.

8.Use o relatório de cobertura de código do PHPUnit como um guia para melhorar seu código. Não use porcentagens de cobertura de código como requisitos.

**Mais Livros Grátis**



Escaneie para baixar



Ouça-o

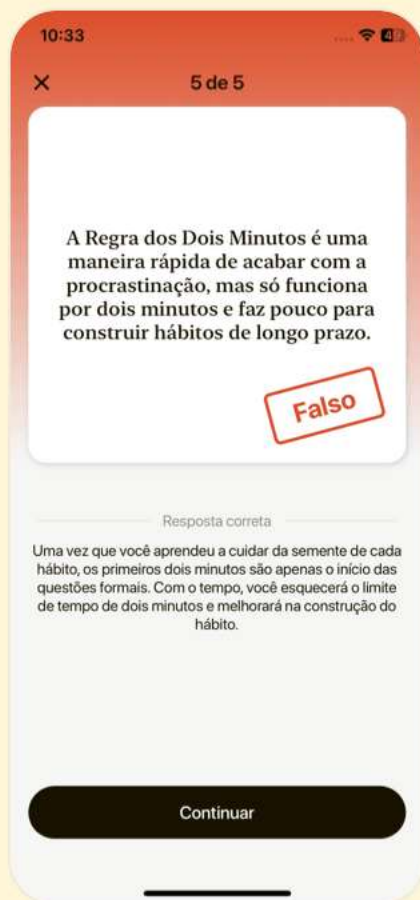


Baixe o app Bookey para desfrutar

# Mais de 1000 resumos de livros com quizzes

**Teste grátis disponível!**

Escanear para baixar





## Capítulo 13 | Frases das páginas 6277-6777

1. Os testes, no entanto, atenuam a incerteza e nos permitem escrever e implantar código com confiança.
2. Instalar uma infraestrutura de testes e escrever testes leva tempo, mas esse é um investimento sábio que traz retornos no futuro.
3. Os testes devem ser uma preocupação primordial antes do desenvolvimento, durante o desenvolvimento e após o desenvolvimento.
4. Se soubermos que cada parte funciona bem por conta própria, podemos ter confiança de que também funcionará bem quando integrada ao todo da aplicação.
5. Testar nossas aplicações constrói confiança e cria código mais previsível.

## Capítulo 14 | Frases das páginas 6778-6959

1. O profiling é como analisamos o desempenho da aplicação. É uma ótima maneira de depurar problemas de desempenho e identificar gargalos



no código da sua aplicação.

2. Você não precisa fazer profiling em suas aplicações PHP imediatamente. Você só deve fazer profiling em aplicações PHP se houver um problema de desempenho que seja difícil de diagnosticar.
3. Xdebug é um dos profilers PHP mais populares, e facilita a análise da pilha de chamadas da sua aplicação para encontrar gargalos e problemas de desempenho.
4. XHPProf é um novo profiler de aplicação PHP. Foi criado pelo Facebook e é destinado a ser executado tanto em desenvolvimento quanto em produção.
5. O futuro do PHP está se desenrolando à medida que falamos, graças a projetos visionários como PHP 7, HHVM, Hack e o PHP-FIG.
6. Acredito que muitos dos melhores recursos do Hack eventualmente encontrarão seu caminho no PHP. E vice-versa.

## **Capítulo 15 | Frases das páginas 6960-7499**

1. Pensando na aplicação do Facebook, não tenho

**Mais Livros Grátis**



Escaneie para baixar



**Ouça-o**

nada além de elogios para as pessoas brilhantes que trabalham no Facebook.

- 2.Os próprios desenvolvedores do Hack preferem chamá-lo de um dialeto do PHP e não uma nova linguagem.
- 3.Seu compilador just-in-time (JIT) oferece desempenho muitas vezes melhor que o PHP-FPM.
- 4.Se o Facebook consegue otimizar até mesmo a menor parte de seu processo de desenvolvimento, colhe uma grande recompensa tanto em eficiência dos desenvolvedores quanto em um código-base mais estável e com melhor desempenho.
- 5.É melhor usar uma Collection em vez de um array do PHP.
- 6.Acredito que a competição aprimorará ambas as linguagens e elas desfrutarão de uma relação simbiótica.

**Mais Livros Grátis**



Escaneie para baixar



**Ouça-o**



Baixe o app Bookey para desfrutar

# Mais de 1000 resumos de livros com quizzes

**Teste grátis disponível!**

Escanear para baixar



## Capítulo 16 | Frases das páginas 7500-7583

1. A comunidade PHP é seu recurso mais valioso. Ela é diversa, vibrante e global.
2. Eu o encorajo a participar da comunidade PHP para aprender e compartilhar com outros desenvolvedores PHP.
3. Se não houver um PUG próximo, você tem várias opções. Você pode iniciar seu próprio PUG.
4. Conferências são uma excelente oportunidade para encontrar e interagir com as melhores mentes da comunidade PHP.
5. Muitos desenvolvedores PHP especialistas doam seu tempo para ajudar novos desenvolvedores PHP a se tornarem melhores.
6. A linguagem PHP muda com frequência.

## Capítulo 17 | Frases das páginas 7584-8358

1. Se você errar, pode destruir a máquina virtual, reconstruí-la e tentar novamente sem consequências.
2. Agora respire fundo, abra seu aplicativo de terminal e (o



mais importante) não tenha medo de cometer erros.

3. Compilar o PHP a partir do código-fonte lhe dá a flexibilidade de ajustar a compilação do PHP às suas especificações exatas.

4. O Homebrew é, sem exceção, a minha maneira favorita de instalar o PHP no OS X.

5. Você deve sempre inspecionar o código remoto antes de executá-lo, não importa quão legítima a fonte possa ser.

## **Capítulo 18 | Frases das páginas 8359-8509**

1. Em vez disso, crie um ambiente de desenvolvimento local em uma máquina virtual que esteja isolada com segurança do seu sistema operacional local.

2. Certifique-se de que sua máquina virtual execute o mesmo sistema operacional que o seu servidor de produção.

3. Recomendo que você crie aliases de linha de comando para esses comandos do Vagrant, pois você irá digitá-los com frequência.

4. Você pode provisionar uma máquina virtual com o Vagrant





e o Puppet ou o Chef.

5.Laravel Homestead é uma abstração sobre o Vagrant.

6.PuPHPet é ideal para aqueles que não sabem como escrever manifests do Puppet.

7.Quaisquer alterações locais são refletidas imediatamente na máquina virtual.

**Mais Livros Grátis**



Escaneie para baixar



**Ouça-o**



Baixe o app Bookey para desfrutar

# Mais de 1000 resumos de livros com quizzes

**Teste grátis disponível!**

Escanear para baixar



# PHP Moderno Perguntas

Ver no site do Bookey

## Capítulo 1 | I. Recursos da Linguagem| Perguntas e respostas

### 1.Pergunta

**Qual é a importância da transformação do PHP em uma linguagem de script moderna?**

Resposta:A transformação do PHP sinaliza um renascimento que vem com novos recursos como namespaces, traits e closures. Essas melhorias permitem um código mais limpo e mais fácil de manter, além de um melhor desempenho através de aprimoramentos como o cache de opcode embutido. Essa evolução capacita os desenvolvedores a criar aplicações web modernas com maior eficiência e flexibilidade.

### 2.Pergunta

**Como o papel do Composer mudou o desenvolvimento em PHP?**

Resposta:O Composer revolucionou o desenvolvimento em

Mais Livros Grátis



Escaneie para baixar



Ouça-o

PHP ao permitir que os desenvolvedores gerenciassem dependências facilmente e integrassem componentes menores e especializados, em vez de ficarem presos a frameworks monolíticos. Essa flexibilidade possibilita a combinação de componentes interoperáveis que atendem melhor às necessidades específicas de cada projeto, promovendo uma abordagem mais modular ao desenvolvimento de aplicações.

### 3.Pergunta

**Qual papel importante Rasmus Lerdorf desempenhou no desenvolvimento do PHP desde o início?**

Resposta:Rasmus Lerdorf criou o PHP como uma coleção de scripts CGI para rastrear visitas ao seu currículo online, estabelecendo as bases para o que se tornaria uma linguagem de programação robusta. Seu conjunto inicial de ferramentas, chamado 'Personal Home Page Tools', lançou as fundações do PHP ao permitir variáveis rudimentares e manipulação de formulários.

### 4.Pergunta

Mais Livros Grátis



Escaneie para baixar



Ouça-o

## **Por que entender a história do PHP é importante para os desenvolvedores modernos?**

Resposta: Entender a história do PHP fornece contexto para sua evolução de scripts simples a uma linguagem de programação poderosa capaz de servir conteúdo web dinâmico. Isso enfatiza a importância de aprender com os desafios e práticas do passado, para que os desenvolvedores possam apreciar plenamente os recursos modernos e como utilizá-los efetivamente em seu fluxo de trabalho de desenvolvimento.

### **5. Pergunta**

## **Quais avanços melhoraram as práticas de desenvolvimento no PHP moderno?**

Resposta: As práticas de desenvolvimento em PHP moderno melhoraram significativamente com a adoção de sistemas de controle de versão como o Git, que ajudam a gerenciar mudanças de código, e a utilização de ambientes de desenvolvimento local que refletem configurações de produção. Além disso, recursos como PHPUnit para testes e



Composer para gerenciamento de dependências contribuem para bases de código mais limpas e confiáveis.

## 6.Pergunta

**O que a emergência de motores de PHP concorrentes como o HHVM significa para a comunidade PHP?**

Resposta:A emergência de motores de PHP concorrentes como o HHVM introduz uma competição benéfica, impulsionando melhorias nos motores de PHP existentes, como o Zend Engine. Essa rivalidade incentiva a inovação e eficiência, garantindo que o PHP permaneça moderno, rápido e capaz de atender às demandas dos desenvolvedores em um cenário de software em rápida evolução.

## 7.Pergunta

**Como a introdução da tipagem estática na linguagem de programação Hack beneficia os desenvolvedores de PHP?**

Resposta:A introdução da tipagem estática na Hack proporciona aos desenvolvedores de PHP a possibilidade de verificação de tipo em tempo de compilação, o que melhora a confiabilidade e previsibilidade do código. Isso permite uma

Mais Livros Grátis



Escaneie para baixar



Ouça-o



maior estabilidade nas aplicações, enquanto ainda permite os recursos dinâmicos do PHP, fechando a lacuna para desenvolvedores que buscam tanto um desenvolvimento rápido quanto segurança de tipos.

## 8.Pergunta

**Qual é a importância da especificação do PHP introduzida em 2014?**

Resposta:A especificação do PHP introduzida em 2014 é significativa porque fornece um modelo canônico para a linguagem, garantindo compatibilidade entre vários motores de PHP e promovendo a adesão a padrões da comunidade. Essa formalização ajuda a orientar os desenvolvedores na criação de aplicações robustas e fomenta um ambiente consistente para o desenvolvimento em PHP.

## 9.Pergunta

**O que a frase 'é um momento empolgante para ser um programador PHP' implica sobre o futuro do PHP?**

Resposta:Essa frase implica um período de rápido crescimento, inovação e engajamento da comunidade dentro

Mais Livros Grátis



Escaneie para baixar



Ouça-o

do ecossistema PHP. Com o desenvolvimento contínuo, melhorias em desempenho e usabilidade, e uma forte rede de suporte da comunidade, os programadores PHP podem esperar novas oportunidades, ferramentas e práticas que elevam ainda mais a linguagem.

## 10.Pergunta

**Como o PHP moderno garante um melhor desempenho das aplicações?**

Resposta:O PHP moderno incorpora recursos como cache de opcode, que otimiza a execução de scripts armazenando o bytecode do script pré-compilado na memória. Isso reduz a carga no servidor e acelera os tempos de resposta, juntamente com melhorias de novos motores e tecnologias, contribuindo para um aumento geral na eficiência e escalabilidade das aplicações.

## Capítulo 2 | 1. O Novo PHP| Perguntas e respostas

### 1.Pergunta

**Qual é um recurso significativo introduzido no PHP 5.3 que ajuda a organizar o código e prevenir conflitos de nomenclatura?**

Mais Livros Grátis



Escaneie para baixar



Ouça-o

Resposta:Namespaces! Eles organizam o código PHP em uma hierarquia virtual e garantem que nomes comuns de classes não entrem em conflito entre diferentes bibliotecas e frameworks.

