

# **Desenvolvimento de um Sistema Web para Compartilhamento de Receitas Utilizando o Framework Yii com PHP**

**Thiago Scaravonatto**

**Thiago Venturini**

**Thiago Thomas**

## **Resumo**

O presente artigo descreve o desenvolvimento de um sistema web denominado "**Recipes**", voltado para o cadastro, gerenciamento e compartilhamento de receitas culinárias. A aplicação foi construída utilizando a linguagem de programação **PHP** em conjunto com o **framework Yii**, adotando o padrão arquitetural **MVC (Model-View-Controller)** para garantir organização, modularidade e reutilização de código. O projeto buscou aliar funcionalidade e usabilidade, permitindo ao usuário interações como o cadastro, exclusão e edição de receitas e cadastro de usuário. Para o armazenamento dos dados, utilizou-se o **banco de dados MySQL**, integrando-se de forma eficiente ao framework. Os resultados obtidos demonstram a viabilidade da solução proposta, evidenciando o Yii como uma ferramenta robusta para o desenvolvimento ágil de aplicações web.

**Palavras-chave:** YII.PHP.RECIPES.SISTEMA.WEB

## **Introdução**

Com o avanço da tecnologia e o crescente uso da internet como meio de acesso à informação, sistemas web tornaram-se ferramentas essenciais para o compartilhamento de conteúdos diversos. Entre os muitos temas de interesse, a culinária se destaca como uma área de forte apelo popular, com ampla demanda por plataformas que permitam o acesso e a organização de receitas de forma prática e intuitiva.

Diante desse cenário, este trabalho apresenta o desenvolvimento do sistema web "**Recipes**", uma aplicação voltada ao gerenciamento e compartilhamento de receitas culinárias. O sistema foi construído utilizando a linguagem de programação **PHP** com o

**framework Yii**, que adota o padrão arquitetural **MVC**, promovendo organização do código, separação de responsabilidades e manutenção facilitada.

A escolha do Yii se deu por sua estrutura leve e poderosa, além de oferecer suporte nativo ao padrão **ORM (Object-Relational Mapping)** via **ActiveRecord**, facilitando o acesso e manipulação de dados de forma orientada a objetos.

A plataforma foi projetada com foco na experiência do usuário, oferecendo funcionalidades como:

- **Cadastro e login de usuários**
- **Cadastro, edição, exclusão e visualização de receitas**
- **Favoritar receitas**
- **Consulta detalhada de cada receita com ingredientes e modo de preparo**

Os dados são armazenados em um banco de dados relacional **MySQL**, com modelagem adequada à lógica do sistema. O projeto buscou aplicar boas práticas de desenvolvimento web, além de promover a aplicação prática dos conceitos aprendidos em ambiente acadêmico.

Este artigo está organizado da seguinte forma: na seção de **Fundamentação Teórica**, são discutidos os conceitos e tecnologias utilizadas; a seção de **Desenvolvimento** detalha o processo de implementação do sistema e a modelagem do banco; em seguida, os **Resultados** obtidos são analisados, incluindo os principais desafios enfrentados.

## **2. Fundamentação Teórica**

### **2.1 Desenvolvimento Web**

O desenvolvimento web consiste na criação de aplicações acessíveis por meio de navegadores, utilizando linguagens e tecnologias específicas para o lado do cliente (frontend) e do servidor (backend). No backend, linguagens como PHP, Python e Java são amplamente utilizadas para manipulação de dados, autenticação de usuários, controle de permissões e lógica de negócios. Já no frontend, tecnologias como HTML, CSS e JavaScript são responsáveis pela apresentação visual e interação com o usuário.

A arquitetura de aplicações web modernas frequentemente adota o padrão **MVC (Model-View-Controller)**, que organiza o código em três camadas distintas, facilitando a manutenção, o reaproveitamento e a escalabilidade da aplicação.

## 2.2 Linguagem PHP

Criada originalmente por Rasmus Lerdorf em 1994, a linguagem **PHP (Hypertext Preprocessor)** é uma das mais populares no desenvolvimento de aplicações web dinâmicas. É uma linguagem de script de código aberto executada no lado do servidor, amplamente suportada por provedores de hospedagem e com vasta documentação.

Entre suas características destacam-se a integração facilitada com bancos de dados relacionais, compatibilidade com servidores web como Apache e Nginx, além do suporte a múltiplos frameworks e bibliotecas que aceleram o desenvolvimento.

