

## PROPOSTA DE SISTEMA AUTOMÁTICO DE GESTÃO DE COMPONENTES ELETRÔNICOS ALMOXARIFADO DO IFSP SJC

**Aldeni Lopes Pinheiro Rodrigues, Otávio Augusto Coutinho de Araújo, Vitor de Oliveira Holanda, Mirella Caetano de Souza, Carlos Eduardo Oliveira da Silva, Júlio Cesar Serafim Casini,**

Instituto Federal de Educação, Ciência e Tecnologia de São Paulo (*IFSP*), *Campus São José dos Campos*, Rodovia Presidente Dutra, km 145, s/n, Jardim Diamante – 12223-201 – São José dos Campos-SP, Brasil, [aldeni.l@aluno.ifsp.edu.br](mailto:aldeni.l@aluno.ifsp.edu.br), [otavio.araujo@aluno.ifsp.edu.br](mailto:otavio.araujo@aluno.ifsp.edu.br), [vitor.holanda@aluno.ifsp.edu.br](mailto:vitor.holanda@aluno.ifsp.edu.br), [carlossilva@ifsp.edu.br](mailto:carlossilva@ifsp.edu.br), [julio.casini@ifsp.edu.br](mailto:julio.casini@ifsp.edu.br), [mirella.souza@ifsp.edu.br](mailto:mirella.souza@ifsp.edu.br)

### Resumo

Este artigo apresenta o estudo e desenvolvimento de um sistema web para controle de estoque de almoxarifados de componentes eletrônicos, utilizando a linguagem de programação Python. O sistema é projetado para monitorar a gestão desses materiais, abrangendo o cadastro de componentes, consulta de estoque, atualização de quantidades, histórico de retirada e devoluções e notificações de reabastecimento. As etapas de desenvolvimento incluem a análise de requisitos, a implementação e a melhoria contínua do sistema. A plataforma proposta visa facilitar a gestão do estoque, assegurando um controle rigoroso e preciso, essencial para a otimização de recursos e minimização de desperdícios no almoxarifado.

**Palavras-chave:** Controle de Estoque, Python, Desenvolvimento Web

**Curso:** Engenharia de Produção

### Introdução

Conforme destacado por Tubino (2007,p.67), “as empresas trabalham com estoques de diferentes tipos que serão administrados [...]”, a gestão de estoques é uma atividade crucial no contexto da fabricação, pois envolve a administração de materiais com o objetivo de alcançar níveis adequados de suprimentos. Um dos principais motivos para a implementação de um planejamento eficaz e controle rigoroso de estoques reside no impacto financeiro significativo que pode ser obtido por aumento da eficácia e eficiência das operações da Organização, pois com os níveis adequados de estoque, não há sobra de materiais e não há falta de insumos para manter a produção conforme planejado.

Gestão de estoque muitas vezes é realizada de maneira manual com uso de anotações manuscritas ou por meios de planilhas. Ainda que cumpra seu papel principal, o risco de erros de digitação é maior, comparada ao uso automático. Nesse contexto, o uso da tecnologia, a fim de ajudar nas atividades de gerenciamento de estoque, vem se tornando popular. Os softwares transformaram-se em um dos recursos mais eficazes capazes de transferir dados de maneira mais rápida, fácil e segura (KLIPPEL,2019).