## 2.Pergunta

**Como os namespaces melhoram a colaboração entre desenvolvedores que utilizam diferentes componentes?**

Resposta:Os namespaces permitem que vários desenvolvedores criem e utilizem classes e funções com os mesmos nomes sem colisão, já que cada classe/função é categorizada sob seu próprio namespace único.

## 3.Pergunta

**Qual é o principal objetivo do recurso conhecido como traits no PHP 5.4?**

Resposta:Traits possibilitam a reutilização de código ao permitir que os desenvolvedores definam métodos reutilizáveis que podem ser incluídos em múltiplas classes, sem forçar essas classes a uma hierarquia comum de herança.

## 4.Pergunta

**Você pode explicar o conceito de 'programar para uma**

Mais Livros Grátis



Escaneie para baixar



Ouça-o

## **interface' no PHP e seus benefícios?**

Resposta: Programar para uma interface significa programar contra uma interface em vez de uma implementação específica. Isso permite que seu código funcione com qualquer classe que implemente a interface, promovendo um código flexível e uma integração mais fácil com componentes de terceiros.

## **5.Pergunta**

### **O que é um gerador em PHP e como ele otimiza o uso de memória?**

Resposta: Um gerador é um iterador especial que retorna valores um por um, permitindo que grandes conjuntos de dados sejam processados sem carregá-los completamente na memória, economizando assim recursos.

## **6.Pergunta**

### **O que são closures em PHP e como elas diferem de funções regulares?**

Resposta: Closures são funções anônimas que capturam seu contexto circundante. Ao contrário das funções regulares, as



closures podem reter o acesso a variáveis que estavam no escopo quando foram definidas, mesmo após seu contexto ter saído do escopo.

## 7.Pergunta

**Por que o Zend OPcache é significativo em aplicações PHP modernas?**

Resposta:O Zend OPcache aumenta significativamente o desempenho da aplicação ao armazenar em cache o bytecode pré-compilado, reduzindo a sobrecarga de análise e compilação do código PHP a cada requisição.

## 8.Pergunta

**Que ferramenta os desenvolvedores PHP podem usar para servir aplicações localmente sem um servidor web completo?**

Resposta:O PHP possui um servidor web embutido desde a versão 5.4, que pode ser utilizado para desenvolvimento local para visualizar aplicações PHP facilmente.

## 9.Pergunta

**Por que não é recomendado usar o servidor embutido do PHP em produção?**

Mais Livros Grátis



Escaneie para baixar



Ouça-o

Resposta:O servidor embutido executa todas as requisições de forma sequencial, levando a um desempenho ruim sob carga. Ele não é projetado para lidar com requisições simultâneas, o que pode resultar em atrasos na aplicação.

## 10.Pergunta

**O que um desenvolvedor deve ter em mente ao importar múltiplos namespaces ou classes em um arquivo?**

Resposta:É melhor importar cada namespace ou classe em linhas separadas para clareza e manutenibilidade, ao invés de combiná-los em uma única linha.

## Capítulo 3 | 2. Recursos| Perguntas e respostas

### 1.Pergunta

**Por que os namespaces são considerados um recurso essencial no PHP Moderno?**

Resposta:Os namespaces criam um ambiente isolado para o código, permitindo que ele coexistam com o código de outros desenvolvedores sem risco de colisões de nomes. Eles estruturam o código em uma hierarquia virtual, semelhante a diretórios em um





sistema de arquivos, garantindo que classes com os mesmos nomes de diferentes fontes não entrem em conflito.

## 2.Pergunta

### **Como declarar um namespace em PHP?**

Resposta:Um namespace é declarado no início de um arquivo PHP da seguinte forma: 'namespace VendorName;'. Isso deve ser colocado em uma linha imediatamente após a tag de abertura '<?php'.

## 3.Pergunta

### **O que significa 'programar para uma interface'?**

Resposta:Programar para uma interface refere-se a projetar seu código para depender de interfaces em vez de implementações específicas. Isso permite que seu código funcione com qualquer classe que implemente a interface, aumentando a flexibilidade e facilitando a integração com componentes de terceiros.

## 4.Pergunta

### **Você pode fornecer um exemplo de uso de traits em PHP?**

Mais Livros Grátis



Escaneie para baixar



Ouça-o

Resposta:Um trait em PHP é definido com a palavra-chave 'trait'. Aqui está um exemplo:

```
...  
  
trait ExampleTrait {  
    public function sayHello() {  
        return 'Olá!';  
    }  
}  
  
class ExampleClass {  
    use ExampleTrait;  
}  
...
```

Neste exemplo, a ExampleClass pode usar o método sayHello() definido em ExampleTrait.

## 5.Pergunta

**Quais são os benefícios de usar geradores?**

Resposta:Os geradores permitem que você crie iteradores



sem a sobrecarga de definir uma classe inteira. Eles geram valores sob demanda, reduzindo significativamente o uso de memória ao gerar valores um de cada vez, sendo especialmente úteis ao iterar sobre grandes conjuntos de dados.

## 6.Pergunta

### **Como funcionam as closures em PHP?**

Resposta:As closures são funções anônimas que podem capturar variáveis de seu contexto circundante, preservando efetivamente seu estado mesmo após a execução da saída do escopo. Você pode anexar variáveis a uma closure usando a palavra-chave 'use' para garantir que elas mantenham seus valores.

## 7.Pergunta

### **O que é um autoloader em PHP e como os namespaces estão relacionados a ele?**

Resposta:Um autoloader em PHP carrega automaticamente classes quando são necessárias, sem exigir instruções explícitas de include ou require. Os namespaces facilitam



esse processo através do padrão de autoloading PSR-4, que mapeia namespaces para estruturas de diretórios.

## 8.Pergunta

**Por que você deve evitar usar várias classes em um único arquivo?**

Resposta: Armazenar várias classes em um único arquivo pode levar a confusão e violar as melhores práticas, como a recomendação de uma classe por arquivo. Isso complica a manutenção e a legibilidade, tornando mais difícil resolver problemas e gerenciar o código.

## 9.Pergunta

**Explique a diferença entre importar e aliasar em namespaces PHP.**

Resposta: Importar torna classes, funções ou constantes específicas disponíveis no arquivo atual, para que possam ser referenciadas sem seu namespace completo. Aliasing permite que você atribua um nome mais curto a elas, tornando o código mais conciso. Por exemplo: ``use Vendor namespace \Vendor\Namespace\Class as Short;`` permite que você use ``Short``



em vez do nome completo da classe.

**Mais Livros Grátis**



Escaneie para baixar



Ouça-o



Ad



Escanear para baixar



App Store  
Escolha dos Editores



22k avaliações de 5 estrelas

## Feedback Positivo

Afonso Silva

... cada resumo de livro não só  
..., mas também tornam o  
... divertido e envolvente. O  
... tou a leitura para mim.

**Fantástico!**



Estou maravilhado com a variedade de livros e idiomas  
que o Bookey suporta. Não é apenas um aplicativo, é  
um portal para o conhecimento global. Além disso,  
ganhar pontos para caridade é um grande bônus!

Brígida Santos

F



O  
só  
o  
O

na Oliveira

... correr as  
... ém me dá  
... comprar a  
... ar!

**Adoro!**



Usar o Bookey ajudou-me a cultivar um hábito de  
leitura sem sobrecarregar minha agenda. O design do  
aplicativo e suas funcionalidades são amigáveis,  
tornando o crescimento intelectual acessível a todos.

Duarte Costa

**Economiza tempo!**



O Bookey é o meu apli  
crescimento intelectual  
perspicazes e lindame  
um mundo de conheci

**Aplicativo incrível!**



Eu amo audiolivros, mas nem sempre tenho tempo para  
ouvir o livro inteiro! O Bookey permite-me obter um resumo  
dos destaques do livro que me interessa!!! Que ótimo  
conceito!!! Altamente recomendado!

Estevão Pereira

**Aplicativo lindo**



Este aplicativo é um salva-vidas para  
de livros com agendas lotadas. Os re  
precisos, e os mapas mentais ajudar  
o que aprendi. Altamente recomend

Teste gratuito com Bookey





## Capítulo 4 | II. Boas Práticas| Perguntas e respostas

### 1.Pergunta

#### **Qual é a importância do Grupo de Interoperabilidade de Frameworks PHP (PHP-FIG)?**

Resposta:O PHP-FIG desempenha um papel vital na melhoria da comunicação e interoperabilidade entre vários frameworks PHP. Fundado por desenvolvedores de frameworks que identificaram os desafios de ambientes isolados, o PHP-FIG cria recomendações que permitem que frameworks compartilhem as mesmas bibliotecas e componentes, levando a uma maior eficiência e colaboração em todo o ecossistema PHP.

### 2.Pergunta

#### **Como as interfaces contribuem para a interoperabilidade em frameworks PHP?**

Resposta:As interfaces permitem que diferentes frameworks PHP trabalhem juntos ao definir um contrato para métodos compartilhados. Isso significa que um framework pode usar



componentes de terceiros (como loggers), contanto que eles sigam uma interface especificada. Isso incentiva o desenvolvimento e o uso de componentes especializados, reduzindo significativamente a redundância entre os frameworks.

### 3.Pergunta

**Por que a autoloading é importante no desenvolvimento PHP moderno?**

Resposta:A autoloading simplifica o processo de gerenciamento de dependências em aplicações PHP. Ela permite que os desenvolvedores carreguem classes automaticamente quando necessário, eliminando a necessidade de incluir arquivos manualmente. Isso não apenas torna o código mais simples, mas também apoia uma abordagem mais organizada e modular para o desenvolvimento.

### 4.Pergunta

**O que são os padrões PSR e por que são essenciais?**

Resposta:PSR significa Recomendações de Padrões PHP, que

Mais Livros Grátis



Escaneie para baixar



Ouça-o

são diretrizes estabelecidas pelo PHP-FIG para padronizar práticas de codificação em vários projetos PHP. Elas fornecem uma base para interoperabilidade, garantindo que os desenvolvedores possam trabalhar de forma integrada com diferentes frameworks e componentes sem precisar reaprender estilos ou padrões de codificação. Seguir os padrões PSR melhora a legibilidade e a manutenibilidade do código.

## 5.Pergunta

**Como o PSR-3 melhora o registro em aplicações PHP?**

Resposta:O PSR-3 define uma interface comum para registro, permitindo que os desenvolvedores criem componentes de registro compatíveis ou integrem facilmente os existentes, como o Monolog. Isso aumenta a flexibilidade das aplicações PHP, pois os desenvolvedores não ficam presos a uma única solução de registro, podendo escolher ou alternar loggers conforme suas necessidades, sem alterar a arquitetura da aplicação.

## 6.Pergunta

Mais Livros Grátis



Escaneie para baixar



Ouça-o

## **Por que os desenvolvedores devem adotar o estilo de codificação estrito do PSR-2?**

Resposta:O estilo de codificação do PSR-2 promove a consistência e a legibilidade em bases de código, especialmente em ambientes colaborativos com múltiplos colaboradores. Ao aderir a um guia de estilo comum, os desenvolvedores garantem que seu código seja facilmente compreensível, reduzindo a curva de aprendizado para novos colaboradores e ajudando na resolução mais rápida de problemas e correção de bugs.

### **7.Pergunta**

## **Qual é o benefício de usar o PSR-4 para autoloading em aplicações PHP?**

Resposta:O PSR-4 fornece uma abordagem padronizada para autoloading de classes, permitindo que os desenvolvedores organizem seu código de maneira previsível. Isso significa que o PHP pode localizar e carregar classes sem a necessidade de funções de autoload personalizadas, resultando em aplicações mais limpas e eficientes, que são



mais fáceis de manter e expandir.

## 8.Pergunta

**Como os padrões PHP podem aumentar a produtividade de um desenvolvedor?**

Resposta:Ao estabelecer diretrizes e práticas comuns, os padrões PHP reduzem a curva de aprendizado para novas tecnologias e frameworks. Os desenvolvedores passam menos tempo tentando descobrir como implementar várias ferramentas e podem se concentrar mais na construção de recursos e na melhoria das aplicações, levando a uma maior produtividade e tempos de entrega mais rápidos.

## 9.Pergunta

**Qual é o impacto dos estilos de código padronizados na colaboração da equipe?**

Resposta:Os estilos de código padronizados reduzem atritos na colaboração em equipe, garantindo que todos os membros possam ler e entender facilmente o código uns dos outros. Isso leva a uma maior eficiência nas revisões de código, depuração e contribuições, pois os desenvolvedores não

**Mais Livros Grátis**



Escaneie para baixar



**Ouça-o**

gastam tempo se acostumando a diferentes estilos de codificação.

