

JL Control: Sistema de gestão de locações

Augusto de Lima Mendonça, Caike Colantonio, Robson Leonardo Lino, Victor Hugo Tarriga Gomes¹

Orientador: Professor Esp. Victor Williams Stafusa da Silva

Faculdade Impacta de Tecnologia
São Paulo, SP, Brasil
11 de junho de 2021

Resumo. Com a necessidade de acompanhar o crescimento imposto pelo mercado as empresas, ações que diminuam erros e aumentem os lucros é o foco da nova geração. Isso somente é possível utilizando tecnologias que auxiliem nesses processos decisórios. Esse projeto tem o objetivo de entregar um *software* de controle de estoque para o gerenciamento do setor de locação.

Palavras-chaves: Locação, Gestão, Estoque.

1. Introdução

A empresa JLima Collection, pioneira no mercado de locação de trajes a rigor em São Paulo, agora também será pioneira na atualização de seu modelo de trabalho e controle com esse novo projeto. Serão desenvolvidos também relatórios e *dashboards* que auxiliarão o corpo diretivo na tomada de decisões, especificamente para o setor de locação da empresa. Como o foco da empresa é o aumento nos lucros e o controle da movimentação de suas locações a mesma não dispõe de recursos tecnológicos para essa finalidade, foi solicitada uma proposta que utiliza de uma ferramenta de baixo custo e com menor tempo possível para entrega do projeto.

Ao término deste projeto será entregue uma solução *web* interativa e de fácil utilização para o negócio, onde a interface será construída com a ferramenta Django.

1.1. A JLima Collection

Fundada em meados de 1980 por José Maria de Lima e Luiza Aparecida Muchiutti e com ajuda de cinco filhos adolescentes, a JLima Collection, iniciou suas atividades comerciais focados no ramo de locação de vestuários cerimoniais, num edifício de médio porte, no bairro da Luz, centro de São Paulo, na rua São Caetano, nº 75, próximo à Estação da Luz do CPTM. Pouco tempo depois mais lojas foram abertas e mudaram a visão da Rua São Caetano, em referência a moda e apetrechos para casamento, tornando assim a tradicional “Rua Das Noivas” de São Paulo. Anos depois, os irmãos de José Maria de Lima também entraram para o ramo de locação de trajes a rigor. Por sua vez, residentes na cidade de Santo André, também tradicionalizaram a Rua Luís Pinto Fláquer com produtos e serviços focados para eventos e cerimônias. No final dos anos 1990, os filhos mais velhos de José Maria ampliam o número da família com três novas lojas no centro de São Paulo. Após alguns anos desde sua chegada ao ramo de alfaiataria, com o crescimento de mercado e com a empresa

¹ Os autores podem ser contatados respectivamente pelos seus correios eletrônicos: guto_limam@hotmail.com, caikecolantonio@hotmail.com, robson.leonardo.lino@hotmail.com, victor.tarriga@gmail.com.

cada vez mais sólida, tendo em vista a ampliação da loja, a mesma foi expandida para o prédio do lado (nº 69), onde se localiza até os dias atuais.

Devido ao constante crescimento da empresa e a abertura de filiais o controle gerencial de locações tornou-se descontrolado, gerando assim desconforto para os gestores de loja, em referência ao controle de estoque não apenas a unidade matriz assim como em suas filiais, gerando a necessidades de integração, das informações de todas as lojas. A maior preocupação da entidade é com seus clientes e a não padronização da gestão de locação, afeta não somente a empresa como todos os atores envolvidos em seus processos. Com foco em aprimorar a qualidade e a otimização de serviços, a JLima Collection, observou a necessidade de ampliar e modernizar seu sistema de gestão de locação. Com isso contatou um grupo de estudantes de análise e desenvolvimento de sistemas para desenvolver seu novo sistema de controle de gestão.

Em uma das reuniões iniciais do projeto, ficou decidido que o sistema integraria todas as lojas por serviço *web*. O que deixaria o custo mais barato, pensando em contraponto a uma solução de *software* local, o que exigiria a troca das máquinas para modelos mais atuais. Esta decisão foi unânime após o ocorrido referente ao COVID-19 no ano de 2020, assim gestores poderiam consultar seus relatórios online trabalhando em *home office*.

