

Desenvolvimento de Sistemas Administrativos com Yii2

Felipe Eduardo Bohnen¹

Miguel Schneiders Flach²

Murilo Morosini³

Roney Bieger Anschau⁴

Leandro Otavio Cordova Vieira⁵

Resumo

Este artigo apresenta um relato de experiência detalhado sobre o desenvolvimento de um sistema administrativo utilizando o *framework* PHP Yii2. O projeto foi realizado no contexto da disciplina de Programação III da Universidade do Oeste de Santa Catarina (UNOESC), como parte das atividades práticas voltadas à compreensão da organização de projetos com *frameworks* PHP. A proposta consistiu na implementação de um sistema com funcionalidades CRUD, uso de ORM, padrão MVC e controle de versão com Git. Este trabalho detalha todas as etapas do projeto, desde a concepção da ideia e modelagem da aplicação até os desafios enfrentados no desenvolvimento, testes e publicação no GitHub.

Palavras-chave: Yii2, *Framework* PHP, Desenvolvimento Web, CRUD, MVC, ORM.

¹ Graduando do Curso de Ciência da Computação - Unoesc Campus São Miguel do Oeste. felipebohnenf@gmail.com

² Graduando do Curso de Ciência da Computação - Unoesc Campus São Miguel do Oeste. miguelsflach65@gmail.com

³ Graduando do Curso de Ciência da Computação - Unoesc Campus São Miguel do Oeste. murilomorosini@gmail.com

⁴ Graduando do Curso de Ciência da Computação - Unoesc Campus São Miguel do Oeste. roneyronyy@hotmail.com

⁵ Docente do Curso de Ciência da Computação - Unoesc Campus Videira. leandro.vieira@unoesc.edu.br

1. INTRODUÇÃO

O cenário atual do desenvolvimento de aplicações web exige muito mais do que apenas proficiência em lógica de programação e gerenciamento de banco de dados. Ele demanda uma compreensão aprofundada da estruturação de código, da reutilização eficiente de componentes, da segregação de responsabilidades e da aplicação de boas práticas de desenvolvimento. Nesse contexto, os *frameworks* PHP emergem como ferramentas cruciais, fornecendo estruturas e conjuntos de ferramentas que não apenas organizam o processo de desenvolvimento, mas também otimizam o tempo de produção e promovem a manutenibilidade a longo prazo (FOWLER, 2002; SOMMERVILLE, 2011).

2. DESENVOLVIMENTO

Frameworks podem ser compreendidos como conjuntos coesos de ferramentas, bibliotecas, convenções e boas práticas que atuam como um esqueleto ou uma estrutura base para o desenvolvimento de sistemas. Eles auxiliam os desenvolvedores na criação de aplicações de maneira mais rápida, organizada, segura e padronizada (LEFF; RAYFIELD, 2001). No universo PHP, a adoção de *frameworks* tornou-se uma prática amplamente disseminada, tanto em ambientes corporativos de grande porte quanto em projetos acadêmicos e de menor escala. Essa popularidade é justificada, em grande parte, pelo suporte intrínseco que oferecem a padrões arquitetônicos consolidados, pela promoção intensiva da reutilização de código e pela elevação do nível de abstração no processo de codificação (GASPARINI; GUEDES, 2020).

A seleção de um *framework* adequado é uma decisão estratégica que deve ser guiada por múltiplos fatores. Dentre eles, destacam-se a complexidade inerente do projeto, a curva de aprendizado associada ao *framework* escolhido, a gama de recursos e funcionalidades que ele oferece e, crucialmente, sua compatibilidade com o escopo e os requisitos específicos do sistema a ser desenvolvido. Adicionalmente, *frameworks* que adotam rigorosamente o padrão MVC (*Model-View-Controller*) e oferecem integração nativa e robusta com ORM (*Object-Relational Mapping*) tendem a simplificar significativamente a manutenção do sistema e aprimorar sua capacidade de escalabilidade futura (DA SILVA; SANTOS, 2019).