## 10.Pergunta

**Como a evolução de frameworks isolados para um ecossistema distribuído beneficia os desenvolvedores PHP?**

Resposta:A mudança para um ecossistema distribuído permite que os desenvolvedores PHP aproveitem uma ampla variedade de componentes e bibliotecas especializadas, promovendo inovação e criatividade. Os desenvolvedores não estão mais restritos pelas limitações de um único framework; em vez disso, podem escolher as melhores ferramentas para as necessidades específicas de seu projeto, levando a aplicações mais robustas e eficazes.

## Capítulo 5 | 3. Padrões| Perguntas e respostas

### 1.Pergunta

**Qual é o problema com os frameworks PHP mais antigos que foram desenvolvidos isoladamente?**

Resposta:Os frameworks PHP mais antigos não foram projetados para se comunicar entre si,

Mais Livros Grátis



Escaneie para baixar



Ouça-o



resultando em ineficiências e limitações na criatividade dos desenvolvedores. Muitas vezes, os desenvolvedores ficam presos ao ecossistema de um único framework, que pode não fornecer todas as ferramentas ou bibliotecas necessárias para seus projetos.

## 2.Pergunta

**Como o PHP Framework Interop Group (PHP-FIG) ajudou a modernizar os frameworks PHP?**

Resposta:O PHP-FIG foi criado por desenvolvedores de frameworks em 2009 para abordar a interoperabilidade e a comunicação entre frameworks. Eles estabeleceram padrões, como as recomendações PSR, que permitem que frameworks compartilhem componentes e bibliotecas, melhorando a colaboração e a eficiência.

## 3.Pergunta

**O que significa PSR e por que são importantes?**

Resposta:PSR significa Recomendações de Padrões PHP.

Elas são importantes porque estabelecem diretrizes para o

Mais Livros Grátis



Escaneie para baixar



Ouça-o

estilo de código, autoloading e interfaces de registro, ajudando a reduzir inconsistências entre os frameworks PHP e facilitando o trabalho dos desenvolvedores em diferentes projetos.

#### 4.Pergunta

**O que é PSR-1 e por que é recomendado para desenvolvedores PHP?**

Resposta:PSR-1 é a recomendação de Estilo Básico de Código que fornece diretrizes simples para ajudar os desenvolvedores a escrever código PHP consistente com os padrões. É fácil de implementar e serve como uma base para escrever código que pode ser facilmente compreendido e mantido por outros.

#### 5.Pergunta

**Qual é a importância do PSR-4 no desenvolvimento PHP?**

Resposta:O PSR-4 define um processo de autoloading padronizado que permite que classes, interfaces e traits PHP sejam carregados automaticamente com base em seus namespaces e estrutura de diretórios. Isso simplifica muito a

Mais Livros Grátis



Escaneie para baixar



Ouça-o

gestão de arquivos de classe e ajuda a evitar problemas comuns com a inclusão manual de arquivos.

## 6.Pergunta

**O que você deve fazer se quiser usar um componente em um projeto PHP sem reinventar a roda?**

Resposta: Você deve procurar por componentes PHP já existentes no Packagist que possam resolver seu problema específico, em vez de escrever a funcionalidade do zero. Isso aproveita o trabalho de outros desenvolvedores e acelera seu processo de desenvolvimento.

## 7.Pergunta

**Como o Composer se relaciona com os componentes PHP?**

Resposta: O Composer é um gerenciador de dependências para PHP que simplifica a instalação e a gestão de componentes PHP. Ele resolve automaticamente e baixa as dependências de um projeto, cuida do autoloading e garante que as versões corretas dos componentes sejam utilizadas.

## 8.Pergunta

**Quais são algumas características de bons componentes**

Mais Livros Grátis



Escaneie para baixar



Ouça-o

## PHP?

Resposta: Bons componentes PHP são focados em resolver um único problema, pequenos em tamanho, cooperativos com outros componentes, bem testados e bem documentados. Essas qualidades ajudam a garantir que sejam fáceis de usar e integrar em projetos.

### 9. Pergunta

**Por que os desenvolvedores PHP devem ter cautela ao escolher os frameworks?**

Resposta: Os desenvolvedores devem considerar o potencial para que os frameworks limitem a flexibilidade e a inovação, além da possibilidade de um framework se tornar obsoleto ou abandonado. Eles devem avaliar cuidadosamente as necessidades de seus projetos para escolher a solução de framework ou componente apropriada.

### 10. Pergunta

**Qual é o propósito do arquivo composer.json em um componente PHP?**

Resposta: O arquivo composer.json define os metadados e as



dependências de um componente PHP, orientando o Composer na instalação, gestão e autoloading do componente. Ele também inclui informações necessárias para a integração do componente com o Packagist.

## **Capítulo 6 | 4. Componentes| Perguntas e respostas**

### **1.Pergunta**

**Por que é mais vantajoso usar componentes em vez de frameworks no PHP Moderno?**

Resposta: Usar componentes proporciona flexibilidade e modularização. Os desenvolvedores podem selecionar apenas os componentes necessários para as exigências específicas de suas aplicações, o que resulta em aplicações mais leves e eficientes. Os frameworks, por outro lado, podem ser pesados e restritivos, exigindo a adesão às convenções do framework, que pode não ser adequada para todos os projetos.

### **2.Pergunta**

**O que define um componente no PHP Moderno?**

**Mais Livros Grátis**



Escaneie para baixar



**Ouçá-o**

Resposta:Um componente é um pequeno pacote de código, focado em resolver um problema específico dentro da sua aplicação PHP, consistindo em classes, interfaces e traits relacionadas que geralmente vivem sob um namespace comum.

### 3.Pergunta

**Quais são as características de um bom componente PHP?**

Resposta:Bons componentes PHP são focados em uma única tarefa, pequenos, cooperativos com outros, bem testados e bem documentados.

### 4.Pergunta

**O que você deve considerar ao decidir usar um framework em vez de componentes?**

Resposta:Considere o tamanho do projeto e a estrutura da equipe; use componentes para projetos menores onde é necessário um controle preciso, enquanto opte por frameworks em projetos maiores que se beneficiam de convenções e estruturas estabelecidas.





## 5.Pergunta

**Como você encontra e instala componentes PHP?**

Resposta:Componentes PHP podem ser encontrados e instalados usando o Packagist, o diretório principal de componentes PHP, junto com o Composer para gerenciar dependências e autoloading.

## 6.Pergunta

**Você pode explicar o que o Composer faz?**

Resposta:O Composer é um gerenciador de dependências que ajuda a gerenciar e instalar componentes PHP de forma simples, resolvendo dependências e autoloading de classes para o seu projeto automaticamente.

## 7.Pergunta

**Quais são algumas melhores práticas para criar seus próprios componentes PHP?**

Resposta:Assegure-se de que seus componentes tenham nomes de vendor/package únicos, use um namespace claro, mantenha uma documentação adequada com instruções de instalação e uso, e mantenha seu código modular e testável.

## 8.Pergunta

Mais Livros Grátis



Escaneie para baixar



Ouça-o

## **O que você deve fazer se encontrar um componente PHP que você gosta?**

Resposta: Você deve dar um star no componente no Packagist e compartilhá-lo para ajudar a ganhar visibilidade entre outros desenvolvedores.

### **9.Pergunta**

## **O que é importante incluir no arquivo composer.json do seu componente?**

Resposta: Inclua propriedades como nome do componente, descrição, palavras-chave, homepage, licença, autores, informações de suporte, dependências e configuração de autoloading.

### **10.Pergunta**

## **Por que é importante entender o Versionamento Semântico ao trabalhar com componentes PHP?**

Resposta: O Versionamento Semântico ajuda a delinear a natureza das mudanças de versão para versão, garantindo que os desenvolvedores compreendam o impacto das atualizações sobre a compatibilidade retroativa.





# Ler, Compartilhar, Empoderar

Conclua Seu Desafio de Leitura, Doe Livros para Crianças Africanas.

## O Conceito



Esta atividade de doação de livros está sendo realizada em conjunto com a Books For Africa. Lançamos este projeto porque compartilhamos a mesma crença que a BFA: Para muitas crianças na África, o presente de livros é verdadeiramente um presente de esperança.

## A Regra



Ganhe 100 pontos



Resgate um livro



Doe para a África

Seu aprendizado não traz apenas conhecimento, mas também permite que você ganhe pontos para causas beneficentes! Para cada 100 pontos ganhos, um livro será doado para a África.

Teste gratuito com Bookee





## Capítulo 7 | 5. Boas Práticas| Perguntas e respostas

### 1.Pergunta

**Por que é importante sanitizar os dados de entrada em aplicações PHP?**

Resposta:Sanitizar os dados de entrada protege sua aplicação de possíveis vetores de ataque, onde códigos maliciosos ou indesejados, como injeções de SQL ou ataques XSS via scripts, podem ser introduzidos através das entradas do usuário. Por exemplo, falhar em sanitizar uma entrada HTML pode levar à execução de scripts maliciosos, prejudicando sua aplicação e expondo dados sensíveis dos usuários.

### 2.Pergunta

**Quais são os três processos-chave para garantir um manuseio seguro de dados em PHP?**

Resposta:Os três processos críticos são: 1. **\*\*Sanitização\*\***: Modificar os dados de entrada para escapar ou remover caracteres inseguros antes que eles cheguem à camada de



armazenamento da aplicação. 2. **\*\*Validação\*\***: Verificar se os dados de entrada atendem aos formatos e critérios esperados. 3. **\*\*Escapamento de Saída\*\***: Garantir que qualquer dado renderizado na saída HTML esteja devidamente codificado para prevenir a execução de scripts maliciosos.

### 3.Pergunta

**Qual é a diferença entre sanitização e validação de dados?**

Resposta:Sanitização envolve alterar os dados de entrada para remover ou codificar caracteres inseguros, efetivamente limpando os dados antes de serem processados ou armazenados. Validação, por outro lado, verifica se a entrada corresponde aos formatos e tipos esperados, sem alterar os dados em si, garantindo que apenas informações corretamente formatadas sejam aceitas.

### 4.Pergunta

**Por que os desenvolvedores devem evitar codificar as credenciais do banco de dados em arquivos PHP?**

Resposta:Codificar as credenciais do banco de dados em



arquivos PHP expõe esses detalhes sensíveis a riscos, especialmente se os arquivos forem servidos publicamente devido a configurações inadequadas do servidor. Em vez disso, as credenciais devem ser armazenadas em um arquivo de configuração seguro fora do diretório raiz.

## 5.Pergunta

**Como o PDO aumenta a segurança ao trabalhar com consultas SQL?**

Resposta:O PDO melhora a segurança ao usar declarações preparadas, que permitem que os desenvolvedores separem a lógica SQL dos dados. Isso significa que os dados de entrada são automaticamente sanitizados e vinculados à declaração SQL, reduzindo significativamente o risco de ataques de injeção de SQL.

## 6.Pergunta

**Quais práticas os desenvolvedores devem seguir para lidar com senhas com segurança em PHP?**

Resposta:Os desenvolvedores nunca devem armazenar ou saber as senhas dos usuários em texto simples. Sempre use





algoritmos de hash como bcrypt para hash das senhas antes do armazenamento, garantindo que não possam ser descriptografadas. Além disso, evite impor restrições desnecessárias na complexidade das senhas e nunca envie senhas por email ou outros canais inseguros.

## 7.Pergunta

**Qual é a codificação de caracteres recomendada para usar em aplicações PHP e por quê?**

Resposta:UTF-8 é a codificação de caracteres recomendada porque é amplamente suportada por navegadores web e pode lidar com uma variedade de caracteres de diferentes idiomas. Usar UTF-8 garante compatibilidade e previne a corrupção de dados ao lidar com caracteres multibyte.

## 8.Pergunta

**Qual é o papel dos manipuladores de exceção em aplicações PHP?**

Resposta:Os manipuladores de exceção fornecem um mecanismo para interceptar exceções não capturadas, permitindo que os desenvolvedores exibam mensagens de



erro amigáveis ao usuário e tomem ações corretivas sem revelar informações sensíveis de depuração. Isso ajuda a melhorar a experiência do usuário e mantém a integridade da aplicação durante situações de erro.

