

**UNIVERSIDADE VIRTUAL DO ESTADO DE SÃO PAULO**

Bernardo Lima dos Santos, 1811895  
Bruna de Oliveira, 2211351  
Paulo Sérgio da Silva Machado, 2213937  
Roddy E. Ramos Gonzáles, 2211457

**Controle de Estoque**

Campinas – SP  
2024

**UNIVERSIDADE VIRTUAL DO ESTADO DE SÃO PAULO**

## **Controle de Estoque**

Relatório Técnico-Científico apresentado na disciplina de Projeto Integrador para o curso de DRP04 – Projeto Integrador em Computação I – Turma 005 da Universidade Virtual do Estado de São Paulo (UNIVESP).

Campinas – SP  
2024

LIMA DOS SANTOS, Bernardo; DE OLIVEIRA, Bruna; DA SILVA MACHADO, Paulo Sergio; RAMOS GONZÁLES, Roddy E. **Controle de Estoque**. 00f. Relatório Técnico-Científico. DRP04 – Projeto Integrador em Computação I – Turma 005 – **Universidade Virtual do Estado de São Paulo**. Tutor: (Nome do Tutor). Polo Campinas, 2024.

## **RESUMO**

O projeto aborda a dificuldade no controle do estoque de matéria-prima de produtos artesanais de madeira, devido à ausência de dados automatizados. A metodologia adotada compreende a criação de um Framework Web utilizando Python-Django, Banco de Dados SQL, implementação vide Docker, em conformidade de Open Source e controle de versionamento no GitHub. A solução inicial proposta consiste no desenvolvimento de um framework que permitirá o controle de usuários, dados de fornecedores e matéria-prima, além de integrar um sistema de controle de estoque alinhado com as vendas.

**PALAVRAS-CHAVE:** Controle de estoque; Framework Web; Python–Django; SQL; Docker.

**LISTA DE ILUSTRAÇÕES**

**FIGURA 1.-** PEÇAS DECORATIVAS. FONTE: PÁGINA DA EMPRESA NO INSTAGRAM, 2024..... 1

**FIGURA 2.-** PLANILHA ORIGINAL DAS MATÉRIAS-PRIMAS..... 11

**FIGURA 3.-** PLANILHA DOS PRODUTOS..... 12

**FIGURA 4.-** PLANILHA DOS PRODUTOS..... 12

**FIGURA 5.-** TELA DE ACESSO DO SUPERUSER PARA MANUTENÇÃO DOS LOGINS DE ACESSO. FONTE: AUTOR (2024)..... 14

**FIGURA 6.-** CONFIGURAÇÃO DO USUÁRIO E SUA PERMISSÕES. FONTE: AUTOR (2024)..... 15

**FIGURA 7.-** SCRIPTS DE CONDICIONAIS DE ACESSO POR USUÁRIO. FONTE: AUTOR (2024)..... 16

**FIGURA 8.-** REPOSITÓRIO DO GITHUB PARA CONTROLE DE VERSIONAMENTO. FONTE: AUTOR (2024)..... 17

## **LISTAS DE TABELAS**

TABELA 1.- TABELA PRODUTO DO BANCO DE DADOS.....	9
TABELA 2 - TABELA FORNECEDOR DO BANCO DE DADOS.....	10
TABELA 3.- TABELA CLIENTE DO BANCO DE DADOS.....	10
TABELA 4.- TABELA PEDIDO DO BANCO DE DADOS.....	11
TABELA 5.- TABELA ESTOQUE DO BANCO DE DADOS. ....	11

## SUMÁRIO

1 INTRODUÇÃO.....	1
2 DESENVOLVIMENTO.....	3
2.1 Objetivos.....	3
2.2 Justificativa e delimitação do problema .....	3
2.3 Fundamentação teórica .....	5
2.4 Metodologia.....	8
2.5 Resultados preliminares: solução inicial .....	13
REFERÊNCIAS .....	18
ANEXOS (OPCIONAL) .....	19
APÊNDICES (OPCIONAL) .....	20

## 1 INTRODUÇÃO

A BV Wood é uma empresa familiar que atua no mercado desde 2011, operando no ramo de fabricação de móveis e de peças de decoração feitos a partir de madeira de reflorestamento e com acabamento rústico. A empresa possui poucos funcionários, sendo alguns deles parentes da família, e divide-se em setores como vendas, compras, produção de peças cruas, decoração, entre outros. Em seu portfólio destacam-se os produtos decorativos, conforme demonstrado na Figura 1.



**Figura 1.-** Peças decorativas. Fonte: página da empresa no Instagram, 2024.

A empresa trabalha com sistema informatizado de gerenciamento para contato de fornecedores, banco de dados de clientes e controle de vendas. Contudo, devido à dificuldade de alimentação das informações, faz o controle de estoque de matéria-prima de forma manual, semanalmente. A ficha é impressa e os funcionários de cada setor registram a quantidade de matéria-prima existente e a quantidade a ser comprada. Entretanto, muitas vezes essas informações não refletem a realidade, introduzindo imprecisão na requisição de matéria-prima e implicando retrabalho para evitar gastos extras com a compra de material desnecessário.