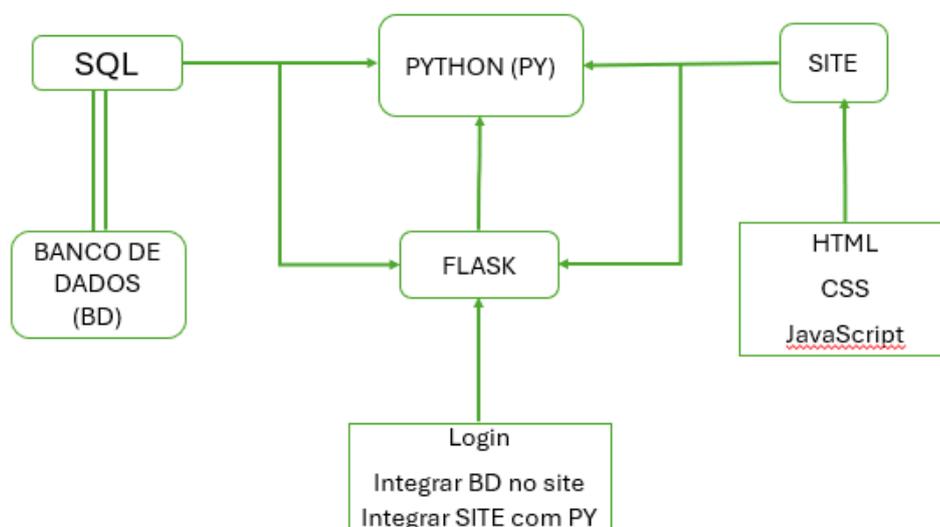
Durante as aulas do Laboratório de Engenharia de Controle e Automação no *IFSP - Campus São José dos Campos*, alertou-se que os docentes enfrentam dificuldades para atividades planejadas devido à falta de componentes e equipamentos de medição, além da dificuldade em localizá-los no estoque do almoxarifado. Diante dessa problemática, foi proposta a realização de pesquisa para o desenvolvimento e implantação de um sistema automático para controle e gestão de estoque dos componentes e equipamentos utilizados nos laboratórios do campus.

A implementação do conceito de controle de estoque será realizada por meio do desenvolvimento de um sistema web. O método para a implantação desse sistema envolve diversas etapas, incluindo a elaboração do sistema de controle de estoque, a execução de testes e a efetivação do sistema, complementada por treinamentos adequados. Sendo assim, o objetivo principal desse estudo foi unir os conhecimentos adquiridos no curso de Engenharia de Controle e Automação para desenvolver um sistema automatizado capaz de promover uma gestão mais eficiente, evitando excessos ou escassez de componentes, minimizando atrasos nos pedidos e possibilitando a geração de relatórios e análises sobre o consumo de materiais, padrões de utilização e desempenho do estoque, a fim de auxiliar na tomada de decisões estratégicas.

## Metodologia

Para o desenvolvimento do projeto do sistema de controle de estoque, foi realizada pesquisa bibliográfica e documental sobre gestão de estoques e linguagem de programação, sendo proposta a utilização da linguagem de programação Python, por ser uma linguagem aberta e de fácil interação. MUELLER (2020). Outras linguagens de programação foram utilizadas, como o HTML que definiu a estrutura do conteúdo da web, o CSS que controla o design da página. FREEMAN(2009). e o JavaScript criando interações dinâmicas. Também foi criado um banco de dados utilizando o SQL para armazenamento das informações. Foram utilizadas duas bibliotecas do Python para auxílio de funções, sendo a primeira o SQLAlchemy que utiliza uma série de comandos para extrair as informações guardadas e armazenar no banco de dados, e a segunda é o Flask para interligar as informações entre site e o banco de dados. SILVA (2019). O funcionamento da programação é descrito na forma de fluxograma e pode ser observado na Figura 1 para entendimento do sistema.

Figura 1 – Fluxograma do Desenvolvimento da Web



Fonte: Próprio Autor

## Resultados

O sistema proposto foi desenvolvido e simulado em sala de aula, mas destaca-se que ainda não foi testado pelo usuário final. Considerando os testes e simulações, se implementado, esse sistema pode desempenhar sua função de forma adequada a atender aos objetivos do setor que será implementado.

O sistema possui uma interface intuitiva e promove uma gestão eficiente do estoque, permitindo um controle de entrada e saída de itens e a possibilidade de cadastro de novos componentes. O controle de acesso é realizado por meio de login e senha e a segurança é controlada por meio da limitação de criação de contas nos registros no banco de dados.

A funcionalidade para adicionar, editar e excluir itens foi implementada com sucesso, garantindo uma integração com o banco de dados.

A integração das operações de compra, devolução e controle de itens provou ser satisfatória, facilitando a identificação de itens em falta, adquiridos e retirados. O sistema, portanto, alcançou seus objetivos de forma eficiente, oferecendo uma solução eficaz para a gestão de estoques.

As figuras 2, 3 e 4 referem-se aos trechos da programação da criação do banco de dados, login de acesso e link de acesso ao site. A Figura 5 refere-se à página de acesso criado.