## 9.Pergunta

**Por que usar uma biblioteca como o Monolog é vantajoso para registro de logs em aplicações PHP?**

Resposta:O Monolog fornece uma estrutura de registro organizada e flexível que permite aos desenvolvedores registrar mensagens em diferentes níveis de severidade e direcioná-las a várias saídas, como arquivos, bancos de dados ou até mesmo emails. Isso garante um rastreamento e gerenciamento de erros abrangentes, ajudando na depuração e monitoramento da saúde da aplicação.

## Capítulo 8 | III. Implantação, Testes e Ajustes| Perguntas e respostas

### 1.Pergunta

**Quais fatores devo considerar ao escolher um plano de hospedagem para minha aplicação PHP?**

Resposta:Considere as necessidades de recursos da

Mais Livros Grátis



Escaneie para baixar



Ouça-o

sua aplicação, orçamento, escalabilidade e preferências de gestão. Servidores compartilhados são econômicos, mas limitados, enquanto VPS oferece flexibilidade e acesso root. Servidores dedicados são adequados para aplicações exigentes, e PaaS é ideal para quem deseja evitar a gestão.

## 2.Pergunta

**Quais são as desvantagens de usar hospedagem compartilhada para uma aplicação PHP?**

Resposta:A hospedagem compartilhada geralmente carece de opções de personalização, fornece recursos limitados e pode ser afetada negativamente por outras contas no mesmo servidor. Normalmente, também não permite acesso SSH remoto, dificultando a automação e os processos de implantação.

## 3.Pergunta

**Por que um Servidor Privado Virtual (VPS) é recomendado para a maioria das aplicações PHP?**

Resposta:Um VPS fornece recursos dedicados com acesso

Mais Livros Grátis



Escaneie para baixar



Ouça-o

root, permitindo mais controle e personalização sem as limitações da hospedagem compartilhada. É escalável e pode crescer com sua aplicação, tornando-o adequado para tráfego moderado a alto.

#### 4.Pergunta

**Quando um servidor dedicado pode ser uma escolha melhor do que um VPS?**

Resposta:Se sua aplicação PHP tem uma alta demanda e supera a relação custo-benefício de um VPS, um servidor dedicado oferece desempenho superior e a capacidade de gerenciar recursos maiores de forma mais eficiente.

#### 5.Pergunta

**Quais são os principais benefícios de usar uma Plataforma como Serviço (PaaS) para aplicações PHP?**

Resposta:A PaaS permite uma implantação rápida sem a complicação da gestão do servidor. Os usuários podem facilmente escalar os recursos para cima ou para baixo conforme necessário, tornando-a perfeita para pequenas aplicações ou protótipos.

Mais Livros Grátis



Escaneie para baixar



Ouça-o

## 6.Pergunta

**Como posso garantir que minha escolha de hospedagem atenda ao crescimento futuro da minha aplicação PHP?**

Resposta:Escolha um plano de hospedagem que permita escalabilidade—seja através de PaaS ou opções de VPS—e avalie regularmente as necessidades de recursos à medida que o tráfego e a funcionalidade da sua aplicação aumentam.

## 7.Pergunta

**O que você deve fazer se o tráfego da sua aplicação crescer significativamente após começar com uma PaaS?**

Resposta:Se sua aplicação crescer significativamente, considere a transição para um VPS ou servidor dedicado para gerenciar o aumento de recursos e garantir a estabilidade de desempenho.

## 8.Pergunta

**Qual é a importância da gestão do servidor em soluções de hospedagem?**

Resposta:Uma gestão eficaz do servidor garante desempenho, segurança e alocação de recursos otimizados. Soluções como VPS e servidores dedicados exigem mais



gestão e conhecimento técnico, enquanto a PaaS simplifica esse aspecto.

## 9.Pergunta

**Por que é crucial verificar se o seu provedor de hospedagem suporta a versão mais recente do PHP?**

Resposta: Usar a versão mais recente do PHP é vital para segurança, desempenho e acesso a novos recursos. Garantir que seu provedor de hospedagem a suporte ajuda a manter a integridade e a eficiência da sua aplicação.

## Capítulo 9 | 6. Hospedagem| Perguntas e respostas

### 1.Pergunta

**Quais são as quatro principais opções para hospedar uma aplicação PHP, e qual é um benefício chave de cada uma?**

Resposta: 1. **\*\*Servidores Compartilhados\*\***: Opção mais acessível, custando tipicamente de R\$5 a R\$50 por mês. Boa para iniciantes absolutos ou pequenas aplicações.

2. **\*\*Servidores Privados Virtuais (VPS)\*\***: Oferece recursos de sistema dedicados e maior controle,





custando de R\$50 a R\$500 por mês, ideal para a maioria das aplicações PHP.

3. **\*\*Servidores Dedicados\*\***: Proporciona desempenho máximo com controle total do sistema, embora tipicamente seja mais caro (centenas por mês).

4. **\*\*Plataformas como Serviço (PaaS)\*\***: Simplifica a implementação ao gerenciar a infraestrutura para você, custando de R\$50 a R\$500 por mês, e é ótimo para desenvolvedores que preferem não gerenciar servidores.

## 2.Pergunta

**Por que os desenvolvedores devem evitar hospedagem compartilhada para mais do que apenas um site pessoal ou pequenos projetos?**

Resposta:A hospedagem compartilhada compartilha recursos com vários usuários, o que pode levar a problemas de desempenho se outra aplicação no mesmo servidor apresentar mau comportamento (por exemplo, consumindo muita

Mais Livros Grátis



Escaneie para baixar



Ouça-o

memória). Além disso, a hospedagem compartilhada geralmente não permite a personalização do software ou acesso via SSH, limitando a capacidade do desenvolvedor de gerenciar ou implementar aplicações.

### 3.Pergunta

**O que é um Servidor Privado Virtual (VPS) e por que ele é frequentemente recomendado para aplicações PHP?**

Resposta:Um VPS funciona como um servidor dedicado, mas roda em hardware compartilhado. Ele fornece recursos alocados, como CPU e memória, apenas para sua aplicação, concede acesso root, permite flexibilidade na instalação de software e é muito mais adequado para lidar com tráfego moderado em comparação com a hospedagem compartilhada.

### 4.Pergunta

**Quando seria aconselhável atualizar de um VPS para um servidor dedicado?**

Resposta:Se sua aplicação PHP experimentar um crescimento significativo, como ganhar centenas de milhares de visitantes por mês, e o desempenho se tornar um problema ou os custos



do seu VPS se tornarem insustentáveis devido ao maior uso de recursos, a transição para um servidor dedicado é aconselhável para melhor desempenho e estabilidade.

## 5.Pergunta

**Quais são as vantagens de usar Plataformas como Serviço (PaaS) para implantar aplicações PHP?**

Resposta:As opções PaaS removem a complexidade da gestão de servidores. Elas permitem que os desenvolvedores implementem aplicações rapidamente através de interfaces amigáveis, gerenciam automaticamente a escalabilidade e frequentemente vêm com suporte para pacotes de software e ambientes necessários, tornando-as ideais para aqueles focados no desenvolvimento em vez da infraestrutura.

## 6.Pergunta

**Qual deve ser o primeiro passo após selecionar um plano de hospedagem para uma aplicação PHP?**

Resposta:O primeiro passo é provisionar seu servidor, o que envolve configurar e ajustar o servidor de acordo com os requisitos da sua aplicação, incluindo a instalação de

Mais Livros Grátis



Escaneie para baixar



Ouça-o

software necessário, como um servidor web e gerenciadores de processo PHP.

## 7.Pergunta

**Qual é o benefício de usar autenticação por par de chaves SSH em vez de autenticação por senha para acesso ao servidor?**

Resposta:A autenticação por par de chaves SSH é mais segura do que a autenticação por senha, pois previne ataques de força bruta. Ela utiliza chaves criptográficas para autenticar usuários, tornando o acesso não autorizado extremamente difícil.

## 8.Pergunta

**O que é PHP-FPM e por que é importante para o gerenciamento de aplicações PHP?**

Resposta:O PHP-FPM (FastCGI Process Manager) gerencia pools de processos PHP para lidar com solicitações web de maneira eficiente, permitindo melhor desempenho e gestão de recursos em comparação com métodos tradicionais. Ele pode escalar processos PHP com base na demanda, o que é essencial para aplicações de alto tráfego.



## 9.Pergunta

**Como o uso do Nginx difere do Apache para aplicações PHP?**

Resposta:O Nginx é frequentemente preferido em relação ao Apache para aplicações PHP devido à sua arquitetura leve e capacidade de servir arquivos estáticos de forma mais eficiente, enquanto utiliza o PHP-FPM para gerenciar solicitações dinâmicas de maneira mais eficiente em termos de recursos.

## 10.Pergunta

**Quais práticas podem aumentar a segurança do servidor após provisionar um ambiente de hospedagem PHP?**

Resposta:A segurança pode ser aprimorada criando um usuário não-root para logins diários, desativando o acesso de login root, implementando autenticação por par de chaves SSH e atualizando regularmente o software e aplicando regras de firewall.







# As melhores ideias do mundo desbloqueiam seu potencial

Essai gratuit avec Bookey



Escanear para baixar





## Capítulo 10 | 7. Provisionamento| Perguntas e respostas

### 1.Pergunta

**Qual é o propósito do provisionamento de servidores no contexto da implementação de uma aplicação PHP?**

Resposta:O provisionamento de servidores é o processo de configurar e preparar um ambiente de servidor para executar uma aplicação PHP. Envolve adquirir um servidor, instalar software necessário, como o servidor web (por exemplo, nginx), e configurar serviços essenciais, como o PHP-FPM, para gerenciar as solicitações da aplicação de forma eficiente. Um bom provisionamento garante que a aplicação funcione de maneira fluida e segura.

### 2.Pergunta

**Por que um desenvolvedor pode preferir usar uma Plataforma como Serviço (PaaS) para a implementação de aplicações PHP?**

Resposta:Usar uma PaaS simplifica o processo de implementação porque o provedor gerencia a infraestrutura



do servidor, permitindo que os desenvolvedores se concentrem mais na codificação, em vez de nas tarefas de administração do sistema. As opções de PaaS normalmente oferecem escalabilidade embutida, backups, atualizações de segurança e outros serviços que podem ser complexos de gerenciar em um servidor dedicado ou VPS.

### 3.Pergunta

**Quais são os passos principais a serem seguidos após adquirir um servidor virtual privado (VPS) para implementar uma aplicação PHP?**

Resposta: Após adquirir um VPS, os passos principais incluem: 1) Fazer login no servidor usando SSH, 2) Atualizar o sistema operacional com as atualizações de software, 3) Criar um usuário não-root para maior segurança, 4) Configurar autenticação por chave SSH, e 5) Instalar e configurar software essencial como nginx e PHP-FPM para servir sua aplicação PHP.

### 4.Pergunta

**Qual configuração de servidor web tradicional era comumente usada antes do surgimento do nginx e do**



## **PHP-FPM?**

Resposta: Tradicionalmente, muitos desenvolvedores usavam o servidor web Apache com o módulo `mod_php` para servir aplicações PHP. Essa configuração envolvia o Apache gerando um processo filho separado para cada solicitação e incorporando um interpretador PHP em cada processo, o que resultava em alto consumo de recursos.

### **5.Pergunta**

**Como o nginx difere do Apache em termos de eficiência de recursos?**

Resposta: O nginx melhora a eficiência de recursos usando uma arquitetura orientada a eventos que lida com várias solicitações em um único processo de trabalho, em vez de gerar processos separados para cada solicitação, como faz o Apache. Isso reduz o uso de memória e oferece um desempenho melhor, especialmente em alta carga de tráfego.

### **6.Pergunta**

**O que é o PHP-FPM e qual é o seu papel no manuseio de solicitações PHP?**

**Mais Livros Grátis**



Escaneie para baixar



**Ouça-o**

Resposta:O PHP-FPM (Gerenciador de Processos FastCGI do PHP) é um software que gerencia um pool de processos PHP para lidar com as solicitações recebidas do servidor web (como o nginx). Ele permite que vários processos PHP sejam utilizados de forma eficiente, melhorando o manuseio de solicitações concorrentes e a gestão de recursos.