Para solucionar isso, pretende-se implantar um sistema informatizado de controle de estoque de matérias-primas, com enfoque em madeira, e que seja confiável, ágil e de fácil operação.

Apesar de a empresa utilizar diversos outros materiais para a manufatura, tais como pregos, corredeiras e tintas, a BV Wood demonstrou, em reunião, que a maior dificuldade de controle ocorre com a madeira.

É importante que a organização desse insumo seja feita com cuidado, sendo necessário que a quantidade esteja garantida para a fabricação dos produtos. Assim, o banco de dados permitirá um controle, no estoque, de entrada e saída da madeira específica para cada produto



## **2 DESENVOLVIMENTO**

### **2.1 OBJETIVOS**

O objetivo é implementar um sistema que atue como ferramenta de busca e gerenciamento da madeira. Com isso será possível acompanhar o status e quantidades disponíveis para a fabricação de cada produto.

Os objetivos específicos serão:

- Digitalizar os registros dos fornecedores de madeira: o funcionário encarregado terá a tarefa de inserir os dados dos fornecedores utilizando informações já disponíveis, além de incluir informações de possíveis novos fornecedores.
- Digitalizar informações sobre pedidos de produtos: os membros da equipe de estoque serão encarregados de inserir detalhes de pedidos de produtos em um formulário on-line no sistema, que serão, então, enviados aos fornecedores.
- Digitalizar informações sobre pedidos de clientes: com base nos dados registrados no sistema de controle de estoque, o funcionário encarregado poderá consultar vendas e atualizar o estoque de produtos, evitando desperdícios com mercadorias de baixo desempenho de venda.

### **2.2 JUSTIFICATIVA E DELIMITAÇÃO DO PROBLEMA**

Os objetivos de uma empresa variam dependendo dos fatores, como missão, visão, valores, setor de atuação e estratégias específicas. Geralmente, o objetivo principal de uma empresa é

maximizar o valor para suas partes interessadas, como proprietários, gerentes, funcionários e clientes. Dentre os objetivos temos:

- Maximização do lucro: para muitas empresas, o objetivo principal é gerar lucro para os acionistas ou os proprietários.
- Crescimento: algumas empresas buscam expandir seus negócios, aumentando sua participação de mercado, expandindo para novas regiões ou desenvolvendo novos produtos e serviços.
- Satisfação do cliente: focar na satisfação e fidelização dos clientes pode ser o principal objetivo de empresas que buscam construir uma base sólida de clientes e garantir sua sobrevivência a longo prazo.
- Responsabilidade social e sustentabilidade: para algumas empresas, o objetivo principal pode ser contribuir para o bem-estar da sociedade e do meio ambiente, adotando práticas sustentáveis e responsáveis.
- Inovação: empresas que valorizam a inovação podem ter como objetivo principal desenvolver novas tecnologias, produtos ou processos que tragam benefícios significativos para seus clientes e para a sociedade como um todo.
- Excelência operacional: focar na eficiência e na melhoria contínua dos processos operacionais pode ser o objetivo principal de empresas que buscam reduzir custos, aumentar a produtividade e garantir a qualidade de seus produtos e serviços.

Para nosso caso, maximizar esse valor e reduzir os custos, então:

- Como reduzir os custos da empresa?

Reduzir os custos de uma empresa é uma tarefa desafiadora, para chegar a esse objetivo existem várias estratégias, dentre delas, a automação. Que consiste em utilizar a tecnologia para automatizar tarefas repetitivas e burocráticas. O que pode reduzir custos com mão de obra e aumentar a eficiência operacional, assim como manter um acompanhamento constante das finanças da empresa para identificar rapidamente qualquer desvio orçamentário e tomar medidas corretivas.

A justificativa desta pesquisa são os ganhos resultantes da implementação do sistema, que incluirá a melhoria do fluxo de dados de forma organizada, criando uma fonte de informação que permitirá o seguinte:

- O gerente terá acesso a um sistema de controle que oferece uma visão abrangente dos processos da empresa, juntamente com a capacidade de visualizar relatórios que servirão de base para algumas decisões.
- O estoquista receberá suporte para decidir quais novos produtos adquirir para reabastecer o estoque.
- Os vendedores terão suas vendas registradas de maneira digital, o que aumentará a eficiência, já que a verificação do estoque e a criação dos pedidos serão automatizadas.