## 2.3 Framework Yii e sua Escolha

O **Yii** (acrônimo para *Yes, it is!*) é um framework de desenvolvimento web baseado em PHP, voltado para aplicações de alta performance. Ele segue o padrão arquitetural **MVC** e oferece um conjunto robusto de ferramentas integradas para desenvolvimento rápido, como:

- **ActiveRecord**, sistema nativo de ORM (Object-Relational Mapping);
- **Gii**, gerador de código automático;
- Sistema de autenticação e controle de acesso;
- Suporte a **componentes reutilizáveis** e injeção de dependência;
- Documentação extensa e comunidade ativa.

A escolha do Yii para o desenvolvimento do sistema "*Recipes*" justifica-se por sua leveza, desempenho, segurança e por facilitar a aplicação de boas práticas de programação. Além disso, o Yii é ideal para projetos acadêmicos por promover uma estrutura organizada e orientada a objetos, permitindo o aprendizado claro de conceitos fundamentais como rotas, modelos, sessões, formulários e integração com banco de dados.

## 2.4 Banco de Dados MySQL

O **MySQL** é um sistema de gerenciamento de banco de dados relacional (SGBDR) amplamente utilizado em aplicações web. Ele organiza os dados em tabelas relacionais e oferece suporte a linguagens de consulta como SQL, permitindo operações como inserção, seleção, atualização e exclusão de dados.

Para aplicações como o "*Recipes*", o MySQL fornece desempenho adequado, integridade referencial entre tabelas (por meio de chaves estrangeiras) e compatibilidade direta com PHP e o Yii Framework. A modelagem do banco de dados com base em entidades como usuários, receitas e favoritos foi fundamental para garantir a coerência e integridade dos dados.

## 2.5 Padrão Arquitetural MVC

O padrão **Model-View-Controller (MVC)** é uma abordagem que separa a aplicação em três componentes principais:

- **Model (Modelo):** representa os dados e a lógica de negócio da aplicação.
- **View (Visão):** define a interface com o usuário, ou seja, como os dados são apresentados.
- **Controller (Controlador):** gerencia a interação entre o usuário, os modelos e as views, controlando o fluxo de dados.

Essa separação facilita o desenvolvimento colaborativo, a manutenção do código e a escalabilidade da aplicação. No Yii, o uso do MVC é nativo e fortemente incentivado, permitindo que cada camada seja desenvolvida de forma modular e organizada.

## 2.6 Mapeamento Objeto-Relacional (ORM)

O **ORM (Object-Relational Mapping)** é uma técnica que permite a manipulação de dados armazenados em bancos de dados relacionais por meio de objetos da linguagem de programação. Com isso, reduz-se a necessidade de escrever instruções SQL diretamente, tornando o código mais seguro, legível e manutenível.

No Yii, o ORM é implementado por meio da classe **ActiveRecord**, que associa cada tabela a uma classe e cada linha a um objeto. Isso permite que operações como buscas, deleções, inserções e atualizações sejam realizadas diretamente com objetos, como no exemplo utilizado no “Recipes”:

```
public function actionDelete($id): Response
{
    $model = $this->findModel(id: $id);

    if ($model->user_id !== Yii::$app->user->id) {
        throw new NotFoundHttpException(message: 'Você não tem permissão para deletar esta receita.');
```

Essa abordagem contribui para uma abstração eficiente da camada de dados e previne vulnerabilidades comuns como **SQL Injection**.

### 3. Desenvolvimento

#### 3.1 Justificativa Tecnológica

A escolha das tecnologias utilizadas no desenvolvimento do sistema “*Recipes*” teve como critério principal a integração entre facilidade de uso, robustez e aderência aos princípios de desenvolvimento web moderno. A linguagem **PHP** foi selecionada por ser amplamente utilizada no desenvolvimento de aplicações web e por sua compatibilidade com diversos servidores e bancos de dados.

O **framework Yii** foi escolhido pela sua estrutura modular, suporte nativo ao padrão arquitetural **MVC (Model-View-Controller)** e por oferecer o recurso **ActiveRecord**, que facilita a interação com o banco de dados utilizando o paradigma orientado a objetos. Além disso, o Yii fornece uma ferramenta chamada **Gii**, que automatiza a geração de código para formulários, modelos, controladores e views, otimizando significativamente o tempo de desenvolvimento e reduzindo erros humanos.