## 7.Pergunta

**Qual recomendação de segurança é fornecida ao configurar um novo servidor para implementar aplicações PHP?**

Resposta:Uma recomendação crucial de segurança é criar um usuário não-root para acesso ao servidor em vez de usar a conta root, que possui poder irrestrito. Essa prática minimiza os riscos de segurança e ajuda a evitar mudanças acidentais que podem desestabilizar o servidor.

## 8.Pergunta

**Quais benefícios a automação do provisionamento de servidores com ferramentas como Ansible ou Puppet oferece?**

Resposta:Automatizar o provisionamento de servidores com

Mais Livros Grátis



Escaneie para baixar



Ouça-o

ferramentas como Ansible ou Puppet economiza tempo e reduz o potencial de erros na configuração. Isso permite processos consistentes e repetíveis, facilitando o gerenciamento de vários servidores e garantindo que todos estejam configurados com as mesmas práticas recomendadas.

## 9.Pergunta

**Qual é o propósito de ajustar as configurações de PHP e como isso se relaciona com o desempenho?**

Resposta:Ajustar as configurações do PHP otimiza o desempenho ao alinhar a alocação de recursos com as necessidades específicas da aplicação. Um ajuste adequado pode melhorar os tempos de resposta, reduzir o consumo de memória e aumentar a estabilidade geral, fazendo com que as aplicações funcionem de maneira mais eficiente sob cargas esperadas.

## 10.Pergunta

**Por que é importante monitorar o uso de memória por processo PHP ao afinar o servidor?**

Resposta:Monitorar o uso de memória por processo PHP é



importante porque ajuda a determinar quantos processos PHP-FPM podem ser executados simultaneamente sem exceder a memória disponível do sistema. Isso garante um desempenho ideal do servidor, evitando falhas ou lentidão devido à exaustão de memória.

## **11.Pergunta**

**Como a desativação da autenticação por senha pode aprimorar a segurança do servidor durante o acesso remoto?**

Resposta:Desativar a autenticação por senha aprimora a segurança do servidor ao prevenir potenciais ataques de força bruta, onde invasores tentam adivinhar senhas repetidamente. Em vez disso, usar a autenticação por chave SSH fornece um método mais seguro de acessar o servidor, pois se baseia em chaves criptográficas, em vez de senhas facilmente adivinháveis.

## **12.Pergunta**

**O que é o Zend OPcache e como ele beneficia as aplicações PHP?**

Resposta:O Zend OPcache é um mecanismo de cache de

**Mais Livros Grátis**



Escaneie para baixar



**Ouça-o**



opcode que armazena o bytecode compilado do PHP na memória, reduzindo a sobrecarga de análise e compilação de scripts PHP em cada solicitação. Isso leva a uma execução mais rápida das aplicações PHP e a um desempenho geral aprimorado.

### **13.Pergunta**

**O que deve ser considerado ao determinar o limite de memória para aplicações PHP?**

Resposta:Ao determinar o limite de memória para aplicações PHP, os desenvolvedores devem considerar a memória total disponível do sistema, o consumo de memória de outros processos que estão rodando no servidor, o uso médio de memória dos processos PHP e o desempenho esperado da aplicação sob diferentes cargas.

### **14.Pergunta**

**Por que é desencorajado permitir um grande número de uploads de arquivos em aplicações PHP?**

Resposta:É desencorajado permitir um grande número de uploads de arquivos em aplicações PHP para melhorar a

**Mais Livros Grátis**



Escaneie para baixar



**Ouça-o**

segurança e gerenciar o uso de recursos de maneira eficaz. Limitar os uploads evita a sobrecarga do servidor, reduz o potencial de abuso e pode simplificar o tratamento de erros ao lidar com entradas dos usuários.

## **Capítulo 11 | 8. Ajustando Perguntas e respostas**

### **1.Pergunta**

**Qual é o objetivo de ajustar a configuração do PHP com definições apropriadas para sua aplicação e servidor de produção?**

Resposta:O objetivo de ajustar a configuração do PHP é melhorar a eficiência do PHP e o desempenho da aplicação, semelhante a como um terno sob medida se ajustaria melhor do que um genérico. É um passo crucial para aumentar o desempenho geral de uma aplicação PHP.

### **2.Pergunta**

**Como a configuração `memory_limit` no `php.ini` impacta as aplicações PHP?**

Resposta:A configuração `memory_limit` especifica a quantidade máxima de memória que um único processo PHP



pode usar. Isso impacta na eficiência com que sua aplicação opera; um valor muito baixo pode resultar em erros de memória para aplicações maiores, enquanto um valor muito alto pode levar a um desperdício de recursos.

### 3.Pergunta

**O que é o Zend OPcache e por que ele é benéfico para aplicações PHP?**

Resposta:O Zend OPcache é um cache de opcode que melhora o desempenho ao armazenar scripts PHP compilados em cache, reduzindo o overhead de leitura e compilação a cada requisição. Isso garante que a aplicação responda mais rapidamente às requisições recebidas.

### 4.Pergunta

**Por que os uploads de arquivos devem ser configurados com limites em aplicações PHP?**

Resposta:Os uploads de arquivos devem ser limitados em tamanho e número para aumentar a segurança da aplicação e garantir um melhor desempenho. Sem limites, sua aplicação pode se tornar vulnerável ao consumo excessivo de recursos

**Mais Livros Grátis**



Escaneie para baixar



**Ouça-o**

ou a exploração.

## 5.Pergunta

**Qual é a importância de definir `max_execution_time` no `php.ini`?**

Resposta:A configuração `max_execution_time` limita quanto tempo um processo PHP pode ser executado antes de ser finalizado. Defini-la muito alta pode levar a aplicações sem resposta, enquanto um valor mais baixo incentiva tempos de resposta mais rápidos e eficientes.

## 6.Pergunta

**Por que o gerenciamento de sessões é importante e o que pode ser feito para otimizá-lo no PHP?**

Resposta:O gerenciamento de sessões é importante para manter o estado do usuário em requisições múltiplas. Usar armazenamentos de dados em memória, como Memcached ou Redis, para dados de sessão pode otimizar o desempenho e melhorar a escalabilidade.

## 7.Pergunta

**Quais vantagens o buffer de saída oferece em aplicações PHP?**

Mais Livros Grátis



Escaneie para baixar



Ouça-o

Resposta:O buffer de saída agrega dados de saída antes de enviá-los ao cliente, reduzindo o número de requisições HTTP feitas. Isso melhora a eficiência geral da rede e os tempos de carregamento das páginas da web.

## 8.Pergunta

**Como o cache realpath contribui para o desempenho das aplicações PHP?**

Resposta:O cache realpath armazena caminhos resolvidos para arquivos incluídos em scripts PHP, evitando a necessidade de procurar caminhos de arquivos repetidamente. Isso pode levar a melhorias de desempenho, especialmente para aplicações que incluem muitos arquivos.

## 9.Pergunta

**Por que é benéfico automatizar a implantação de aplicações PHP?**

Resposta:Automatizar a implantação torna o processo simples, previsível e reversível. Isso reduz o risco de erro durante a implantação e torna mais fácil reverter caso necessário, incentivando os desenvolvedores a implantar



atualizações com mais frequência.

### 10.Pergunta

**Qual é o papel do controle de versão na implantação de aplicações PHP?**

Resposta:O controle de versão, como o Git, permite que os desenvolvedores acompanhem mudanças, revertam versões e gerenciem ramificações para funcionalidades sem afetar a base de código principal. Isso é crucial para automatizar e garantir os processos de implantação.

### 11.Pergunta

**O que é o Capistrano e como ele simplifica a implantação de PHP?**

Resposta:O Capistrano é uma ferramenta de implantação que automatiza o processo de implantação de aplicações PHP em servidores remotos via SSH, tornando as implantações simples, previsíveis e reversíveis. Ele mantém diretórios de aplicação e gerencia versões de forma eficaz.

### 12.Pergunta

**Quais são algumas boas práticas para configurar o php.ini para uso em produção?**

Mais Livros Grátis



Escaneie para baixar



Ouça-o



Resposta:Boas práticas incluem definir limites de memória apropriados, habilitar o OPcache, configurar limites de upload de arquivos, definir um max\_execution\_time baixo para scripts e usar métodos seguros de gerenciamento de sessões.

### 13.Pergunta

**Como você deve responder se sua implantação PHP introduzir um bug?**

Resposta:Você pode simplesmente usar o recurso de rollback da sua ferramenta de implantação (como o Capistrano) para reverter para a última versão estável da sua aplicação, minimizando o tempo de inatividade e o impacto sobre os usuários.

### 14.Pergunta

**Qual é a abordagem recomendada para lidar com tarefas de longa duração no PHP?**

Resposta:Tarefas de longa duração devem ser transferidas para processos de trabalhador separados ou executadas em segundo plano para evitar atrasos na experiência do usuário



na thread principal da aplicação.

### 15.Pergunta

**Qual é o impacto da otimização de recursos do servidor no desempenho do PHP?**

Resposta:Otimizar recursos do servidor, como alocação de memória e minimização do consumo de processos, impacta diretamente o desempenho do PHP, garantindo que a aplicação possa lidar com a carga e o tráfego esperados de forma eficiente.

## Capítulo 12 | 9. Implantação| Perguntas e respostas

### 1.Pergunta

**Qual é o principal objetivo de usar controle de versão para o desenvolvimento de aplicações PHP?**

Resposta:O principal objetivo de usar controle de versão é rastrear mudanças no código, permitindo que os desenvolvedores marquem lançamentos, voltem a estados anteriores e experimentem novas funcionalidades de forma isolada, sem impactar a produção. Isso garante um fluxo de trabalho seguro



e eficiente.

## 2.Pergunta

**Como deve ser o processo de implantação de uma aplicação PHP?**

Resposta:O processo de implantação deve ser simples (idealmente um comando de linha), previsível (sem efeitos colaterais inesperados) e reversível (permitindo retornos rápidos a versões estáveis anteriores em caso de erros).

## 3.Pergunta

**Por que é essencial automatizar a implantação de aplicações?**

Resposta:Automatizar a implantação de aplicações reduz a complexidade e o risco, tornando o processo mais simples e confiável, o que incentiva atualizações frequentes em produção e ajuda a manter a qualidade do software.

## 4.Pergunta

**Quais ferramentas podem ser usadas para automatizar a implantação de aplicações PHP?**

Resposta:O Capistrano é uma ferramenta popular para automatizar implantações, mas alternativas incluem

**Mais Livros Grátis**



Escaneie para baixar



**Ouça-o**

Deployer, Magallanes e Rocketeer.

### 5.Pergunta

**Como o Capistrano simplifica o processo de implantação?**

Resposta:O Capistrano utiliza SSH para se comunicar com servidores remotos, organiza as implantações em diretórios distintos, mantém várias versões para retrocesso e oferece uma interface de comando direta para implantar e reverter mudanças.

### 6.Pergunta

**O que você deve estabelecer antes de implantar uma aplicação com o Capistrano?**

Resposta:Você deve configurar a autenticação usando um par de chaves SSH entre sua máquina local e os servidores remotos, assim como entre os servidores remotos e o repositório Git, permitindo conexões seguras durante a implantação.

### 7.Pergunta

**Qual é o papel do Composer no processo de implantação?**

Resposta:O Composer é um gerenciador de dependências para PHP que garante que todas as bibliotecas e

Mais Livros Grátis



Escaneie para baixar



Ouça-o

dependências necessárias sejam instaladas automaticamente durante a implantação, simplificando o processo.

## 8.Pergunta

**Quais vantagens um processo de implantação reversível oferece?**

Resposta:Um processo de implantação reversível oferece uma rede de segurança, permitindo que os desenvolvedores revertam rapidamente quaisquer erros cometidos durante a implantação, reduzindo a ansiedade associada ao envio de código para a produção.

## 9.Pergunta

**Quando você deve escrever testes no processo de desenvolvimento?**

Resposta:Os testes devem ser escritos antes, durante e após o desenvolvimento para garantir que o comportamento da aplicação e as funcionalidades apropriadas sejam continuamente verificadas ao longo do ciclo de codificação.

## 10.Pergunta

**O que são Testes Unitários e como eles funcionam em aplicações PHP?**

Mais Livros Grátis



Escaneie para baixar



Ouça-o

Resposta: Testes unitários verificam pedaços isolados de código, como classes ou funções, para garantir que se comportem conforme o esperado. O PHPUnit é o framework mais comum usado para escrever e executar testes unitários em PHP.