## **2.3 FUNDAMENTAÇÃO TEÓRICA**

Encontraram-se algumas fontes que abordam aspectos referentes à criação de um Framework Web utilizando Python-Django, entre elas:

- Documentação oficial do Django, que fornece uma descrição abrangente do framework, seus recursos e benefícios. Pode-se encontrar informações sobre Mapeamento Objeto-Relacional. <https://docs.djangoproject.com/en/stable/>

- Livro “Django para iniciantes” por William S. Vincent, que fornece uma introdução prática ao desenvolvimento web com Django. Explora os conceitos básicos e avançados do Django, incluindo Mapeamento Objeto-Relacional. <https://djangoforbeginners.com/>

A criação de um Framework Web utilizando Python-Django envolve o desenvolvimento de uma estrutura robusta e flexível para a construção de aplicativos web. Python é uma linguagem de programação popular e Django é um framework de alto nível para desenvolvimento web em Python. Com Django, é possível criar rapidamente aplicações web escaláveis e seguras, aproveitando suas características como ORM (*Object-Relational Mapping*), administração automática e sistema de autenticação, por exemplo. O desenvolvimento com Django segue padrões de arquitetura que promovem a organização do código e a reutilização de componentes, resultando em maior produtividade e boa manutenção do projeto.

Encontraram-se, também, algumas fontes que abordam aspectos referentes a banco de dados SQL, entre elas:

- Documentação do MySQL, o qual é um dos sistemas mais populares de gerenciamento de banco de dados SQL. Sua documentação oficial fornece uma descrição abrangente dos conceitos básicos do SQL e como funciona um banco de dados relacional. <https://dev.mysql.com/doc/>
- Artigo "What is SQL (Structured Query Language)?" da Microsoft, esse artigo fornece uma introdução ao SQL, explicando seus fundamentos e como é usado para manipular dados em bancos de dados relacionais. <https://docs.microsoft.com/en-us/sql/>

Um banco de dados SQL é um sistema de gerenciamento de banco de dados relacional que utiliza a linguagem SQL (*Structured Query Language*) para armazenar, manipular e recuperar dados. SQL é uma linguagem padronizada utilizada para criar, consultar e gerenciar bancos de dados relacionais. Bancos de dados SQL são compostos por tabelas, que armazenam os dados em linhas e colunas, e suportam operações como inserção, atualização, exclusão e consulta de dados. Eles oferecem recursos como transações, controle de concorrência, integridade referencial e segurança de acesso para garantir a consistência e confiabilidade dos dados armazenados.

Encontraram-se, ainda, algumas fontes que abordam aspectos referentes à implementação via Docker, entre as quais:

- Documentação oficial do Docker, que oferece uma descrição detalhada dos conceitos básicos do Docker, incluindo o empacotamento de aplicativos em contêineres, distribuição e execução em diferentes ambientes. <https://docs.docker.com/>
- Tutoriais Python Docker do Real Python, que fornecem uma introdução prática ao uso do Docker com Python, explicando os conceitos fundamentais e como usar o Docker para empacotar e distribuir aplicativos Python. <https://realpython.com/tutorials/docker/>

A implementação via Docker é uma abordagem para desenvolver, distribuir e executar aplicativos de forma consistente e isolada em diferentes ambientes. Docker é uma plataforma de virtualização de contêineres que permite empacotar um aplicativo com todas as suas dependências em um contêiner, garantindo que ele funcione da mesma maneira em qualquer ambiente. Os contêineres Docker são leves, portáteis e autossuficientes, o que os torna ideais para facilitar o desenvolvimento, implantação e escalabilidade de aplicativos. Com o Docker, é possível criar imagens de contêineres que contêm todas as bibliotecas, frameworks e componentes necessários para executar um aplicativo específico, e, então, implantá-las em qualquer host que tenha o Docker instalado. Isso simplifica o processo de desenvolvimento, elimina problemas de compatibilidade e melhora a eficiência operacional.

Encontraram-se algumas fontes que abordam aspectos referentes à conformidade com o Open Source e seus princípios, entre elas:

- Site da Open Source Initiative (OSI). A OSI é uma organização dedicada a promover e proteger o código aberto. Seu site oferece informações detalhadas sobre as diferentes licenças de código aberto e os princípios da conformidade com o Open Source. <https://opensource.org/>
- Open Sources Guides da Linux Foundation, que fornecem uma visão geral dos princípios básicos de conformidade com o Open Source, incluindo a importância da compreensão das licenças de código aberto e as melhores práticas para garantir a conformidade. <https://www.linuxfoundation.org/resources/open-source-guides>

A conformidade com o Open Source refere-se ao cumprimento das licenças e diretrizes associadas ao uso, distribuição e modificação de software de código aberto. O código aberto é caracterizado pela transparência, colaboração e liberdade de acesso e modificação do código-fonte. Para garantir a conformidade com o Open Source, as organizações devem estar cientes das licenças de software de código aberto utilizadas em seus projetos e garantir que estejam em conformidade com seus termos. Isso pode incluir a atribuição adequada de créditos aos autores

originais, a divulgação de modificações feitas ao código-fonte e o respeito aos requisitos de licenciamento, como a distribuição do código-fonte de projetos derivados. A conformidade com o Open Source promove a transparência, o respeito pela comunidade de desenvolvimento de código aberto e a adoção responsável de tecnologias de código aberto.