### 3.2 Modelagem do Banco de Dados

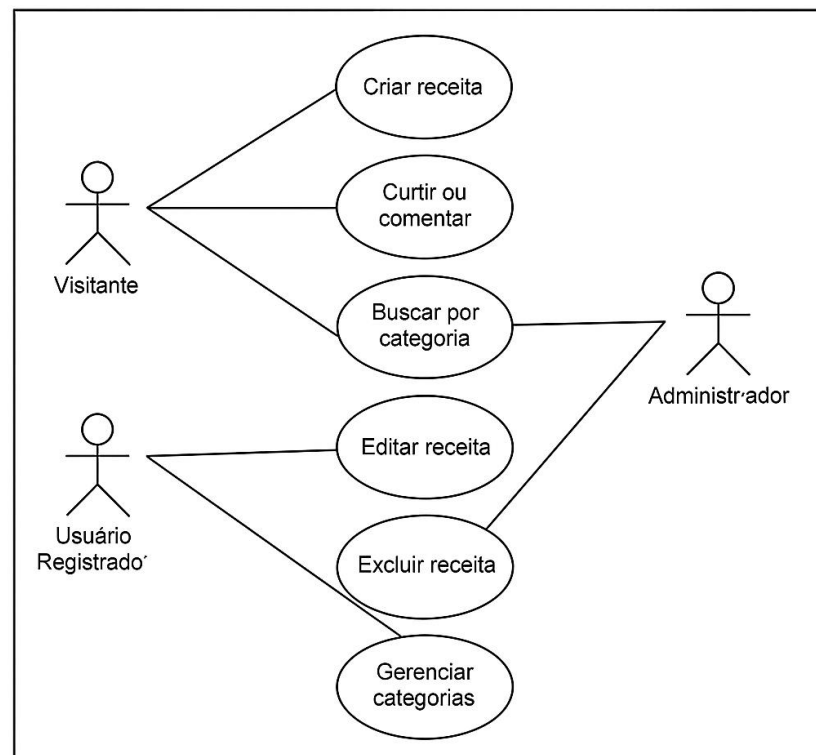
A modelagem do banco de dados foi elaborada com o objetivo de refletir as entidades principais do sistema e seus relacionamentos. Foram definidas as seguintes tabelas principais:

- **user**: armazena dados dos usuários cadastrados (id, username, password\_hash, auth\_key).
- **recipe**: armazena as receitas criadas (id, user\_id, title, slug, description, cook\_time, image, created\_at, updated\_at).
- **category**: registra as categorias de receitas (id, name, slug, created\_at, updated\_at).
- **login**: para armazenamento de logins (id, user\_id, login\_time).
- **recipe\_category**: faz a associação entre recipe e category (recipe\_id, category\_id)