## 11. Pergunta

**Como o PHPUnit pode ajudar a melhorar a qualidade do código?**

Resposta: O PHPUnit permite que os desenvolvedores escrevam testes que confirmam o comportamento do código, fornece feedback imediato sobre mudanças no código e ajuda a garantir que regressões sejam detectadas precocemente, o que melhora a qualidade geral do código.

Mais Livros Grátis



Escaneie para baixar



Ouça-o



Ad



Escanear para baixar



# Experimente o aplicativo Bookey para ler mais de 1000 resumos dos melhores livros do mundo

Desbloqueie **1000+** títulos, **80+** tópicos

Novos títulos adicionados toda semana

Product & Brand

 Liderança & Colaboração

 Gerenciamento de Tempo

 Relacionamento & Comunicação

 Estratégia de Negócios

 Criatividade

 Memórias

 Conheça a Si Mesmo

 Psicologia

Empreendedorismo

 História Mundial

 Comunicação entre Pais e Filhos

 Autocuidado

 Mente

## Visões dos melhores livros do mundo

amento  
pos

Os 7 Hábitos das  
Pessoas Altamente  
Eficazes



Mini Hábitos



Hábitos Atômicos



O Clube das 5  
da Manhã



Como Fazer Amigos  
e Influenciar  
Pessoas



Com  
Não



Teste gratuito com Bookey



## Capítulo 13 | 10. Testes| Perguntas e respostas

### 1.Pergunta

**Por que os testes são importantes no desenvolvimento de PHP?**

Resposta:Os testes garantem que as aplicações funcionem como esperado, reduzindo medos em relação à implantação e aumentando a confiança na integridade do código. Eles mitigam incertezas que surgem com o lançamento de software, levando a economias de custo e prevenindo dores de cabeça futuras com bugs.

### 2.Pergunta

**Quando os desenvolvedores devem escrever testes durante o processo de desenvolvimento?**

Resposta:Os testes devem ser escritos antes do desenvolvimento, durante o desenvolvimento e após o desenvolvimento. Essa abordagem proativa garante que todas as funcionalidades da aplicação sejam validadas continuamente.



### 3.Pergunta

**Qual a diferença entre testes unitários e testes funcionais?**

Resposta: Testes unitários focam em componentes individuais como classes e métodos para verificar sua correção em isolamento, enquanto testes funcionais validam o comportamento da aplicação em um nível macro para garantir que todos os componentes trabalhem juntos corretamente.

### 4.Pergunta

**Como funciona o Desenvolvimento Orientado a Testes (TDD)?**

Resposta: No TDD, os testes são escritos antes do código real da aplicação. Inicialmente, esses testes falham, pois a funcionalidade correspondente ainda não foi implementada. Depois, os desenvolvedores escrevem apenas o suficiente para que os testes passem, promovendo um processo de desenvolvimento intencional e iterativo.

### 5.Pergunta

**O que é o Desenvolvimento Orientado a Comportamento (BDD) e como é estruturado?**

Mais Livros Grátis



Escaneie para baixar



Ouça-o

Resposta:O BDD envolve a escrita de cenários em uma linguagem que descreve como a aplicação deve se comportar. Inclui o SpecBDD, que foca em detalhes de implementação usando histórias legíveis, e o StoryBDD, que se concentra na lógica de negócios mais ampla sem entrar em especificidades.

## 6.Pergunta

### **Como os testes do PHPUnit devem ser organizados?**

Resposta:Os testes do PHPUnit devem ser agrupados em casos de teste e organizados em suítes de teste. Cada caso de teste, que é uma classe que estende o PHPUnit\_Framework\_TestCase, contém múltiplos testes individuais que afirmam cenários específicos.

## 7.Pergunta

### **Qual o papel da cobertura de código nos testes?**

Resposta:Relatórios de cobertura de código indicam quais partes do código foram executadas durante os testes, ajudando os desenvolvedores a identificar código não testado. É um guia para garantir testes completos, mas não





deve ser um requisito absoluto.

## 8.Pergunta

**Por que os testes contínuos são importantes e como podem ser automatizados?**

Resposta:Os testes contínuos automatizam testes para serem executados contra cada commit de código, garantindo que novas alterações não quebrem funcionalidades existentes.

Serviços como Travis CI integram a execução de testes com sistemas de controle de versão, fornecendo feedback instantâneo sobre a qualidade do código.

## 9.Pergunta

**Quais são algumas ferramentas e frameworks comuns para testar aplicações PHP?**

Resposta:As ferramentas de teste comuns para PHP incluem PHPUnit para testes unitários, PHPSpec para SpecBDD e Behat para testes de StoryBDD, cada uma servindo a metodologias específicas dentro do desenvolvimento de aplicações PHP.

**Capítulo 14 | 11. Profiling| Perguntas e respostas**

Mais Livros Grátis



Escaneie para baixar



Ouça-o

## 1.Pergunta

**O que é o profiling em PHP e por que é importante?**

Resposta:O profiling em PHP é o processo de análise do desempenho da aplicação para identificar gargalos e problemas de desempenho. É importante porque ajuda os desenvolvedores a localizar áreas específicas do código que estão desacelerando a aplicação, permitindo otimizações direcionadas, o que pode melhorar significativamente a experiência do usuário e a eficiência.

## 2.Pergunta

**Quando você deve usar um profiler para suas aplicações em PHP?**

Resposta:Você deve usar um profiler quando houver problemas de desempenho que sejam difíceis de diagnosticar diretamente. Se certas áreas da sua aplicação estão funcionando mais lentamente do que o esperado e você não consegue identificar a causa, o profiling pode ajudar a descobrir problemas de desempenho ocultos.





### 3.Pergunta

**Quais são as ferramentas de benchmarking e como elas ajudam a identificar problemas de desempenho?**

Resposta:As ferramentas de benchmarking, como Apache Bench e Siege, simulam múltiplos usuários acessando sua aplicação para medir métricas de desempenho, como requisições por segundo e tempos de resposta. Após a execução desses testes, elas fornecem insights sobre como sua aplicação lida com carga, ajudando a identificar gargalos potenciais.

### 4.Pergunta

**Quais são alguns exemplos de profilers populares de PHP?**

Resposta:Alguns profilers populares de PHP incluem o Xdebug, que é útil durante o desenvolvimento, mas pode consumir muitos recursos, e o XHPProf, criado pelo Facebook, que pode ser usado tanto em desenvolvimento quanto em produção. New Relic e Blackfire também são profilers notáveis que fornecem insights detalhados sobre o



desempenho.

## 5.Pergunta

### **Qual é a diferença entre Xdebug e XHPProf?**

Resposta:O Xdebug é tipicamente usado durante o desenvolvimento para depuração e profiling, mas pode ser intensivo em recursos e não é recomendado para uso em produção. O XHPProf, por outro lado, foi projetado para ambientes de desenvolvimento e produção, oferecendo uma pegada de desempenho mais leve e está integrado a ferramentas de visualização como o XHGUI.

## 6.Pergunta

### **Como funciona o profiling do Xdebug e quais são suas configurações?**

Resposta:O profiling do Xdebug gera relatórios de desempenho analisando a pilha de chamadas do PHP. As principais configurações incluem habilitar o profiler sob demanda com 'xdebug.profiler\_enable\_trigger' definido como 1 e especificar o diretório de saída para os resultados do profiler com 'xdebug.profiler\_output\_dir'. Isso permite

**Mais Livros Grátis**



Escaneie para baixar



**Ouçá-o**

que você controle quando o profiling está ativo.

## 7.Pergunta

### **O que é o New Relic e como ele difere do Xdebug e do XHPProf?**

Resposta:O New Relic é um serviço web profiler de PHP que fornece insights de desempenho em tempo quase real através de um painel online. Ao contrário do Xdebug e do XHPProf, que funcionam diretamente como extensões do PHP, o New Relic opera através de um daemon proprietário e é um serviço pago, o que o torna diferente em termos de acessibilidade e funcionalidade.

## 8.Pergunta

### **Quais vantagens o XHPProf oferece em relação a profilers tradicionais de PHP?**

Resposta:O XHPProf permite uma análise detalhada de desempenho sem overhead significativo de recursos. Ele fornece métricas de desempenho em tempo real e é projetado para uso tanto em desenvolvimento quanto em produção, permitindo que os desenvolvedores investiguem lentidões em

**Mais Livros Grátis**



Escaneie para baixar



**Ouça-o**

ambientes de produção.

## 9.Pergunta

**Como os desenvolvedores podem implementar melhorias de desempenho com base nos resultados do profiling?**

Resposta:Com os insights obtidos através do profiling, os desenvolvedores podem se concentrar em gargalos específicos, otimizar consultas ao banco de dados, simplificar caminhos de execução do código, reduzir o consumo de recursos e, potencialmente, refatorar o código para um melhor desempenho.

## 10.Pergunta

**Quais desenvolvimentos futuros são esperados para o PHP e seu ecossistema?**

Resposta:O futuro do PHP inclui melhorias contínuas como o PHP 7, que promete melhor desempenho e eficiência de memória. Além disso, projetos como HHVM e Hack estão impulsionando o PHP em direção a uma experiência de desenvolvimento aprimorada, indicando uma evolução vibrante do ecossistema do PHP.

**Mais Livros Grátis**



Escaneie para baixar



**Ouça-o**

## Capítulo 15 | 12. HHVM e Hack| Perguntas e respostas

### 1.Pergunta

**Por que o HHVM foi desenvolvido pelo Facebook?**

Resposta:O HHVM foi desenvolvido para superar o gargalo de desempenho apresentado pelo tradicional Zend Engine, que se tornou uma limitação à medida que a base de usuários do Facebook cresceu exponencialmente. O Facebook queria melhorar o desempenho sem aumentar os custos de infraestrutura, levando à criação do HHVM, que oferece melhor eficiência por meio de sua compilação JIT.

### 2.Pergunta

**Quais são os principais recursos da linguagem Hack?**

Resposta:A Hack introduz tipagem estrita, novas estruturas de dados e verificação de tipos em tempo real que melhoram a estabilidade e previsibilidade do código. Seus recursos são projetados para ajudar a lidar com grandes bases de código, otimizando a eficiência do desenvolvimento no Facebook.

**Mais Livros Grátis**



Escaneie para baixar



**Ouça-o**

### 3.Pergunta

**Como a Hack auxilia na detecção de erros em comparação com o PHP?**

Resposta:A tipagem estática da Hack e o verificador de tipos em tempo real identificam erros mais cedo no processo de desenvolvimento, enquanto a tipagem dinâmica do PHP verifica os tipos em tempo de execução, frequentemente resultando em erros que aparecem mais tarde no ciclo de desenvolvimento.

### 4.Pergunta

**O que torna a Hack diferente do PHP tradicional?**

Resposta:Embora a Hack seja predominantemente compatível com versões anteriores do PHP, ela adiciona recursos como tipos estritos, coleções e um sistema de verificação de tipos mais avançado, o que a torna mais adequada para projetos de grande escala e complexos.

### 5.Pergunta

**Quando os desenvolvedores devem considerar o uso do HHVM?**

Resposta:Os desenvolvedores devem considerar o HHVM se

**Mais Livros Grátis**



Escaneie para baixar



**Ouçá-o**



já otimizarem suas aplicações em PHP e ainda assim precisarem de melhorias adicionais de desempenho. É recomendável abordar primeiro otimizações mais simples, como a redução de requisições HTTP, antes de implementar o HHVM.

## 6.Pergunta

**Qual é a abordagem recomendada para migrar do PHP para a Hack?**

Resposta:Comece mudando a tag de abertura do PHP de '`<?php`' para '`<?hh`' no código. Introduza gradualmente a tipagem estática e os recursos específicos da Hack ao longo do tempo, permitindo uma transição suave.

## 7.Pergunta

**Qual é a importância da especificação da linguagem PHP mencionada no capítulo?**

Resposta:A especificação da linguagem PHP foi introduzida para garantir consistência nas implementações de PHP, incluindo o Zend Engine e o HHVM, beneficiando a comunidade PHP ao fornecer uma base clara para o



desenvolvimento futuro.

## 8.Pergunta

**O HHVM é adequado para todas as aplicações PHP?**

Resposta:Não, o HHVM não é adequado para todas as aplicações. Ele é especialmente valioso para aplicações que já otimizaram outros aspectos de desempenho e precisam de melhorias adicionais. Os desenvolvedores devem avaliar se realmente precisam do HHVM antes de trocar.