2.1. JUSTIFICATIVA DA ESCOLHA DO FRAMEWORK

A decisão de empregar o Yii2 no desenvolvimento deste projeto foi o resultado de uma análise comparativa criteriosa e aprofundada entre os *frameworks* PHP mais proeminentes do mercado: Laravel, Symfony, CodeIgniter, Phalcon e o próprio Yii2. Nossa avaliação considerou diversos aspectos técnicos e práticos que seriam determinantes para o sucesso do projeto e para a experiência de aprendizado da equipe. Os principais critérios de avaliação incluíram: a curva de aprendizado, que impactaria diretamente o tempo de ramp-up da equipe; a produtividade, visando a otimização do tempo de desenvolvimento; o suporte robusto a operações CRUD (*Create, Read, Update, Delete*), essenciais para um sistema administrativo; a integração simplificada e eficiente com bancos de dados relacionais; e a disponibilidade de recursos de geração automática de código, que acelerariam as tarefas repetitivas (KOTTAKIS; LOPES, 2018).

Nessa análise, o Yii2 emergiu como a opção mais vantajosa e completa, destacando-se por oferecer um conjunto de funcionalidades poderosas e bem integradas:

Gii (Gerador de CRUD Automático): Uma das maiores vantagens do Yii2 é sua ferramenta de geração de código, o Gii. Ele permite a criação automática de modelos, controladores e visões para operações CRUD, economizando um tempo considerável no desenvolvimento de funcionalidades básicas e repetitivas (YII FRAMEWORK, 2024).

Active Record (ORM Intuitivo): O Yii2 oferece um sistema de ORM baseado no padrão *Active Record*, que facilita enormemente a interação com o banco de dados. Com ele, as operações de manipulação de dados podem ser realizadas de forma orientada a objetos, tornando o código mais limpo, legível e menos propenso a erros de SQL (MARTIN, 2009).

Suporte Completo ao Padrão MVC: O *framework* implementa de forma rigorosa o padrão *Model-View-Controller*, promovendo a clara separação de responsabilidades. Isso resulta em um código mais organizado, modular e fácil de manter, permitindo que diferentes membros da equipe trabalhem em partes distintas do sistema sem conflitos (SOMMERVILLE, 2011).

Sistema de Migrations para Controle de Versão do Banco de Dados: As *migrations* do Yii2 são ferramentas essenciais para gerenciar as alterações no esquema do banco de dados ao longo do tempo. Elas permitem que as modificações sejam

controladas por versão, aplicadas e revertidas de forma programática, garantindo a sincronização do banco de dados entre os diferentes ambientes de desenvolvimento e produção (YII FRAMEWORK, 2024).

Estrutura Modular e Arquitetura Limpa: O Yii2 é projetado com uma arquitetura modular, o que significa que o sistema pode ser dividido em módulos independentes, facilitando a escalabilidade e a organização de projetos complexos. Sua estrutura limpa e bem definida contribui para a clareza e a manutenibilidade do código (DA SILVA; SANTOS, 2019).

Além das características técnicas, a escolha também foi significativamente influenciada pela experiência prévia de alguns integrantes da equipe com o Yii2, o que reduziu a curva de aprendizado inicial, e pela qualidade e abrangência de sua documentação oficial.

3. METODOLOGIA

A metodologia de desenvolvimento seguiu uma abordagem incremental, começando pela modelagem da aplicação, passando pela configuração do ambiente, criação do banco de dados, geração do CRUD com o Gii, até a personalização das *views* e funcionalidades.

A primeira etapa consistiu na instalação do Yii2 utilizando o Composer e na escolha do template "*basic*", que fornece uma estrutura inicial enxuta, com exemplos funcionais de login, navegação e controle de acesso. Em seguida, foi realizada a configuração da conexão com o banco de dados no arquivo `db.php`, conforme mostra a Figura 1.

Figura 1 - Arquivo de configuração `db.php`



Fonte: Os Autores (2025).

Com a conexão estabelecida, passou-se à definição da estrutura do banco de dados. A entidade "Produto" foi modelada com os seguintes atributos principais: nome (*string*), descrição (*string*), preço (*float*) e estoque (*inteiro*). As *migrations* foram utilizadas para criar a tabela de forma versionada e segura.