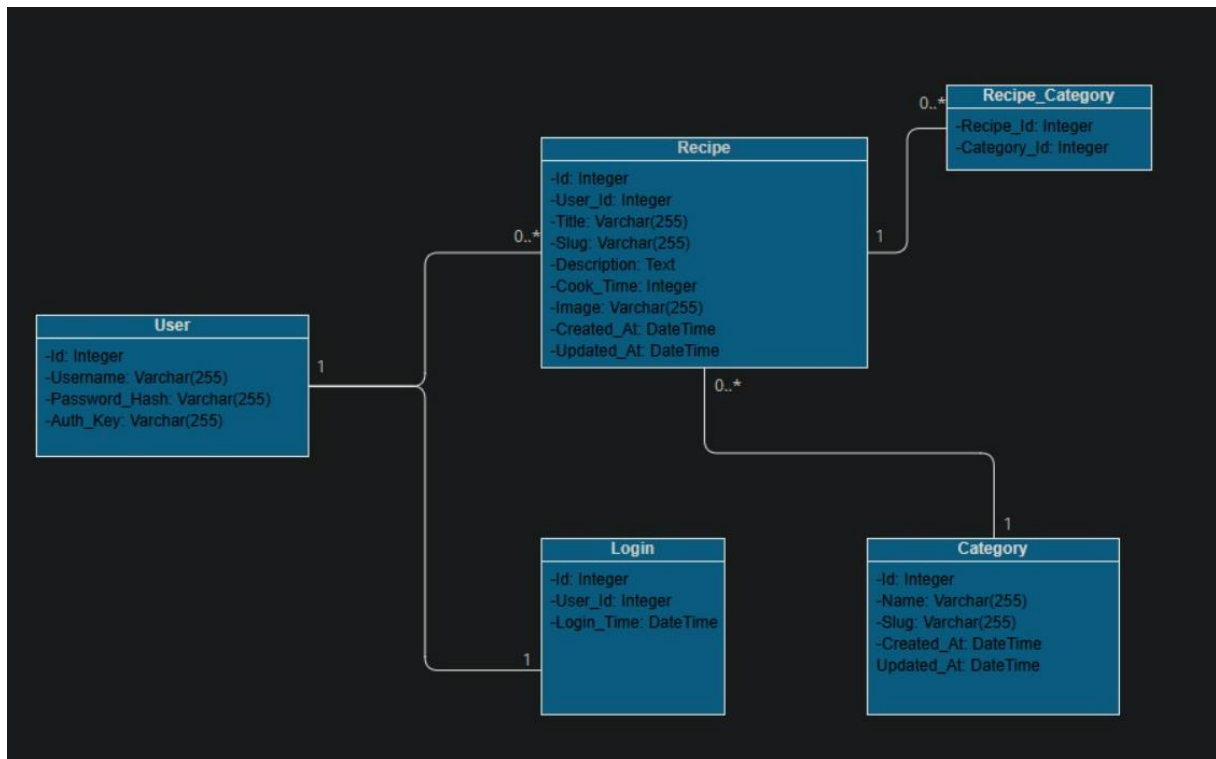
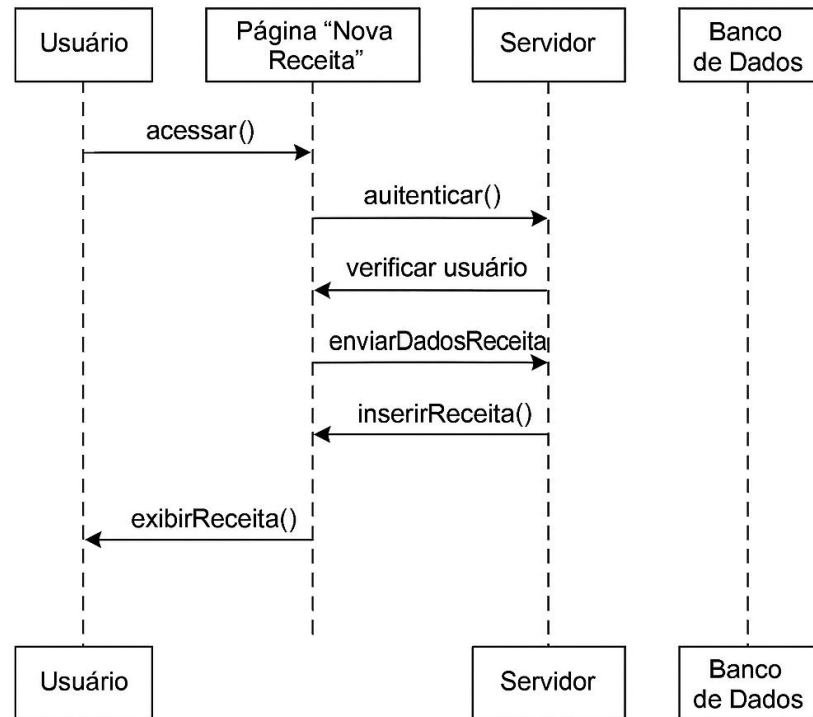
O relacionamento entre usuários e receitas é do tipo **um para muitos**, já que cada usuário pode cadastrar múltiplas receitas.

Um diagrama entidade-relacionamento (ER) pode ser apresentado como suporte visual, evidenciando as chaves primárias e estrangeiras e os relacionamentos existentes.

#### Diagrama de Casos de Uso



## Diagrama de Sequência



### 3.3 Integração com o Banco via ActiveRecord (ORM)

O Yii adota o padrão **ORM (Object-Relational Mapping)** por meio da classe **ActiveRecord**, que permite que cada tabela do banco de dados seja representada por uma classe em PHP. Essa abordagem facilita a manipulação dos dados sem a necessidade de escrever instruções SQL diretamente.