## 9.Pergunta

**Quais são as vantagens de usar tipagem estática na Hack?**

Resposta:Usar tipagem estática na Hack leva a uma detecção mais precoce de erros relacionados a tipos, um comportamento do código mais previsível e seguro, e potencialmente menos necessidade de testes unitários extensivos em comparação com linguagens de tipagem dinâmica.

**Mais Livros Grátis**



Escaneie para baixar



**Ouçá-o**



Escanear para baixar



# Por que o Bookey é um aplicativo indispensável para amantes de livros



## Conteúdo de 30min

Quanto mais profunda e clara for a interpretação que fornecemos, melhor será sua compreensão de cada título.



## Clipes de Ideias de 3min

Impulsione seu progresso.



## Questionário

Verifique se você dominou o que acabou de aprender.



## E mais

Várias fontes, Caminhos em andamento, Coleções...

Teste gratuito com Bookey



## Capítulo 16 | 13. Comunidade| Perguntas e respostas

### 1.Pergunta

**Por que a comunidade PHP é considerada um recurso valioso para os desenvolvedores?**

Resposta:A comunidade PHP é diversa, vibrante e global, oferecendo oportunidades de aprendizado e compartilhamento de experiências com outros desenvolvedores PHP. Participar da comunidade ajuda tanto iniciantes quanto especialistas a melhorarem suas habilidades, se manterem informados sobre as melhores práticas e se encontrarem com pessoas de mentalidade semelhante.

### 2.Pergunta

**O que são Grupos de Usuários PHP (PUGs) e como eles podem beneficiar os desenvolvedores?**

Resposta:Os Grupos de Usuários PHP (PUGs) são encontros locais de desenvolvedores PHP que oferecem ótimas oportunidades de networking. Eles permitem que os

Mais Livros Grátis



Escaneie para baixar



Ouça-o

membros se conectem, compartilhem conhecimento, discutam desafios e colaborem em projetos. Participar ou iniciar um PUG pode aprimorar significativamente sua experiência de aprendizado e expandir sua rede profissional.

### 3.Pergunta

**O que você deve fazer se não houver um Grupo de Usuários PHP (PUG) por perto?**

Resposta:Se não houver um PUG local disponível, você pode iniciar seu próprio grupo para reunir entusiastas de PHP da sua região ou se juntar a plataformas online como o NomadPHP, que oferece encontros remotos com palestrantes e discussões sobre diversos tópicos relacionados ao PHP.

### 4.Pergunta

**Como as conferências PHP melhoram o aprendizado e o networking de um desenvolvedor?**

Resposta:As conferências PHP são projetadas para promover networking com líderes do setor e colegas, além de oferecer palestras e workshops que abordam as últimas tendências e tecnologias em PHP. Elas também proporcionam

**Mais Livros Grátis**



Escaneie para baixar



**Ouça-o**



oportunidades de adquirir conhecimento diretamente de especialistas, o que pode influenciar consideravelmente a carreira e a base de conhecimento de um desenvolvedor.

## 5.Pergunta

### **Qual é o papel da mentoria na comunidade PHP?**

Resposta:A mentoria conecta desenvolvedores PHP experientes com iniciantes que precisam de orientação. Mentores experientes podem oferecer conselhos e apoio inestimáveis, ajudando os iniciantes a navegar nas complexidades do desenvolvimento em PHP e incentivando-os a avançar em suas carreiras.

## 6.Pergunta

### **Como os desenvolvedores podem se manter atualizados com os recursos e práticas mais recentes do PHP?**

Resposta:Os desenvolvedores podem se manter informados seguindo diversos recursos, incluindo o site oficial do PHP, listas de discussão e redes sociais como o Twitter. Eles também podem ouvir podcasts relacionados ao PHP, que muitas vezes apresentam discussões sobre novos recursos,

**Mais Livros Grátis**



Escaneie para baixar



**Ouça-o**



práticas e notícias da comunidade.

## 7.Pergunta

**Quais são alguns recursos online recomendados para desenvolvedores PHP?**

Resposta:Recursos importantes incluem sites como php.net para documentação, php-fig.org para padrões e phptherightway.com para melhores práticas. Listas de discussão e contas no Twitter dedicadas ao PHP também fornecem notícias e atualizações contínuas.

## 8.Pergunta

**Qual é a importância da mentoria para iniciantes e desenvolvedores PHP experientes?**

Resposta:Para iniciantes, a mentoria oferece suporte e conselhos que podem aumentar significativamente a confiança e o nível de habilidade, enquanto para desenvolvedores experientes, a mentoria é uma oportunidade de retribuir à comunidade e reforçar seu próprio conhecimento ao ensinar os outros.

**Capítulo 17 | A. Instalando PHP| Perguntas e respostas**

Mais Livros Grátis



Escaneie para baixar



Ouça-o

## 1.Pergunta

**O que são gerenciadores de pacotes e como eles beneficiam a instalação do PHP no Linux?**

Resposta:Gerenciadores de pacotes, como o aptitude no Ubuntu e o yum no CentOS, simplificam o processo de encontrar, instalar, atualizar e remover software. Eles oferecem uma maneira prática de gerenciar pacotes de software, garantindo que as dependências sejam resolvidas automaticamente. Isso é especialmente útil para instalar o PHP, pois pode gerenciar rapidamente todos os componentes e extensões necessários.

## 2.Pergunta

**Por que é importante verificar a versão do PHP instalada via gerenciador de pacotes?**

Resposta:É crucial verificar a versão do PHP instalada porque os gerenciadores de pacotes às vezes oferecem versões desatualizadas. Por exemplo, o Ubuntu 14.04 LTS pode instalar o PHP 5.5.9, enquanto versões mais novas,



como o PHP 5.6.3, podem estar disponíveis. Verificar a versão garante que você esteja usando a iteração mais segura e rica em recursos do PHP.

### 3.Pergunta

**Qual é a importância de adicionar repositórios de terceiros para a instalação do PHP?**

Resposta: Adicionar repositórios de terceiros, como o

`Personal Package Archive (PPA)` de Ondrej que os usuários acessem versões mais recentes e mantidas pela comunidade do PHP. Isso é essencial para desenvolvedores que precisam das últimas características e atualizações de segurança que não estão disponíveis nos repositórios padrão.

### 4.Pergunta

**Quais são os passos para instalar o PHP no Ubuntu usando um PPA?**

Resposta: 1. Instale o binário `add-apt-repository`: ``sudo apt-get install python-software-properties``. 2. Adicione o PPA: ``sudo add-apt-repository ppa:ondrej/php5-5.6``. 3.



Atualize a lista de pacotes: ``sudo apt-get update``. 4. Instale o PHP CLI: ``sudo apt-get install php5-cli``. 5. Instale as extensões necessárias: ``sudo apt-get install php5-curl php5-gd php5-json php5-mcrypt php5-mysqlnd``. 6. Verifique a instalação: ``php -v``.

## 5.Pergunta

**Quais passos devem ser seguidos para instalar o PHP no CentOS?**

Resposta: 1. Adicione o repositório EPEL: ``sudo rpm -Uvh <URL>`` para epel-release. 2. Ative os repositórios de terceiros: ``sudo rpm -Uvh <URL>`` para remi-release. 3. Instale o PHP CLI: ``sudo yum -y --enablerepo=epel,remi,remi-php56 install php-cli``. 4. Instale as extensões: ``sudo yum -y --enablerepo=epel,remi,remi-php56 install php-gd php-mbstring php-mcrypt php-mysqlnd php-opcache php-pdo``. 5. Verifique a instalação do PHP: ``php -v``.

## 6.Pergunta

**Quais são as duas maneiras recomendadas para instalar o PHP no OS X?**

Mais Livros Grátis



Escaneie para baixar



Ouça-o

Resposta:As duas maneiras recomendadas para instalar o PHP no macOS são usando o MAMP (um aplicativo baseado em GUI que agrupa Apache, MySQL e PHP) ou o Homebrew (um gerenciador de pacotes de linha de comando que permite controle total sobre o software instalado). O MAMP é adequado para aqueles que preferem uma interface gráfica, enquanto o Homebrew é recomendado para usuários confortáveis com a linha de comando.

## 7.Pergunta

**Qual é o papel das Ferramentas de Linha de Comando do Xcode no processo de instalação do Homebrew no macOS?**

Resposta:As Ferramentas de Linha de Comando do Xcode fornecem componentes essenciais, como o compilador gcc, que são necessários para o Homebrew construir e instalar pacotes de software. Instalar essas ferramentas é um pré-requisito para configurar o Homebrew no macOS.

## 8.Pergunta

**Por que um desenvolvedor deve considerar compilar o PHP a partir do código-fonte, e quais são os passos**

Mais Livros Grátis



Escaneie para baixar



Ouça-o

## **básicos envolvidos?**

Resposta: Compilar o PHP a partir do código-fonte permite que os desenvolvedores personalizem a instalação com opções específicas adequadas às suas aplicações, como habilitar certas extensões ou otimizações. Os passos básicos são: 1. Baixar o código-fonte do PHP. 2. Extrair. 3. Instalar as dependências necessárias. 4. Configurar as opções de compilação usando `./configure`. 5. Compilar o software usando ``make``. 6. Instalar usando ``make install``.

## **9.Pergunta**

**Quais as vantagens que o Homebrew oferece em relação ao MAMP para a instalação do PHP?**

Resposta: O Homebrew oferece mais flexibilidade e personalização para gerenciar pacotes instalados, permitindo que os desenvolvedores instalem múltiplas versões do PHP e extensões associadas, e as configurem de acordo com suas necessidades. É uma ferramenta de linha de comando, o que pode atrair aqueles que preferem trabalhar com comandos de terminal em vez de uma interface gráfica.





## Capítulo 18 | B. Ambientes de Desenvolvimento Local| Perguntas e respostas

### 1.Pergunta

**Por que é recomendado evitar o uso da pilha de software padrão fornecida pelo sistema operacional para desenvolvimento local em PHP?**

Resposta: Usar a pilha de software padrão pode levar a problemas de compatibilidade e instabilidade, especialmente quando as atualizações do sistema operacional podem sobrescrever configurações personalizadas. É melhor criar um ambiente de desenvolvimento local em uma máquina virtual isolada do seu sistema operacional principal para evitar esses problemas.

### 2.Pergunta

**Qual é a importância de fazer coincidir o sistema operacional do seu ambiente de desenvolvimento local com o do servidor de produção?**

Resposta: Fazer coincidir o sistema operacional garante que os ambientes operem de maneira consistente, o que minimiza



o risco de erros de implementação e em tempo de execução que podem ocorrer devido a discrepâncias nas versões de software e nas configurações.

### 3.Pergunta

**Como o VirtualBox e o Vagrant simplificam a criação de máquinas virtuais para desenvolvimento em PHP?**

Resposta:O VirtualBox fornece a plataforma para criar e gerenciar máquinas virtuais, enquanto o Vagrant oferece uma interface de linha de comando amigável para configurar, iniciar e desmontar essas VMs facilmente, permitindo que os desenvolvedores se concentrem mais na codificação do que na configuração.

### 4.Pergunta

**O que são boxes do Vagrant e como são usadas no desenvolvimento em PHP?**

Resposta:As boxes do Vagrant são máquinas virtuais pré-configuradas que servem como ponto de partida para ambientes de desenvolvimento. Os desenvolvedores podem selecionar uma box com base nas necessidades do seu projeto

Mais Livros Grátis



Escaneie para baixar



Ouça-o

e, em seguida, provisioná-la com a pilha de software necessária, simplificando significativamente o processo de configuração.

## 5.Pergunta

**Por que o provisionamento é importante para uma máquina virtual do Vagrant e quais opções estão tipicamente disponíveis para esse processo?**

Resposta:O provisionamento é crucial porque configura a máquina virtual com o software necessário para executar a aplicação PHP, como servidores web ou bancos de dados.

Opções comuns para provisionamento incluem ferramentas como Puppet e Chef, que gerenciam a instalação e configuração do software.

## 6.Pergunta

**Como o Laravel Homestead simplifica a configuração do ambiente de desenvolvimento local para aplicações PHP?**

Resposta:O Laravel Homestead é uma box do Vagrant pré-configurada que inclui uma pilha de software completa adaptada para o desenvolvimento em PHP, facilitando a criação de um ambiente de desenvolvimento sem precisar

Mais Livros Grátis



Escaneie para baixar



Ouça-o

instalar individualmente cada componente.