Encontraram-se algumas fontes que abordam conceitos sobre o controle de versionamento no GitHub, entre elas:

- Documentação oficial do GitHub, que fornece uma visão detalhada de como usar o controle de versionamento e outros recursos da plataforma GitHub. <https://docs.github.com/en>
- Artigo "*Understanding the GitHub Flow*" do *GitHub Guides*, ele explica o fluxo de trabalho recomendado para colaborar em projetos no GitHub, incluindo o uso de ramificações, pull requests e merge. <https://guides.github.com/introduction/flow/>

O controle de versionamento no GitHub refere-se à prática de gerenciar e controlar as alterações feitas no código-fonte de um projeto de software usando a plataforma GitHub. O GitHub é um serviço de hospedagem de código-fonte baseado em Git, que é um sistema de controle de versão distribuído. Com o controle de versionamento no GitHub, os desenvolvedores podem acompanhar todas as mudanças feitas no código-fonte ao longo do tempo, facilitando o trabalho colaborativo, o gerenciamento de conflitos e a reversão de alterações indesejadas. Os desenvolvedores podem criar ramificações (*branches*) para desenvolver novos recursos ou corrigir bugs sem afetar o código principal, e depois mesclar (*merge*) essas alterações de volta ao ramo principal (*branch master*) quando estiverem prontas. Além disso, o GitHub fornece recursos como *issues* (problemas), *pull requests* (solicitações de alteração) e *wikis* para facilitar a colaboração e o gerenciamento de projetos de software.

## 2.4 METODOLOGIA

A metodologia adotada compreende a criação de um Framework Web utilizando Python-Django, Banco de Dados SQL, implementação vide Docker, em conformidade de Open Source e controle de versionamento no Github.

Como primeiro passo, criou-se um esboço básico do Modelo Entidade – Relacionamento (MER) para este sistema. Este modelo terá entidades que representam diferentes aspectos do sistema e relacionamentos que mostram como essas entidades conectam-se entre si:

- Produto.
- Fornecedor.
- Pedido.
- Cliente.
- Estoque – Matéria-prima.

Os relacionamentos entre essas entidades serão:

- Cada produto é fornecido por um fornecedor. O relacionamento é um-para-muitos entre fornecedor e produto.
- Cada pedido contém um ou mais produtos. O relacionamento é um-para -muitos entre pedido e produto.
- Cada pedido é feito por um cliente. O relacionamento é um-para-muitos entre pedido e cliente.
- Cada produto tem uma quantidade específica no estoque. O relacionamento é um-para-um entre produto e estoque.

Para realizar o desenvolvimento, são formuladas as tabelas do banco de dados e a estrutura de cada uma delas:

1. PRODUTO(ID, Nome, Preço)

- ID: identificador do produto para buscas no sistema, funciona como chave primária.
- Nome: nome do produto.
- Preço: preço, sempre atualizado, do produto.

PRODUTO	
<u>ID</u> (PK)	Integer(20)
Nome	Varchar(100)
Preço	Double(10)

*Tabela 1.- Tabela PRODUTO do banco de dados.*

2. FORNECEDOR(ID, Nome, Email, Telefone)

- ID: identificador do fornecedor para buscas no sistema, funciona como chave primária.
- Nome: nome do fornecedor.

- Email: para realizar o contato com o fornecedor caso seja necessário.
- Telefone: para realizar o contato com o fornecedor caso seja necessário.

FORNECEDOR	
<u>ID</u> (PK)	Integer(20)
Nome	Varchar(100)
Email	Varchar(100)
Telefone	Integer(20)

*Tabela 2.- Tabela FORNECEDOR do banco de dados.*

3. CLIENTE(ID, Nome, Email, Telefone)

- ID: identificador do cliente para buscas no sistema, funciona como chave primária.
- Nome: nome do cliente.
- Email: para realizar o contato com o cliente caso seja necessário.
- Telefone: para realizar o contato com o cliente caso seja necessário.

CLIENTE	
<u>ID</u> (PK)	Integer(20)
Nome	Varchar(100)
Email	Varchar(100)
Telefone	Integer(20)

*Tabela 3.- Tabela CLIENTE do banco de dados.*

4. PEDIDO(ID, Data, Total, Cliente\_ID)

- ID: identificador do pedido para buscas no sistema, funciona como chave primária.
- Data: data do pedido.
- Total: total do pedido.
- Cliente\_ID: para quem foi realizado o pedido, funciona como chave estrangeira.

PEDIDO	
ID (PK)	Integer(20)
Data	Date



Total	Varchar(100)
Cliente_ID (FK)	Integer(20)

Tabela 4.- Tabela PEDIDO do banco de dados.

#### 5. ESTOQUE(Produto\_ID, Quantidade)

- Produto\_ID: identificador do produto para buscas no sistema, funciona como chave primária e chave estrangeira.
- Quantidade: quantidade no estoque daquele produto.

ESTOQUE	
Produto_ID (PK, FK)	Integer(20)
Quantidade	Integer(20)

Tabela 5.- Tabela ESTOQUE do banco de dados.