Figura 2 – Programação Banco de Dados

```

app.py > ...
1 from flask import Flask, render_template, request, redirect, url_for, flash
2 from flask_sqlalchemy import SQLAlchemy
3 from flask_login import LoginManager, UserMixin, login_user, login_required, logout_user, current_user
4 from flask_migrate import Migrate
5 from datetime import datetime
6
7 app = Flask(__name__)
8 app.config['SECRET_KEY'] = 'your_secret_key'
9 app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///users.db'
10 db = SQLAlchemy(app)
11 migrate = Migrate(app, db)
12 login_manager = LoginManager()
13 login_manager.init_app(app)
14 login_manager.login_view = 'login'
15
16
17 class User(UserMixin, db.Model):
18     id = db.Column(db.Integer, primary_key=True)
19     username = db.Column(db.String(150), unique=True, nullable=False)
20     password = db.Column(db.String(150), nullable=False)
21     devolucoes = db.relationship('Devolucao', backref='user', lazy=True)
22
23 class Devolucao(db.Model):
24     id = db.Column(db.Integer, primary_key=True)
25     pessoa_retirada = db.Column(db.String(150), nullable=False)
26     quantidade = db.Column(db.Integer, nullable=False)
27     peça = db.Column(db.String(150), nullable=False)
28     tipo = db.Column(db.String(50), nullable=False)
29     user_id = db.Column(db.Integer, db.ForeignKey('user.id'), nullable=False)

```

Fonte: Próprio Autor

Figura 3 – Programação Login

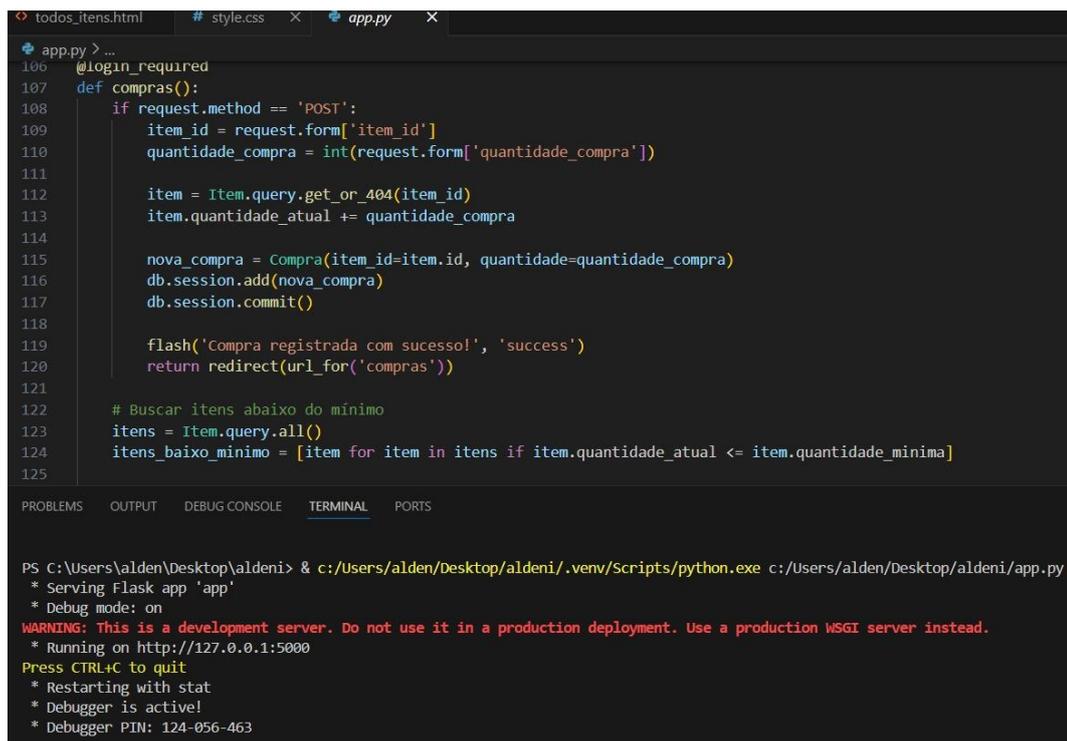
```

app.py > ...
46
47
48 @login_manager.user_loader
49 def load_user(user_id):
50     return User.query.get(int(user_id))
51
52
53 @app.route('/login', methods=['GET', 'POST'])
54 def login():
55     if request.method == 'POST':
56         username = request.form['username']
57         password = request.form['password']
58         user = User.query.filter_by(username=username).first()
59         if user and user.password == password:
60             login_user(user)
61             return redirect(url_for('home'))
62         else:
63             flash('Login Unsuccessful. Please check username and password', 'danger')
64     return render_template('login.html')
65
66
67 @app.route('/register', methods=['GET', 'POST'])
68 @login_required
69 def register():
70     if request.method == 'POST':
71         username = request.form['username']
72         password = request.form['password']
73         new_user = User(username=username, password=password)
74         db.session.add(new_user)
75         db.session.commit()