## 7.Pergunta

**Qual é uma dica prática para gerenciar comandos do Vagrant para aumentar a produtividade?**

Resposta:Criar aliases de linha de comando para comandos do Vagrant frequentemente usados pode agilizar o fluxo de trabalho, reduzindo a necessidade de digitar repetidamente comandos longos.

## 8.Pergunta

**Quais são os benefícios de usar pastas sincronizadas entre a máquina local e a máquina virtual no desenvolvimento em PHP?**

Resposta:Usar pastas sincronizadas permite que os desenvolvedores façam alterações em sua máquina local que são imediatamente refletidas na máquina virtual. Isso melhora a eficiência do fluxo de trabalho, pois não há necessidade de transferências manuais de arquivos.

## 9.Pergunta

**Quais ferramentas um novato no Vagrant pode usar para ajudar a configurar seu ambiente de desenvolvimento**

Mais Livros Grátis



Escaneie para baixar



Ouça-o

**sem escrever scripts de provisionamento complexos?**

Resposta: Ferramentas como o PuPHPet fornecem uma interface gráfica para configurar facilmente scripts do Puppet sem precisar de um conhecimento extenso de scripting, enquanto o Vaprobash permite uma configuração simples através de ajustes em arquivos.

## **10.Pergunta**

**Quais estratégias os desenvolvedores devem seguir para começar a utilizar o Vagrant e máquinas virtuais de forma eficaz?**

Resposta: Os desenvolvedores devem começar entendendo as necessidades de seu projeto, selecionando a box do Vagrant apropriada, configurando o Vagrantfile, provisionando com os componentes necessários e aproveitando ferramentas como o Laravel Homestead para configurações simplificadas.

**Mais Livros Grátis**



Escaneie para baixar



**Ouça-o**



Ad



Escanear para baixar



App Store  
Escolha dos Editores



22k avaliações de 5 estrelas

## Feedback Positivo

Afonso Silva

...cada resumo de livro não só  
..., mas também tornam o  
...divertido e envolvente. O  
...tizou a leitura para mim.

**Fantástico!**



Estou maravilhado com a variedade de livros e idiomas  
que o Bookey suporta. Não é apenas um aplicativo, é  
um portal para o conhecimento global. Além disso,  
ganhar pontos para caridade é um grande bônus!

Brígida Santos

F



O  
só  
o  
O

na Oliveira

...correr as  
...ém me dá  
...omprar a  
...ar!

**Adoro!**



Usar o Bookey ajudou-me a cultivar um hábito de  
leitura sem sobrecarregar minha agenda. O design do  
aplicativo e suas funcionalidades são amigáveis,  
tornando o crescimento intelectual acessível a todos.

Duarte Costa

**Economiza tempo!**



O Bookey é o meu apli  
crescimento intelectual  
perspicazes e lindame  
um mundo de conheci

**Aplicativo incrível!**



Eu amo audiolivros, mas nem sempre tenho tempo para  
ouvir o livro inteiro! O Bookey permite-me obter um resumo  
dos destaques do livro que me interessa!!! Que ótimo  
conceito!!! Altamente recomendado!

Estevão Pereira

**Aplicativo lindo**



Este aplicativo é um salva-vidas para  
de livros com agendas lotadas. Os re  
precisos, e os mapas mentais ajudar  
o que aprendi. Altamente recomend

Teste gratuito com Bookey





# PHP Moderno Quiz e teste

Ver a resposta correta no site do Bookey

## Capítulo 1 | I. Recursos da Linguagem| Quiz e teste

1.O PHP foi criado originalmente por Rasmus

Lerdorf em meados da década de 1990 e era conhecido como Ferramentas de Página Inicial Pessoal.

2.O PHP 3 introduziu melhorias significativas e se tornou amplamente bem-sucedido, atingindo uma participação de mercado de 50% dos servidores web até o final de 1998.

3.As práticas de desenvolvimento de PHP Moderno enfatizam o uso do Composer para gerenciamento de dependências e PSRs para padrões de código.

## Capítulo 2 | 1. O Novo PHP| Quiz e teste

1.Os namespaces foram introduzidos no PHP 5.3.0 para ajudar a organizar o código e evitar colisões de nomes.

2.O PHP permite a importação e o uso de alias para namespaces, mas a sintaxe não exige o uso da

**Mais Livros Grátis**



Escaneie para baixar



**Ouça-o**

palavra-chave 'use'.

3.Os geradores no PHP fornecem uma iteração eficiente em termos de memória, gerando valores sob demanda em vez de armazená-los em arrays.

## **Capítulo 3 | 2. Recursos| Quiz e teste**

- 1.Namespaces foram introduzidos no PHP 5.3.0 e ajudam a evitar conflitos de nomes de classes em grandes projetos.
- 2.Traits, introduzidos no PHP 5.4, exigem que classes compartilhem um ancestral comum para reutilizar código.
- 3.Geradores, disponíveis a partir do PHP 5.5, permitem a criação de iteradores que pré-alocam grandes quantidades de memória para retornar valores.

**Mais Livros Grátis**



Escaneie para baixar



**Ouça-o**



Baixe o app Bookey para desfrutar

# Mais de 1000 resumos de livros com quizzes

**Teste grátis disponível!**

Escanear para baixar



## Capítulo 4 | II. Boas Práticas| Quiz e teste

- 1.O Grupo de Interoperação de Frameworks PHP (PHP-FIG) foi estabelecido em 2009 para melhorar a interoperabilidade entre frameworks PHP.
- 2.As diretrizes do PSR-2 sugerem o uso de tabs para indentação no código PHP.
- 3.O PSR-4 fornece recomendações para convenções de nomenclatura consistentes no código PHP.

## Capítulo 5 | 3. Padrões| Quiz e teste

- 1.Os frameworks PHP são principalmente categorizados em frameworks macro, frameworks micro e frameworks legados.
- 2.O PHP Framework Interop Group (PHP-FIG) foi estabelecido em 2009 para desenvolver novas linguagens de programação para PHP.
- 3.O Composer é usado para gerenciar componentes PHP e suas dependências, facilitando a integração para os desenvolvedores.



## Capítulo 6 | 4. Componentes| Quiz e teste

- 1.PHP Moderno recomenda o uso exclusivo de frameworks monolíticos para construir aplicações.
- 2.Os componentes devem ser focados, minimalistas, cooperativos, bem testados e bem documentados de acordo com o PHP Moderno.
- 3.Packagist é o diretório principal para pacotes PHP e ajuda os desenvolvedores a encontrar e avaliar componentes.

**Mais Livros Grátis**



Escaneie para baixar



**Ouça-o**



Baixe o app Bookey para desfrutar

# Mais de 1000 resumos de livros com quizzes

**Teste grátis disponível!**

Escanear para baixar





## Capítulo 7 | 5. Boas Práticas| Quiz e teste

1. Sanitizar a entrada é importante para proteger sua aplicação PHP de caracteres inseguros.
2. É uma boa prática enviar senhas em texto simples para os usuários por e-mail para recuperação de senha.
3. Usar PDO com declarações preparadas ajuda a prevenir injeções de SQL em aplicações PHP.

## Capítulo 8 | III. Implantação, Testes e Ajustes| Quiz e teste

1. Servidores compartilhados são a opção mais amigável para desenvolvedores ao hospedar aplicações PHP.
2. Um Servidor Privado Virtual (VPS) oferece acesso root via SSH e permite a instalação de qualquer software necessário.
3. Plataforma como Serviço (PaaS) é a melhor opção para desenvolvedores que preferem máxima gestão e controle sobre seus ambientes de hospedagem.

## Capítulo 9 | 6. Hospedagem| Quiz e teste

Mais Livros Grátis



Escaneie para baixar



Ouça-o

1. Servidores compartilhados são a opção de hospedagem mais cara para aplicações PHP.
2. Plataformas como Serviço (PaaS) fornecem gerenciamento de servidores para desenvolvedores.
3. Servidores Privados Virtuais (VPS) fornecem acesso root SSH e permitem a personalização de software.

**Mais Livros Grátis**



Escaneie para baixar



Ouça-o



Baixe o app Bookey para desfrutar

# Mais de 1000 resumos de livros com quizzes

**Teste grátis disponível!**

Escanear para baixar



## Capítulo 10 | 7. Provisionamento| Quiz e teste

- 1.O provisionamento de um servidor para uma aplicação PHP pode ser feito de maneira eficaz sem usar qualquer serviço configurado manualmente pelo usuário.
- 2.A autenticação por chave SSH é recomendada em vez da autenticação por senha para proteger o servidor.
- 3.Desativar a autenticação por senha é desnecessário para garantir o acesso ao servidor durante a implementação da aplicação PHP.

## Capítulo 11 | 8. Ajustando| Quiz e teste

- 1.A configuração ``memory_limit`` no arquivo ``php.ini`` tem um padrão de 256M.
- 2.Habilitar o buffering de saída no PHP pode ajudar a enviar dados de forma mais eficiente.
- 3.Zend OPcache é um recurso que está desativado por padrão nas instalações do PHP.

## Capítulo 12 | 9. Implantação| Quiz e teste

- 1.O processo de implantação em aplicações PHP



Moderno deve ser complicado e exigir comandos manuais.

2.Capistrano é uma ferramenta de implantação projetada principalmente para aplicações PHP.

3.Usar controle de versão é fundamental para rastrear alterações e facilitar a capacidade de reversão em implantações PHP.

**Mais Livros Grátis**



Escaneie para baixar



**Ouça-o**



Baixe o app Bookey para desfrutar

# Mais de 1000 resumos de livros com quizzes

**Teste grátis disponível!**

Escanear para baixar





## Capítulo 13 | 10. Testes| Quiz e teste

1. Testes são um componente desnecessário no desenvolvimento de aplicações PHP.
2. Testes unitários verificam a funcionalidade de classes individuais de forma isolada.
3. Cobertura de código de 100% é necessária para um resultado de teste bem-sucedido.

## Capítulo 14 | 11. Profiling| Quiz e teste

1. A análise de desempenho só é útil durante a fase de desenvolvimento e não deve ser usada em ambientes de produção.
2. Xdebug é um conhecido analisador de desempenho para desenvolvimento que gera saídas legíveis por humanos.
3. Você deve usar um analisador de desempenho apenas quando problemas de desempenho não forem evidentes com outras ferramentas de benchmarking.

## Capítulo 15 | 12. HHVM e Hack| Quiz e teste

1. HHVM foi lançado pela primeira vez pelo Facebook em 2013 para melhorar o desempenho



das aplicações PHP.

2.Hack é uma linguagem de programação completamente separada que não mantém nenhuma compatibilidade com PHP.

3.A tipagem estática, introduzida pelo Hack, permite a verificação de tipos em tempo real, melhorando a correção do código.

**Mais Livros Grátis**



Escaneie para baixar



Ouça-o



Baixe o app Bookey para desfrutar

# Mais de 1000 resumos de livros com quizzes

**Teste grátis disponível!**

Escanear para baixar



## Capítulo 16 | 13. Comunidade| Quiz e teste

1. Ingressar em um Grupo Local de Usuários de PHP (PUG) pode aprimorar suas habilidades e sua rede de contatos como desenvolvedor.
2. É desnecessário participar de conferências de PHP para se manter atualizado sobre novas funcionalidades da linguagem.
3. Procure ou torne-se um mentor para aprimorar suas habilidades e contribuir para a comunidade PHP.

## Capítulo 17 | A. Instalando PHP| Quiz e teste

1. As distribuições Linux usam o mesmo gerenciador de pacotes para instalar o PHP.
2. O MAMP é um método recomendado para instalar o PHP no OS X.
3. Não é necessário baixar o código-fonte ao construir o PHP a partir do código-fonte.

## Capítulo 18 | B. Ambientes de Desenvolvimento Local| Quiz e teste

1. Desenvolver aplicações PHP em um computador



local é desencorajado porque pode levar a possíveis sobrescritas durante as atualizações do sistema operacional.

2.O Vagrant simplifica o processo de gerenciamento de VMs do VirtualBox através de uma interface de linha de comando.

3.As boxes do Vagrant não são ambientes pré-configurados para construir aplicações e devem ser montadas do zero.

**Mais Livros Grátis**



Escaneie para baixar



**Ouça-o**



Baixe o app Bookey para desfrutar

# Mais de 1000 resumos de livros com quizzes

**Teste grátis disponível!**

Escanear para baixar