A BV Wood forneceu os seguintes dados, transformados inicialmente em planilhas. Na Figura 2 estão as descrições das matérias-primas que são usadas para a fabricação de determinados produtos. Posteriormente, eles serão relacionados Materia\_prima <=> Produto.

	A	B	C	D	E	F
1	ESTOQUE - MATERIA PRIMA					
2	ID	Descrição	Dimensões		Valor	
3			Largura	Comprimento	Peça	m <sup>2</sup>
4	Pai0001	Painel Pinus 1,8 FL	2,50	1,00	R\$ 138,00	R\$ 55,20
5	Pai0002	Painel Pinus 1,5 FL	2,50	1,00	R\$ 120,00	R\$ 48,00
6	Pai0003	Painel AL 1,8	2,50	1,18	R\$ 180,00	R\$ 61,02
7	Pai0004	Painel AL 1,5	2,50	1,18	R\$ 165,00	R\$ 55,93
8	Pai0005	Painel JO 1.8	2,00	1,00	R\$ 140,00	R\$ 70,00
9	Pai0006	Painel JO 1,5	2,00	1,00	R\$ 120,00	R\$ 60,00
10	Pai0007	Painel AL 2,5	2,00	1,00	R\$ 230,00	R\$ 115,00
11	Pai0008	Painel AL 3,0	2,30	1,20	R\$ 310,00	R\$ 112,32
12	Pai0009	Painel BR 3,5	2,00	1,00	R\$ 250,00	R\$ 125,00
13	Cai0001	Caibro 5x5 (BERI)	3,00	0,05	R\$ 21,30	R\$ 142,00
14	Cai0002	Caibro 7x7	3,00	0,07	R\$ 39,00	R\$ 185,71
15	Cai0003	Caibro 8x8	3,00	0,08	R\$ 34,12	R\$ 142,17
16	Tab0001	Tabua 10mm	3,00	0,15	R\$ 11,70	R\$ 26,00
17	Tab0002	Tabua 1,5	3,00	0,15	R\$ 15,00	R\$ 33,33
18	Tab0003	Tabua 2,0	3,00	0,20	R\$ 25,95	R\$ 43,25
19	Tab0004	Tabua 3,5	3,00	0,20	R\$ 39,00	R\$ 65,00
20						

Figura 2.- Planilha original das matérias-primas.

As Figura 3 e Figura 4 mostra a planilha com as descrições dos produtos fabricados pela BV Wood.

	A	B	C	D	E	F	G	H	I	J	K	
1	CHACHEPÔ											
2	DESCRIÇÃO	TIPO MADEIRA	Comprimento (M)	Altura (M)	M²	VALOR M²	CUSTO DE MADEIRA	P.V. MADEIRA (4x)	Fundo	fundoTotal x4	Custo MDF	P
3	Cachepô H 8x8x8	Tabua Fina	0,32	0,1	0,032	R\$ 30,00	R\$ 0,96	R\$ 3,84	R\$ 0,10	R\$ 0,40	R\$ -	R
4	Cachepô Cílios P 8x8x8	Tabua Fina	0,32	0,1	0,032	R\$ 30,00	R\$ 0,96	R\$ 3,84	R\$ 0,10	R\$ 0,40	R\$ 0,20	R
5	Cachepô Cílios G 8x8x10	Tabua Fina	0,32	0,1	0,032	R\$ 30,00	R\$ 0,96	R\$ 3,84	R\$ 0,10	R\$ 0,40	R\$ 0,20	R
6	tampa	Tabua Fina	0,1	0,1	0,01	R\$ 30,00	R\$ 0,30	R\$ 1,20	R\$ 0,10	R\$ 0,40	R\$ 0,20	R
7	Caixa Organizadora Cílios 10x30x10	Tabua Fina	0,8	0,1	0,08	R\$ 30,00	R\$ 2,40	R\$ 9,60	R\$ 0,30	R\$ 1,20	R\$ 0,50	R
8	Porta Talher	Tabua Fina	0,5	0,12	0,06	R\$ 30,00	R\$ 1,80	R\$ 7,20	R\$ 0,15	R\$ 0,60	R\$ 1,00	R
9	12x12x8	Tabua Fina	0,5	0,1	0,05	R\$ 30,00	R\$ 1,50	R\$ 6,00	R\$ 0,15	R\$ 0,60	R\$ -	R
10	12x12x16	Tabua Fina	0,5	0,16	0,08	R\$ 30,00	R\$ 2,40	R\$ 9,60	R\$ 0,15	R\$ 0,60	R\$ -	R
11	12x12x24	Tabua Fina	0,5	0,25	0,125	R\$ 30,00	R\$ 3,75	R\$ 15,00	R\$ 0,15	R\$ 0,60	R\$ -	R
12	Conico P	Tabua Fina	0,5	0,12	0,06	R\$ 30,00	R\$ 1,80	R\$ 7,20	R\$ 0,15	R\$ 0,60	R\$ -	R
13	Conico G	Tabua Fina	0,6	0,14	0,084	R\$ 30,00	R\$ 2,52	R\$ 10,08	R\$ 0,20	R\$ 0,80	R\$ -	R
14	Conico com aba	Tabua Fina	0,8	0,12	0,096	R\$ 30,00	R\$ 2,88	R\$ 11,52	R\$ 0,25	R\$ 1,00	R\$ -	R
15	quadrado com aba	Tabua Fina	0,8	0,12	0,096	R\$ 30,00	R\$ 2,88	R\$ 11,52	R\$ 0,25	R\$ 1,00	R\$ -	R
16								R\$ -		R\$ -		
17	Floreiras	Mad	Compr.	Larg	M²	Total	x4	Outras Mad	Total Mad x4	Outros		
18	Penca 6 Coração	Tabua Fina	0,1	0,6	0,06	R\$ 30,00	R\$ 1,80	R\$ 7,20	R\$ -	R\$ 2,00	R	
19	Penca 4 Coração	Tabua Fina	0,1	0,4	0,04	R\$ 30,00	R\$ 1,20	R\$ 4,80	R\$ -	R\$ 1,00	R	
20	Lanterna P	Tabua 2.0	0,18	0,36	0,0648	R\$ 40,00	R\$ 2,59	R\$ 10,37	R\$ 1,00	R\$ 4,00	R\$ 1,00	R
21	Lanterna G	Tabua 2.0	0,18	0,36	0,0648	R\$ 40,00	R\$ 2,59	R\$ 10,37	R\$ 1,20	R\$ 4,80	R\$ 1,00	R
22	Luminária tripé de mesa	Painel	0,5	0,15	0,075	R\$ 55,00	R\$ 4,13	R\$ 16,50	R\$ 1,00	R\$ 4,00	R\$ 6,00	R
23	Luminária tripé de chão	Painel	1,3	0,15	0,195	R\$ 55,00	R\$ 10,73	R\$ 42,90	R\$ 1,50	R\$ 6,00	R\$ 6,00	R
24												