```

Fonte: Próprio Autor

Figura 4 – Link de Acesso ao site



```

app.py > ...
106 @login_required
107 def compras():
108     if request.method == 'POST':
109         item_id = request.form['item_id']
110         quantidade_compra = int(request.form['quantidade_compra'])
111
112         item = Item.query.get_or_404(item_id)
113         item.quantidade_atual += quantidade_compra
114
115         nova_compra = Compra(item_id=item.id, quantidade=quantidade_compra)
116         db.session.add(nova_compra)
117         db.session.commit()
118
119         flash('Compra registrada com sucesso!', 'success')
120         return redirect(url_for('compras'))
121
122     # Buscar itens abaixo do mínimo
123     itens = Item.query.all()
124     itens_baixo_minimo = [item for item in itens if item.quantidade_atual <= item.quantidade_minima]
125
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\alden\Desktop\aldeni> & c:/Users/alden/Desktop/aldeni/.venv/Scripts/python.exe c:/Users/alden/Desktop/aldeni/app.py
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 124-056-463
    
```

Figura 5 - Página de acesso



Fonte: Próprio Autor

## Discussão

O sistema automático de gestão de componentes atendeu a expectativa esperada e desempenhou o seu papel principal ao realizar a inserção, edição e exclusão dos itens. Durante o processo de desenvolvimento do sistema houve alguns impedimentos no quesito da escolha da linguagem de programação. A equipe chegou no acordo e foi escolhida a linguagem de programação Python, pois é uma linguagem de programação que atende aos quesitos propostos do projeto, além de ser de fácil entendimento.

## Conclusão

Embora não tenha sido possível a implementação do sistema e a validação junto aos usuários do IFSP - Campus São José dos Campos, considerando os testes e simulações realizados em ambiente controlado (sala de aula), pode-se concluir que a criação do sistema atende aos requisitos propostos e é capaz de promover a gestão dos estoques, atendendo a demanda local.

Para além do desenvolvimento do referido sistema, esse projeto permitiu explorar diversas áreas do conhecimento, desde o gerenciamento de um estoque, até utilização das linguagens de programação, alcançando o objetivo proposto para esse estudo, de unir os conhecimentos adquiridos no curso de Engenharia de Controle e Automação para desenvolver um sistema automatizado de gestão de estoques .

## Referências

FREEMAN, E.; FREEMAN, E. Use a cabeça! HTML com CSS e XHTML. 2ª.ed Alta Books, 2009; SILVA, M.S. Fundamentos de HTML5 e CSS3.;

KLIPPEL, Maycon Marques. Gestão de estoques e a curva ABC: análise em uma distribuidora de bananas no município de Cacoal/RO. 32f. Trabalho de Conclusão de Curso (Bacharelado em Ciências Contábeis) - Universidade Federal de Rondônia, Cacoal, 2019

MUELLER, John Paul. Começando a programar em Python para leigos. 2ª.ed Rio de Janeiro Alta Books 2020

SILVA, Tiago. 'Flask de A a Z: crie aplicações web mais completas e robustas em Python'. 1ª.ed Casa do Código. São Paulo. Brasil. 2019.

TUBINO, Dalvio Ferrari. Planejamento e controle da produção: teoria e prática. 2ª.ed São Paulo: Atlas, 2007