

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 NotFoundException(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.

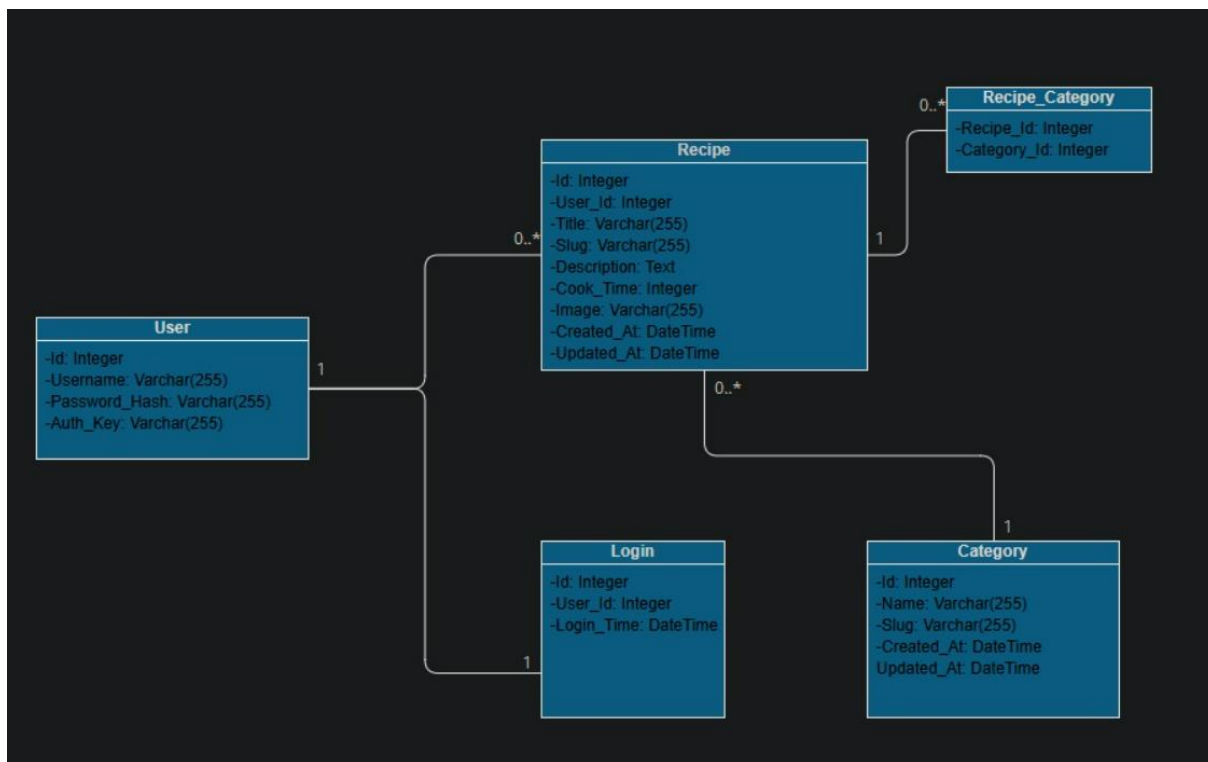
3.2 Modelagem do Banco de Dados

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.



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 NotFoundHttpException(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.

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