Figura 3.- Planilha dos produtos.

	A	B	C	D	E	F	G	H	I	J	K	L
1	Linha Gourmet											
2	DESCRIÇÃO	TIPO MADEIRA	Comprimento (M)	Altura (M)	M²	VALOR M²	CUSTO DE MADEIRA	P.V. MADEIRA (4x)	Tercerização	P.V TERC	CORTE	MONTAGE
3	Trio Tabua coração	Painel 1.8	0,16	0,6	0,096	R\$ 55,00	R\$ 5,28	R\$ 21,12	R\$ 2,00	R\$ 4,00	0	0
4	Trio Recorte coração	Painel 1.8	0,16	0,6	0,096	R\$ 55,00	R\$ 5,28	R\$ 21,12	R\$ 2,00	R\$ 4,00	0	0
5	Tabua love P	Painel 1.8	0,28	0,12	0,034	R\$ 55,00	R\$ 1,85	R\$ 7,39	R\$ 0,80	R\$ 1,60	0	0
6	Tabua love G	Painel 1.8	0,36	0,12	0,043	R\$ 55,00	R\$ 2,38	R\$ 9,50	R\$ 1,00	R\$ 2,00	0	0
7	Tabua love GG	Painel 1.8	0,46	0,12	0,055	R\$ 55,00	R\$ 3,04	R\$ 12,14	R\$ 1,20	R\$ 2,40	0	0
8	Tabua Afeto	Painel 1.8	0,3	0,22	0,066	R\$ 55,00	R\$ 3,63	R\$ 14,52	R\$ 1,00	R\$ 2,00	0	0
9	Porta copo Clean	Tabua fina	0,1	0,1	0,01	R\$ 30,00	R\$ 0,30	R\$ 1,20		R\$ -	0,2	0
10	Porta papel Clean	Tabua fina	0,1	0,24	0,024	R\$ 30,00	R\$ 0,72	R\$ 2,88		R\$ -	0,5	0,2
11	Porta Guardanapo Clean	Tabua fina	0,1	0,1	0,01	R\$ 30,00	R\$ 0,30	R\$ 1,20		R\$ -	0,2	0,1
12	Colher de pau coração	Painel 1.8	0,3	0,06	0,018	R\$ 55,00	R\$ 0,99	R\$ 3,96	R\$ 0,80	R\$ 1,60		
13	Escada Pano de prato	Tabua fina	1,7	0,02	0,034	R\$ 50,00	R\$ 1,70	R\$ 6,80		R\$ -	0,8	0,2
14												

Figura 4.- Planilha dos produtos.

Com esses dados serão realizadas as relações entre produto e matéria-prima.

## **2.5 RESULTADOS PRELIMINARES: SOLUÇÃO INICIAL**

Na busca por soluções eficazes para os desafios enfrentados pelas empresas modernas, o desenvolvimento de um framework adequado torna-se crucial.