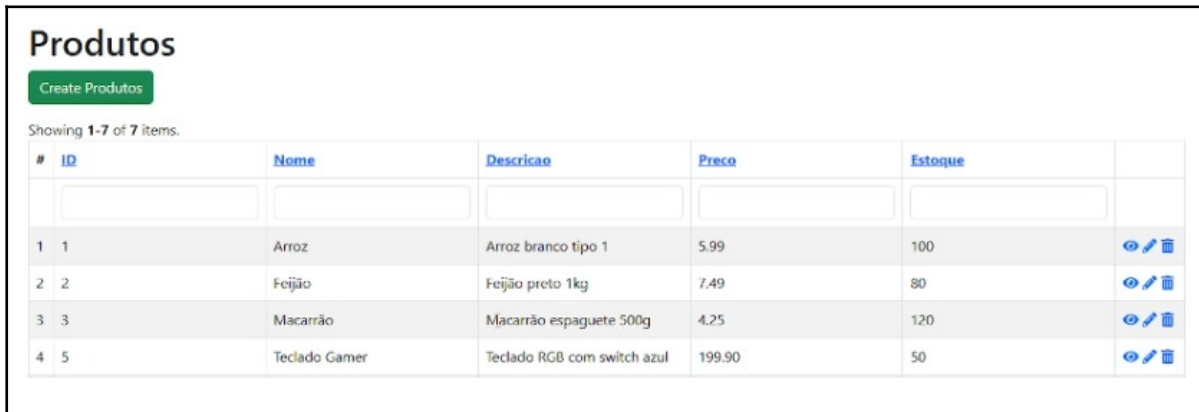
Na sequência, utilizou-se a ferramenta Gii para gerar automaticamente os arquivos do modelo (*Model*), controlador (*Controller*) e as views associadas (*Views*). A geração do modelo Produtos.php pode ser visualizada na Figura 2, onde são definidas também as regras de validação, tipos de dados e rótulos para os atributos.

Figura 2 - Modelo Produtos.php com regras de validação









```
Produtos.php X
models > Produtos.php > ...
1  <?php
2
3  namespace app\models;
4
5  use Yii;
6
7  /**
8   * This is the model class for table "produtos".
9   *
10   * @property int $id
11   * @property string $nome
12   * @property string $descricao
13   * @property float $preco
14   * @property int $estoque
15   */
16  class Produtos extends \yii\db\ActiveRecord
17  {
18
19
20      /**
21       * {@inheritdoc}
22       */
23      public static function tableName()
24      {
25          return 'produtos';
26      }
27
28      /**
29       * {@inheritdoc}
30       */
31      public function rules()
32      {
33          return [
34              [['nome', 'descricao', 'preco', 'estoque'], 'required'],
35              [['descricao'], 'string'],
36              [['preco'], 'number'],
37              [['estoque'], 'integer'],
38              [['nome'], 'string', 'max' => 255],
39          ];
40      }
41
42      /**
43       * {@inheritdoc}
44       */
45      public function attributeLabels()
46      {
47          return [
48              'id' => 'ID',
49              'nome' => 'Nome',
50              'descricao' => 'Descricao',
51              'preco' => 'Preco',
52              'estoque' => 'Estoque',
53          ];
54      }
55  }
```

A aplicação resultante já continha as funcionalidades CRUD completamente operacionais. Foi realizada então a personalização das *views* para melhorar a experiência do usuário, incluindo filtros por colunas, ordenação automática e paginação, como ilustrado na Figura 3.

Figura 3 - Tela de listagem de produtos com filtros



The screenshot shows a web interface titled "Produtos". At the top left, there is a green button labeled "Create Produtos". Below it, a text label says "Showing 1-7 of 7 items." The main content is a table with the following columns: #, ID, Nome, Descrição, Preço, and Estoque. The table contains four rows of data. Each row has a set of three icons (eye, pencil, trash) on the right side. Above the table, there are input fields for filtering the data.