1.2. O problema da gestão de locações da JLima Collection

O problema de gestão de locação afeta a empresa devido à falta de controle na movimentação das entradas e saídas de trajes locados, informações incompletas de cada peça em estoque; possibilidade de perda de peças locadas; e consequentes prejuízos financeiros.

Os benefícios deste novo Sistema de gestão, JLControl, são:

- Acompanhar o processo de ponta a ponta, com informações em tempo real;
- Organização de estoques por lojas;
- Identificação de trajes locados e em estoque;
- Controle das datas de ajustes, retirada e devolução;
- Agilidade na elaboração de orçamento para clientes;

1.3. Stakeholders e restrições

Quadro 1 – Stakeholders e restrições.

Usuários	Descrição
Atendentes de loja	<ul style="list-style-type: none">● Consultar trajes locados e em estoque.● Orçar locações.● Elaborar, avaliar e emitir contratos de locação.● Controlar a movimentação de trajes na oficina.● Controlar datas de retirada e devolução de trajes.
Atendente de caixa	<ul style="list-style-type: none">● Consultar contratos de locação.● Consultar as datas de retirada e devolução de trajes.
Gestores	<ul style="list-style-type: none">● Consultar e emitir relatórios gerenciais de locação.● Cadastrar novas peças e trajes.

Fonte: Os autores.

2. Solução proposta

O objetivo deste projeto é implementação de uma solução *web* através de um *Data Mart* para desenvolvimento de *dashboards* de forma automatizada com o propósito de auxiliar o setor de vendas para que possam identificar as locações e que possam possibilitar melhores estratégias de locações e aumento de organização.

A implantação do projeto na empresa é fundamental para facilitar a análise dos seus resultados e aumentar a competitividade no mercado, visando facilitar a visualização dos dados através das *dashboards* detalhados e dinâmicos será possível tomar decisões mais assertivas e forma mais rápida que podem auxiliar na estratégia de locações.

2.1. Requisitos do sistema

Quadro 2 – Requisitos de sistema.

ID	Descrição
SSS01	O sistema deve permitir seu uso em qualquer <i>desktop</i> com conexão à <i>internet</i> .
SSS02	O sistema deve permitir o cadastro de usuários.
SSS03	O sistema deve permitir a alteração de dados do usuário.
SSS04	O sistema deve permitir a exclusão de dados do usuário.
SSS05	O sistema deve permitir o usuário fazer <i>login</i> com usuário e senha.
SSS06	O sistema deve permitir que o usuário peça uma nova senha.
SSS07	O sistema deve permitir que o gestor adicione trajes a rigor no estoque.
SSS08	O sistema deve permitir que o gestor altere trajes a rigor no estoque.
SSS09	O sistema deve permitir que o gestor inative trajes a rigor no estoque.
SSS10	O sistema deve permitir que o atendente consulte trajes a rigor locados e em estoque.
SSS11	O sistema deve permitir que o atendente consulte datas de retirada e devolução de trajes a rigor.
SSS12	O sistema deve permitir que o atendente consulte o andamento dos ajustes na costura.
SSS13	O sistema deve permitir que o atendente reserve os trajes a rigor escolhidos.