O primeiro passo rumo à criação do framework consistiu na preparação do ambiente de desenvolvimento. Após cuidadosa análise das necessidades e dos requisitos do projeto, optou-se pela utilização do framework Python-Django, reconhecido por sua robustez, flexibilidade e ampla gama de recursos. A escolha desta tecnologia proporcionou um ambiente adequado para o desenvolvimento ágil e eficiente do framework.

Com o ambiente de desenvolvimento devidamente configurado, deu-se início à implementação do acesso ao framework, juntamente com a catalogação de tipos de usuário.

Visando garantir a rastreabilidade e o controle das alterações realizadas ao longo do processo de desenvolvimento, foi integrado ao framework um sistema de versionamento utilizando o GitHub, para que seja possível manter um histórico organizado e acessível de todas as modificações efetuadas no projeto.

Ao longo do processo de desenvolvimento, houve obstáculos que impactaram diretamente o ritmo e a eficiência do projeto, impedindo a evolução com o dinamismo desejado. Devido a conflitos de horário e sobrecarga de trabalho, houve atraso no acesso a alguns dados. Consequentemente, a criação de banco de dados para a implementação do framework ocorreu fora do tempo planejado. Contudo, destaca-se que a questão de sobrecarga será melhorada com a implementação da automatização do controle.

Em síntese, o desenvolvimento do framework para gestão de acesso e catalogação de usuários representa um passo fundamental na busca por soluções eficazes para os desafios enfrentados pela empresa. É essencial que a empresa esteja preparada para superar os desafios e adaptar-se às mudanças do ambiente, a fim de garantir o sucesso contínuo do framework e sua capacidade de atender às demandas em constante evolução do mercado.

The screenshot displays the Django administration interface. At the top, the browser address bar shows the URL `127.0.0.1:8000/admin/login/?next=/admin/`. The main heading is "Django administration". Below this, the login form contains a "Username:" field with the value "univesp" and a "Password:" field with masked characters ".....". A "Log in" button is positioned below the password field.

Below the login form, there is a table listing users. The table has columns for "USERNAME", "EMAIL ADDRESS", "FIRST NAME", "LAST NAME", and "STAFF STATUS". The "univesp" user is highlighted with a green checkmark in the "STAFF STATUS" column, indicating they are a staff user. The other users have a red 'x' in the "STAFF STATUS" column, indicating they are not staff users.

	USERNAME	EMAIL ADDRESS	FIRST NAME	LAST NAME	STAFF STATUS
<input type="checkbox"/>	Roddy@gmail.com	Roddy@gmail.com			✗
<input type="checkbox"/>	bernardo@gmail.com	bernardo@gmail.com			✗
<input type="checkbox"/>	bruna@empresa.com	bruna@empresa.com			✗
<input type="checkbox"/>	funcionario@gmail.com	funcionario@gmail.com			✗
<input type="checkbox"/>	funcionario_2@empresa.com	funcionario_2@empresa.com			✗
<input type="checkbox"/>	paulo@gmail.com	paulo@gmail.com			✗
<input checked="" type="checkbox"/>	univesp				✓

7 users

**Figura 5.-** Tela de acesso do superuser para manutenção dos logins de acesso. Fonte: autor (2024).

Change user

Roddy@gmail.com

HISTORY

Username:

Roddy@gmail.com

Required. 150 characters or fewer. Letters, digits and @/./+/-/\_ only.

Password:

algorithm: pbkdf2\_sha256 iterations: 720000 salt: vSWiWb\*\*\*\*\* hash: 11QK9y\*\*\*\*\*

Raw passwords are not stored, so there is no way to see this user's password, but you can change the password using this form.

Personal info

First name:

Roddy

Last name:

Ramos

Email address:

Roddy@gmail.com

User permissions:

Available user permissions ?

Q Filter

Administration | log entry | Can add log entry

Administration | log entry | Can change log entry

Administration | log entry | Can delete log entry

Administration | log entry | Can view log entry

Authentication and Authorization | group | Can add g

Authentication and Authorization | group | Can chang

Authentication and Authorization | group | Can delete

Authentication and Authorization | group | Can view g

Authentication and Authorization | permission | Can a

Authentication and Authorization | permission | Can c

Authentication and Authorization | permission | Can d

Authentication and Authorization | permission | Can v

Choose all

Chosen user permissions ?

Q Filter

Usuarios | user | Realizar Venda

Remove all

Specific permissions for this user. Hold down "Control", or "Command" on a Mac, to select more than one.

Permissions

☒ Active

Designates whether this user should be treated as active. Unselect this instead of deleting accounts.

☐ Staff status

Designates whether the user can log into this admin site.

☐ Superuser status

Designates that this user has all permissions without explicitly assigning them.

Groups:

Available groups ?

Q Filter

Choose all

Chosen groups ?

