

Sistema Web para Controle de Estoque

Marco Antonio Borghetti¹

Maria Isabel Wirth Marafon²

Vinícius Andrei Wille Wonka³

Leandro Otavio Cordova Vieira⁴

Trabalho apresentado ao curso de Ciência da Computação como parte dos requisitos para avaliação no componente curricular de Programação III.

¹Discente do Curso de Ciência da Computação
Unoesc-Campus de São Miguel do Oeste
Rua Oiapoc, 2011. São Miguel do Oeste-SC
marco.borghetti@unoesc.edu.br

²Discente do Curso de Ciência da Computação
Unoesc-Campus de São Miguel do Oeste
Rua Oiapoc, 2011. São Miguel do Oeste-SC
maria.im@unoesc.edu.br

³Discente do Curso de Ciência da Computação
Unoesc-Campus de São Miguel do Oeste
Rua Oiapoc, 2011. São Miguel do Oeste-SC
vinicius.wille@unoesc.edu.br

⁴Docente do Curso de Ciência da Computação
Unoesc-Campus de São Miguel do Oeste
Rua Oiapoc, 2011. São Miguel do Oeste-SC
leandro.vieira@unoesc.edu.br

RESUMO

A ausência de um sistema informatizado para controle de estoque ainda é uma realidade em diversas pequenas empresas, que recorrem a registros manuais sujeitos a falhas, atrasos e falta de visibilidade sobre a situação real dos produtos armazenados. Este artigo apresenta o desenvolvimento de uma aplicação web responsiva e de fácil uso para controle de estoque, construída com PHP, MySQL e o framework Yii2. A solução permite o cadastro de produtos, registro de movimentações de entrada e saída, consulta de estoque em tempo real, dashboard com dados agregados, e controle de acesso por autenticação de usuários. A proposta visa reduzir erros operacionais e aumentar a eficiência da gestão de inventário com um sistema leve, de rápida implantação e baixa curva de aprendizagem.

Palavras-chave: Controle de Estoque. Sistema Web. Pequenas Empresas. Yii2. PHP. MySQL. Auditoria de Inventário.

SUMÁRIO

1. INTRODUÇÃO.....	4
2. FUNDAMENTAÇÃO TÉCNICA.....	4
3. OBJETIVOS.....	4
3.1 Objetivo Geral.....	4
3.2 Objetivos Específicos.....	5
4. METODOLOGIA.....	5
5. DESENVOLVIMENTO.....	5
5.1 Testes e Execução.....	6
6. RESULTADOS OBTIDOS.....	7
7. CONSIDERAÇÕES FINAIS.....	7
8. APRENDIZADOS E EXPERIÊNCIA DO GRUPO.....	8
9. REFERÊNCIAS.....	8

1. INTRODUÇÃO

O controle de estoque é uma das atividades mais críticas dentro de uma organização comercial, independentemente do seu porte. O gerenciamento inadequado pode causar impactos financeiros significativos, como perdas por vencimento de produtos, falta de mercadorias para venda, ou excesso de itens estocados. Pequenas empresas, em especial, enfrentam desafios adicionais devido à escassez de recursos para adoção de soluções tecnológicas robustas.

Neste contexto, a digitalização dos processos de estoque surge como uma necessidade urgente. Este trabalho apresenta o desenvolvimento de um sistema web simples e funcional que permite o controle de estoque de forma eficiente e confiável, utilizando tecnologias amplamente conhecidas e acessíveis.

2. FUNDAMENTAÇÃO TÉCNICA

O desenvolvimento da aplicação utilizou princípios da programação orientada a objetos (POO), o que permitiu maior modularidade, reutilização de código e manutenção facilitada. A arquitetura do projeto segue o padrão MVC (Model-View-Controller), separando responsabilidades entre a lógica de negócios, a interface do usuário e a manipulação de dados.

A autenticação dos usuários foi implementada utilizando `password_hash()` e `password_verify()`, promovendo segurança no armazenamento de senhas. O controle de sessões foi aplicado para garantir que apenas usuários autenticados pudessem acessar funcionalidades sensíveis, como o cadastro de produtos ou movimentações de estoque.

A aplicação também faz uso de versionamento de banco de dados por meio de *migrations* e gestão de dependências através do *Composer*, o que facilita o deploy e a colaboração entre desenvolvedores.

3. OBJETIVOS

3.1 Objetivo Geral

Desenvolver um sistema web responsivo, acessível e de fácil utilização para controle de estoque em pequenas empresas.

3.2 Objetivos Específicos

- Permitir o cadastro, edição e exclusão de produtos.
- Registrar movimentações de entrada e saída com rastreabilidade.
- Exibir dashboards com dados de produtos e valor total em estoque.
- Restringir funcionalidades através de autenticação de usuário.
- Garantir usabilidade por meio de uma interface simples e intuitiva.