Por exemplo, para buscar uma receita por id, utiliza-se o relacionamento definido no modelo:

```
protected function findModel($id): ?Recipe
{
    if (($model = Recipe::findOne(condition: ['id' => $id])) !== null) {
        return $model;
    }

    throw new NotFoundException(message: 'The requested page does not exist.');
```

Essa integração permite ganho de produtividade, segurança contra SQL injection, e clareza na leitura e manutenção do código.

### 3.4 Funcionalidades Implementadas

O sistema "*Recipes*" oferece as seguintes funcionalidades principais do lado do usuário:

- **Cadastro de Usuário:** formulário com validação de campos e armazenamento seguro da senha (com hash).
- **Login de Usuário:** autenticação utilizando componentes nativos do Yii e sessões de usuário.
- **Cadastro de Receita:** formulário para inserção de título, modo de preparo e imagem ilustrativa.
- **Edição e Exclusão de Receitas:** somente o autor da receita pode realizar alterações ou removê-la.
- **Favoritar Receita:** o usuário pode marcar receitas como favoritas, criando um vínculo de acesso rápido.
- **Visualização Detalhada:** interface para exibir os detalhes completos de uma receita, com estrutura visual limpa.



As views foram construídas com HTML, CSS e o sistema de templates do Yii, garantindo uma boa experiência de navegação.

### 3.5 Etapas do Desenvolvimento

O desenvolvimento do sistema seguiu uma abordagem incremental, com as seguintes fases:

1. **Levantamento de Requisitos:** definição das funcionalidades mínimas viáveis.
2. **Modelagem do Banco de Dados:** estruturação lógica e física do banco em MySQL.
3. **Configuração do Ambiente Yii:** instalação via Composer, criação do projeto base e configuração de conexão com banco de dados.
4. **Implementação das Funcionalidades:** desenvolvimento orientado a modelos e controladores.
5. **Testes e Ajustes:** testes manuais de fluxo, validação de formulários, autenticação e permissões.

## 4 Uso de Padrões de Projeto

### 4.1 Model-View-Controller (MVC)

O padrão MVC organiza o código em três camadas: Model, View e Controller. Os Models cuidam dos dados e regras de negócio, as Views exibem a interface ao usuário e os Controllers recebem requisições e comandam o fluxo. Essa separação facilita manutenção e colaboração, pois cada parte fica isolada e com responsabilidade única.

### 4.2 Active Record

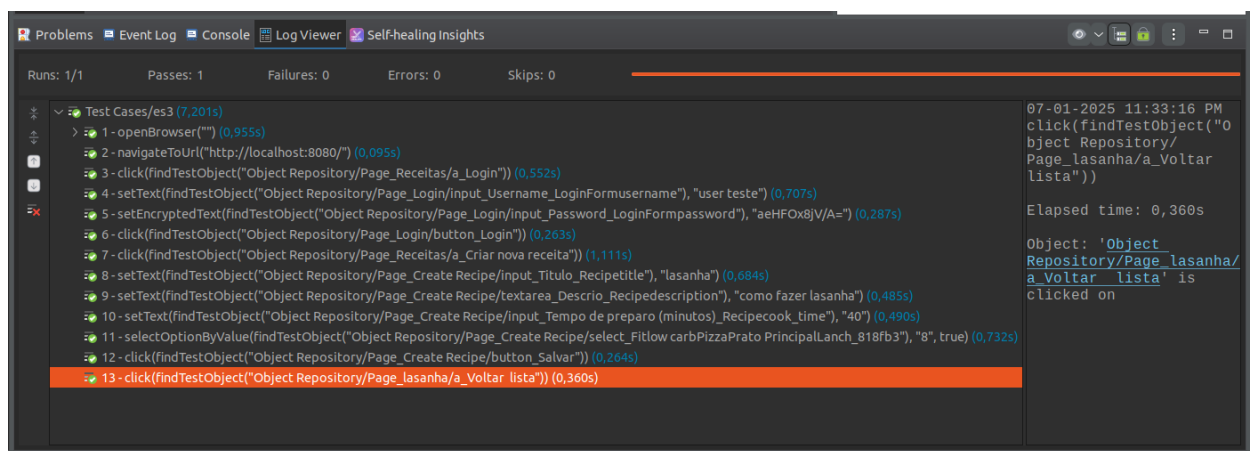
Cada classe de modelo no Yii2 mapeia diretamente uma tabela do banco de dados, evitando escrever SQL de forma manual. Isso torna o código mais legível e reduz o risco de erros de sintaxe ou vulnerabilidades.