Q Filter

vendedor

Remove all

+

Cargo

Cargo:

Vendedor

SAVE

Save and add another

Save and continue editing

Delete

Figura 6.- Configuração do Usuário e sua Permissões. Fonte: autor (2024).

```

from django.http import HttpResponseRedirect
from django.shortcuts import render
from rolepermissions.decorators import has_permission_decorator
from .models import Users
from django.shortcuts import redirect
from django.urls import reverse
from django.contrib import auth

@has_permission_decorator('cadastrar_vendedor')
def cadastrar_vendedor(request):
    if request.method == "GET":
        vendedores = Users.objects.filter(cargo="V")
        return render(request, 'cadastrar_vendedor.html')
    if request.method == 'POST':
        email = request.POST.get('email')
        senha = request.POST.get('senha')

        user = Users.objects.filter(email=email)

        if user.exists():
            # TODO: Utilizar messages do Django
            return HttpResponseRedirect('Email Existente')

        user = Users.objects.create_user(username=email,
                                          email=email,
                                          password=senha,
                                          cargo="V")

        # TODO: Redirecionar com uma mensagem
        return HttpResponseRedirect('Conta Criada')

def login(request):
    if request.method == "GET":
        if request.user.is_authenticated:
            return redirect(reverse('plataforma'))
        return render(request, 'login.html')

    elif request.method == 'POST':
        login = request.POST.get('email')
        senha = request.POST.get('senha')

        user = auth.authenticate(username=login, password=senha)

        if not user:
            # TODO: Redirecionar com mensagem de erro
            return HttpResponseRedirect('Usuario inválido')

        auth.login(request, user)
        return HttpResponseRedirect('Usuario logado com sucesso')

```

**Figura 7.-** Scripts de condicionais de acesso por usuário. Fonte: autor (2024).

The screenshot shows a GitHub repository page for a user named 'belima93'. The repository is titled 'UNIVESP-PI\_DRP04-S\_005-G\_006' and is marked as 'Private'. The interface includes a top navigation bar with links to Code, Issues, Pull requests, Actions, Projects, Security, Insights, and Settings. Below the repository name, there's a section for branches (main) and tags (0). A file explorer shows the repository structure: Docs, Project, .gitignore, and README.md. The README file is selected, displaying the project description: 'Projeto Integrador - UNIVESP: Desenvolvimento de um software com framework web que utilize noções de banco de dados, praticando controle de versão.' It also lists the group members: Bernardo Lima dos Santos, Bruna de Oliveira, Paulo Sergio da Silva Machado, and Roddy E. Ramos Gonzáles. On the right side, there are sections for About (Projeto Integrador - UNIVESP), Releases (No releases published), Packages (No packages published), and Languages (Python 84.6%, HTML 15.4%).

belima93 / UNIVESP-PI\_DRP04-S\_005-G\_006

Code Issues Pull requests Actions Projects Security Insights Settings

UNIVESP-PI\_DRP04-S\_005-G\_006 Private

main 3 Branches 0 Tags

Go to file Add file Code

belima93 feat: Iniciado a criação do Framework com Django, configuração dos ... dc7516f - yesterday 10 Commits

Docs docs: Adicionado o Plano Ação last week

Project feat: Iniciado a criação do Framework com Django, confi... yesterday

.gitignore Initial commit last week

README.md docs: Inclusão dos pré requisitos e criação do ambiente v... 5 days ago

README

## UNIVESP-PI\_DRP04-S\_005-G\_006

Projeto Integrador - UNIVESP: Desenvolvimento de um software com framework web que utilize noções de banco de dados, praticando controle de versão.

Integrantes do Grupo:

- Bernardo Lima dos Santos, RA 1811895
- Bruna de Oliveira, RA 2211351
- Paulo Sergio da Silva Machado, RA 2213937
- Roddy E. Ramos Gonzáles, RA 2211457

About

Projeto Integrador - UNIVESP

Readme

Activity

0 stars

1 watching

0 forks

Releases

No releases published

[Create a new release](#)

Packages

No packages published

[Publish your first package](#)

Languages

Python 84.6% HTML 15.4%

**Figura 8.-** Repositório do Github para controle de versionamento. Fonte: autor (2024).

## REFERÊNCIAS

Documentação oficial do Django, <https://docs.djangoproject.com/en/stable/>

William S. Vincent, **Django para iniciantes**, <https://djangoforbeginners.com/>

Documentação do MySQL, <https://dev.mysql.com/doc/>

Artigo "What is SQL (Structured Query Language)?" da Microsoft, <https://docs.microsoft.com/en-us/sql/>

Documentação oficial do Docker, <https://docs.docker.com/>

Tutoriais Python Docker do Real Python, <https://realpython.com/tutorials/docker/>

Site da Open Source Initiative (OSI), <https://opensource.org/>

Open Sources Guides da Linux Foundation, <https://www.linuxfoundation.org/resources/open-source-guides>

Documentação oficial do GitHub, <https://docs.github.com/en>

Artigo "*Understanding the GitHub Flow*" do *GitHub Guides*, <https://guides.github.com/introduction/flow/>



**ANEXOS (OPCIONAL)**

## **APÊNDICES (OPCIONAL)**