4. METODOLOGIA

A metodologia adotada para o desenvolvimento do sistema seguiu princípios da engenharia de software com foco em desenvolvimento incremental, priorizando entregas contínuas e testáveis. Utilizou-se uma abordagem prática com o apoio de ferramentas modernas e boas práticas de desenvolvimento web. A equipe optou por organizar o trabalho em etapas curtas e bem definidas, permitindo revisões constantes e ajustes durante o ciclo de desenvolvimento.

O processo foi orientado por práticas comuns da modelagem orientada a objetos e da divisão clara de responsabilidades entre back-end, front-end e banco de dados. Todas as decisões técnicas foram tomadas considerando a facilidade de manutenção, simplicidade para o usuário final e possibilidade de expansão futura do sistema.

5. DESENVOLVIMENTO

A linguagem de programação escolhida foi o PHP na versão 7.4, amplamente utilizada no desenvolvimento web. Para organizar e estruturar o código, adotamos o framework Yii2, que além de ser leve e de fácil configuração, oferece suporte nativo ao padrão MVC (*Model-View-Controller*), facilitando a separação de responsabilidades e a escalabilidade do projeto. O banco de dados utilizado foi o MySQL, por sua confiabilidade e compatibilidade com o ambiente de desenvolvimento escolhido, o XAMPP. As dependências foram gerenciadas via Composer.

Na camada de apresentação, utilizamos HTML e CSS com o auxílio do Bootstrap, garantindo responsividade e boa usabilidade em diferentes tamanhos de tela. O Chart.js foi utilizado na construção dos gráficos de movimentação do dashboard, permitindo uma visualização amigável dos dados.

A estrutura do projeto foi organizada em diretórios funcionais: controladores responsáveis pela lógica da aplicação, modelos com regras de validação e acesso ao banco de dados, e views contendo as interfaces renderizadas. Scripts de inicialização, configurações de banco e *migrations* foram estruturados de forma clara, permitindo fácil replicação do ambiente.

O desenvolvimento seguiu as seguintes etapas práticas: inicialmente, modelamos os dados com base nos requisitos. Em seguida, criamos as migrations que estruturam o banco de dados. Posteriormente, desenvolvemos os CRUDs de produtos e movimentações, integrando formulários e tabelas com validação de dados. A seguir, implementamos o layout com Bootstrap e criamos o dashboard informativo com cards e gráficos. Na reta final, foi implementado o sistema de autenticação com hashes de senha e sessões PHP, restringindo o acesso às rotas principais. Por fim, realizamos testes manuais e ajustes com base no feedback de uso.

Essa abordagem proporcionou um fluxo de trabalho eficiente, com entregas frequentes e funcionais. O uso das ferramentas modernas, como o Yii2, Composer e Bootstrap, contribuiu para um desenvolvimento rápido, organizado e com foco na experiência do usuário final.

5.1 Testes e Execução

Como parte do processo de desenvolvimento da aplicação, foram realizados testes manuais e validações constantes em cada etapa. A equipe optou por uma abordagem prática, testando os módulos logo após sua implementação para garantir seu correto funcionamento e detectar eventuais falhas de forma antecipada. Isso incluiu o cadastro e edição de produtos, registro de movimentações e verificação dos dados refletidos no dashboard.

A execução do projeto em ambiente local foi facilitada pela adoção do XAMPP, que forneceu os serviços necessários do Apache e MySQL para hospedar a aplicação. O framework Yii2, utilizado no desenvolvimento, oferece um servidor embutido para testes rápidos, executado por meio do comando *php yii serve*.

Para executar o sistema, é necessário ter o PHP 7.4 ou superior instalado, juntamente com o Composer para gerenciar as dependências. Após clonar o repositório e acessar o diretório do projeto, realiza-se a instalação das dependências com *composer install*. Em

seguida, as tabelas do banco de dados são geradas por meio do comando `php yii migrate`, que executa os scripts de criação definidos nas *migrations*. Por fim, o servidor pode ser iniciado com `php yii serve --port=8080`, tornando a aplicação acessível via navegador no endereço <http://localhost:8080>.

Antes disso, é importante configurar corretamente as credenciais de acesso ao banco de dados no arquivo `config/db.php`, garantindo que a conexão com o MySQL seja estabelecida com sucesso.

Esses testes e procedimentos asseguraram que o sistema fosse entregue funcional, com todos os recursos operando de forma integrada e estável.

6. RESULTADOS OBTIDOS

O sistema entregue apresentou um conjunto sólido de funcionalidades que atendem aos principais requisitos definidos no início do projeto. Entre os principais resultados, destaca-se a implementação do módulo de cadastro de produtos, onde é possível informar o nome, a unidade de medida (como UN, ME, KG ou LT) e o preço em reais com formatação adequada. Além disso, foi desenvolvido um módulo de movimentações de estoque, no qual o operador pode registrar entradas e saídas de produtos, escolhendo o item em um campo de seleção, definindo a quantidade e a data da movimentação.