### 4.3 Observer (Behaviors)

Behaviors são “observadores” que escutam eventos do ciclo de vida dos modelos, como antes de inserir ou atualizar.

Eles permitem automatizar tarefas auxiliares, por exemplo, preencher timestamps ou gerar slugs, assim, delegando ações repetitivas a componentes independentes.

## 5 Teste de Funcionalidade



O teste funcional foi executado com o Katalon Studio, validando o fluxo principal da aplicação: autenticação de usuário, cadastro de uma nova receita e redirecionamento automático de volta à lista de receitas. O log exibido confirma que todas as etapas foram concluídas com sucesso.

## 6 Teste de Carga

Summary Report

Name:

Summary Report

Comments:

Write results to file / Read from file

Filename

Browse...

Log/Display Only: ☐ Errors ☐ Successes 

Configure

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
Login	80	5	4	9	1.19	0.00%	54.0/min	27.91	0.23	31751.0
Ustar Receitas	80	33	13	55	12.44	0.00%	54.0/min	104.91	0.15	119402.2
Criar Receita	80	32	23	44	5.02	0.00%	54.0/min	26.87	0.83	30594.3
TOTAL	240	23	4	55	15.01	0.00%	2.7/sec	159.61	1.21	60582.5

Para avaliar o desempenho da aplicação, foi utilizado o Apache Jmeter. O teste simulou 80 usuários simultâneos acessando a aplicação ao longo de um período de 90 segundos.

Durante o teste, foram avaliadas três operações principais:

- Login de usuário
- Listagem de receitas
- Criação de nova receita

Os resultados indicaram que todas as requisições foram executadas com sucesso, sem ocorrência de erros. O **tempo médio de resposta** para cada operação foi:

- **Login:** 5 ms
- **Listar Receitas:** 33 ms
- **Criar Receita:** 32 ms

Esses tempos demonstram um bom desempenho da aplicação sob carga moderada, mantendo respostas rápidas e estáveis mesmo com múltiplos usuários simultâneos.

## 7 Teste Unitários

A aplicação foi testada utilizando o Codeception, framework de testes integrado ao Yii2. Os testes cobrem os principais modelos e regras de negócio, garantindo que as validações, geração de slugs, timestamps e autenticação estejam funcionando corretamente.

Foram desenvolvidos os seguintes testes:

- **CategoryTest**  
Valida obrigatoriedade do nome, geração automática de slugs, unicidade, timestamps e listagem para dropdown.

- **LoginFormTest**  
Verifica os comportamentos do login com usuário inexistente, senha incorreta e login bem-sucedido.
- **RecipeTest**  
Testa a validação de campos obrigatórios, geração de slug único, timestamps e o relacionamento com favoritos.
- **UserTest**  
Valida busca por ID, token de acesso, nome de usuário e autenticação de senha e chave de sessão.

## RESULTADOS

Após a conclusão do processo de desenvolvimento, o sistema "**Recipes**" apresentou resultados positivos quanto ao seu funcionamento, usabilidade e organização estrutural. O sistema permite aos usuários interagirem de forma intuitiva com as principais funcionalidades propostas.

Essas funcionalidades foram implementadas de maneira consistente com os princípios do padrão **MVC** e utilizando o **ActiveRecord** para integração segura com o banco de dados. A interface gráfica foi desenvolvida com base na arquitetura de templates do Yii, assegurando coerência visual e boa experiência de navegação.

Os testes indicaram que o sistema é funcional, estável e atende aos requisitos definidos no levantamento inicial. O uso do ORM via ActiveRecord contribuiu para a consistência na manipulação dos dados e reduziu significativamente a ocorrência de erros em operações com o banco.

Apesar dos obstáculos, o processo proporcionou **aprendizado prático significativo** sobre desenvolvimento web com frameworks modernos, estruturação de código com padrões arquiteturais e segurança no tratamento de dados. A experiência contribuiu diretamente para o amadurecimento na aplicação dos conceitos de engenharia de software e desenvolvimento backend com PHP.

## Referências

Copyright © 2008-2025 by Yii Software. Guia Definitivo para Yii 2. Disponível em: Parte superior do formulário Parte inferior do formulário

<https://www.yiiframework.com/doc/guide/2.0/pt-br/start-installation>

Link da Apresentação

<https://www.youtube.com/watch?v=ORgeENfMR1o>

Link do Repositório no Github

<https://github.com/ThiagoThomas09/recipes>