#	ID	Nome	Descrição	Preço	Estoque	
1	1	Arroz	Arroz branco tipo 1	5.99	100	  
2	2	Feijão	Feijão preto 1kg	7.49	80	  
3	3	Macarrão	Macarrão espaguete 500g	4.25	120	  
4	5	Teclado Gamer	Teclado RGB com switch azul	199.90	50	  

Fonte: Os Autores (2025).

4. RESULTADOS E DISCUSSÃO

O sistema foi entregue com todas as funcionalidades propostas. A estrutura gerada pelo Yii2 mostrou-se limpa, modular e de fácil compreensão. A separação clara entre modelo, controlador e visualização permitiu uma distribuição eficiente das tarefas entre os membros da equipe, bem como uma rápida localização de eventuais erros ou pontos de melhoria.

O uso do *Active Record* trouxe uma camada de abstração bastante intuitiva, permitindo operações como inserção, atualização e consulta a dados sem a necessidade de escrever SQL diretamente. Isso não apenas facilitou o desenvolvimento, mas também reduziu o risco de erros comuns em consultas, como falhas de sintaxe ou vulnerabilidades de injeção.

Ademais, o sistema de *migrations* garantiu que todas as alterações estruturais fossem versionadas e pudessem ser replicadas entre ambientes (desenvolvimento, testes, produção) de forma segura e controlada. Essa característica é essencial para projetos em equipe e para a manutenção do histórico de modificações no banco de dados.

Foi possível observar que, mesmo com uma aplicação simples, o Yii2 oferece recursos que facilitam a expansão futura, como autenticação de usuários, controle de acesso baseado em regras e integração com componentes externos.

5. CONSIDERAÇÕES FINAIS

O desenvolvimento do sistema de cadastro de produtos utilizando o Yii2 proporcionou uma experiência enriquecedora. A ferramenta demonstrou ser robusta e versátil, mesmo em projetos de pequena escala. O ganho de produtividade com o uso do Gii e a estruturação clara oferecida pelo MVC tornaram o processo de desenvolvimento mais fluido, padronizado e seguro.

A experiência adquirida neste projeto representa uma base sólida para aplicações futuras, uma vez que o Yii2 pode ser ampliado para incluir funcionalidades mais complexas como gestão de usuários, painéis administrativos, relatórios e integração com APIs externas.

Em suma, o projeto serviu não apenas como prova de conceito da eficácia do Yii2 em aplicações CRUD, mas também como uma oportunidade de aprendizado valiosa para compreensão dos fundamentos do desenvolvimento web moderno com PHP.

<https://www.youtube.com/watch?v=WjMLJ7sIlyo>

REFERÊNCIAS

DA SILVA, V.; SANTOS, L. **Desenvolvimento de aplicações web com PHP e MVC: Um estudo comparativo entre frameworks**. Revista de Informática Aplicada, v. 15, n. 2, p. 45-56, 2019.

FOWLER, M. **Patterns of Enterprise Application Architecture**. Boston: Addison-Wesley, 2002.

GASPARINI, I.; GUEDES, L. **Engenharia de Software e Desenvolvimento Web**. Rio de Janeiro: LTC, 2020.

KOTTAKIS, D.; LOPES, R. **Comparativo de Frameworks PHP sob a ótica de produtividade e escalabilidade**. Anais do Simpósio Brasileiro de Engenharia de Software, 2018.

LEFF, A.; RAYFIELD, J. T. **Web-application development using the Model/View/Controller design pattern**. In: *Proceedings of the IEEE International Enterprise Distributed Object Computing Conference*, 2001.

MARTIN, R. C. **Clean Code: A Handbook of Agile Software Craftsmanship**. Upper Saddle River: Prentice Hall, 2009.

SOMMERVILLE, I. **Engenharia de Software**. 9. ed. São Paulo: Pearson, 2011.

YII FRAMEWORK. **Yii2 Guide Documentation**. Disponível em: <https://www.yiiframework.com/doc/guide/2.0/en>. Acesso em: 19 maio 2025.