Outro destaque foi o dashboard informativo, que apresenta em tempo real o total de produtos cadastrados, o valor financeiro estimado do estoque e um gráfico interativo de movimentações ao longo do tempo, utilizando a biblioteca Chart.js. A segurança e o controle de acesso também foram considerados com a criação de um sistema de autenticação de usuários, com telas de login e cadastro, uso de hash para senha e restrição de rotas. Por fim, a interface responsiva foi desenvolvida com o framework Bootstrap, contendo elementos de navegação claros, labels explicativos, breadcrumbs e botões compactos, que garantem uma boa experiência de uso em diferentes dispositivos.

7. CONSIDERAÇÕES FINAIS

O sistema desenvolvido se mostrou eficiente em resolver os principais desafios enfrentados por pequenas empresas no controle de estoque. Sua interface simples, aliada às funcionalidades essenciais, proporciona ganhos em precisão, agilidade e organização, sem exigir infraestrutura complexa ou conhecimento técnico avançado por parte do usuário.

Para trabalhos futuros, recomenda-se:

- Exportação de relatórios (PDF/Excel);
- Níveis de permissão por usuário;
- Integração com sistemas de vendas ou ERPs.

Para conhecer mais sobre o desenvolvimento e funcionamento do sistema, o vídeo explicativo está disponível em: <https://youtu.be/Mxxuubww-TY>.

O código-fonte completo pode ser acessado no repositório GitHub: https://github.com/Wyllye/Prog3_Framework.

8. APRENDIZADOS E EXPERIÊNCIA DO GRUPO

Durante o desenvolvimento do projeto, cada integrante do grupo teve a oportunidade de aplicar e aprimorar conhecimentos práticos em PHP, banco de dados, estruturação de projetos com MVC, autenticação de usuários e front-end responsivo. Além do desenvolvimento técnico, houve avanço significativo na organização do trabalho em equipe, divisão de tarefas e no uso de ferramentas de versionamento e testes. A entrega do projeto possibilitou vivenciar na prática a criação de uma solução real com foco na usabilidade e impacto direto em um problema cotidiano de gestão.

9. REFERÊNCIAS

BALLOU, Ronald H. *Logística empresarial*. 1. ed. São Paulo: Atlas, 2006.

LAUDON, Kenneth C.; LAUDON, Jane P. *Sistemas de informação gerenciais*. São Paulo: Pearson, 2012.

YII Framework. *Yii2 Documentation*. Disponível em: <https://www.yiiframework.com/doc>. Acesso em: jun. 2025.

. Metodologia

Durante o desenvolvimento deste sistema, optou-se por uma metodologia baseada na engenharia de software tradicional, porém aplicada de forma prática e iterativa. O foco esteve na entrega funcional por etapas, sempre garantindo testes frequentes e versões utilizáveis ao longo do processo. A equipe iniciou os trabalhos com o levantamento de requisitos fundamentais, priorizando funcionalidades essenciais ao controle de estoque, como cadastro de produtos, movimentações e controle de acesso.

A linguagem de programação escolhida foi o PHP na versão 7.4, amplamente utilizada no desenvolvimento web. Para organizar e estruturar o código, adotamos o framework Yii2, que além de ser leve e de fácil configuração, oferece suporte nativo ao padrão MVC (Model-View-Controller), facilitando a separação de responsabilidades e a escalabilidade do projeto. O banco de dados utilizado foi o MySQL, por sua confiabilidade e compatibilidade com o ambiente de desenvolvimento escolhido, o XAMPP. As dependências foram gerenciadas via Composer.

Na camada de apresentação, utilizamos HTML e CSS com o auxílio do Bootstrap, garantindo responsividade e boa usabilidade em diferentes tamanhos de tela. O Chart.js foi utilizado na construção dos gráficos de movimentação do dashboard, permitindo uma visualização amigável dos dados.

A estrutura do projeto foi organizada em diretórios funcionais: controladores responsáveis pela lógica da aplicação, modelos com regras de validação e acesso ao banco de dados, e views contendo as interfaces renderizadas. Scripts de inicialização, configurações de banco e migrations foram estruturados de forma clara, permitindo fácil replicação do ambiente.

O desenvolvimento seguiu as seguintes etapas práticas: inicialmente, modelamos os dados com base nos requisitos. Em seguida, criamos as migrations que estruturam o banco de dados. Posteriormente, desenvolvemos os CRUDs de produtos e movimentações, integrando formulários e tabelas com validação de dados. A seguir, implementamos o layout com Bootstrap e criamos o dashboard informativo com cards e gráficos. Na reta final, foi implementado o sistema de autenticação com hashes de senha e sessões PHP, restringindo o acesso às rotas principais. Por fim, realizamos testes manuais e ajustes com base no feedback de uso.

Essa abordagem proporcionou um fluxo de trabalho eficiente, com entregas frequentes e funcionais. O uso das ferramentas modernas, como o Yii2, Composer e Bootstrap, contribuiu para um desenvolvimento rápido, organizado e com foco na experiência do usuário final.