SSS14	O sistema deve permitir que o atendente alugue os trajes a rigor escolhidos.
SSS15	O sistema deve permitir que o atendente cancele a locação de trajes a rigor.
SSS16	O sistema deve permitir que o atendente consulte a ficha de medidas.
SSS17	O sistema deve permitir que o atendente preencha a ficha de medidas.
SSS18	O sistema deve permitir que o atendente altere a ficha de medidas.
SSS19	O sistema deve permitir que o atendente consulte o contrato de locação.
SSS20	O sistema deve permitir que o atendente cadastre o cliente da locação.
SSS21	O sistema deve permitir que o atendente altere o cliente da locação.
SSS22	O sistema deve permitir que o atendente exclua o cliente da locação.
SSS23	O sistema deve permitir que o gestor emita relatórios completos de trajes a rigores locados e em estoque.
SSS24	O sistema deve permitir que o gestor emita os relatórios completos de trajes a rigor por nome, tipo, modelo, corte, mais locados, menos locados, valores, quantidade de locações feitas por dia, mês e ano.
SSS25	O sistema deve atualizar os relatórios conforme novos dados sejam inclusos.
SSS26	O sistema deve permitir que o gestor possa acessar e emitir o resumo das locações em movimento, organizadas por data.
SSS27	O sistema deve permitir que o gestor possa acessar e emitir a relação das devoluções pendentes, organizadas por data.
SSS28	O sistema deve permitir o envio de avisos para clientes via <i>e-mail</i> .
SSS29	O sistema deve armazenar tais avisos contendo como histórico a data do envio, conteúdo e usuário que realizou o envio, bem como o remetente.

Fonte: Os autores.

2.2. Regras de negócio

Quadro 3 – Regras de negócio.

Requisito	Descrição
RN 01	A locação deverá ser entregue pelo cliente no prazo máximo de uma semana.

RN 02	O prazo de ajustes do traje a rigor é de três dias da data do pedido.
RN 03	O prazo para retirada do traje a rigor é de cinco dias da data do pedido.
RN 04	O prazo para o recebimento do <i>feedback</i> é de um dia data da retirada do traje.
RN 05	O prazo para os ajustes emergenciais é de uma hora.
RN 06	O prazo para nova solicitação é imediato.
RN 07	Será gerada uma multa de R\$ 100,00 por peça devolvida em más condições.

Fonte: Os autores.

2.3. Descrição das funcionalidades

Abaixo serão apresentadas as funcionalidades das telas do sistema JLControl, seguindo um passo a passo desde o *login*, Menu Principal, Preenchimento de Locação, Ajustes/Costura, Relatórios de Gestão, Pesquisa de Locações, Pesquisa de Trajes, Admin trajes, Admin Usuários.

2.3.1. Login

Aqui é realizado o *login* do usuário, que pode ou não ser administrador do sistema, colocando o usuário e senha, caso esteja correto, a tela redirecionará para a tela principal do sistema, caso esteja errado vai mostrar uma mensagem de usuário e senha incorreta, nessa tela é possível clicar no botão esqueci minha senha que será redirecionada para a tela do Django admin.

Figura 1 – Menu de *login*.

Fonte: Os autores.

2.3.2. Menu principal

Após efetuar o *login* via tela de *login* todas as telas vão ter o menu que possui três opções *dropdown*, que vão levar para as telas do sistema, é dividida em três sessões, Sistema, Pesquisar e Admin, ao clicar Em Sistema > Locar vai para a tela de locação, ao Sistema > Costura vai para a tela de costura, ao clicar em Pesquisar > Pesquisar Locações abre o modal

para pesquisar locações e ao clicar no em Pesquisar > Pesquisar Trajes abre o *modal* para pesquisar trajes, ao clicar Admin > Trajes abre o portal do Admin do Django para cadastrar novos trajes, ao clicar no Admin > Usuários abre o portal do Admin do Django para cadastrar novos usuários, o último botão da esquerda é o botão de “Sair”, que clicando no botão é efetuado o *logout* do sistema e é redirecionado para a tela de *login*.

Figura 2 – Menu principal.



Fonte: Os autores.

2.3.3. Preenchimento de locação

A tela principal e mais importante do JLControl, nessa tela é possível encontrar o cliente que está realizando a locação clicando no botão “Encontrar cliente”, quando é encontrado preenche os *inputbox* automaticamente com os dados, caso não é encontrado ou é necessário criar um novo. No canto superior esquerdo dos dados de cliente, existe um *checkbox* que indica se o cliente é estrangeiro ou não, ao marcar, oculta o CPF e RG e aparece documento externo. O atendente insere os dados nos *inputbox* que ao realizar o salvamento da locação, o cliente é criado.

Abaixo dos dados de cliente, temos dois campos para realizar a pesquisa do traje para inserir na locação, é possível fazer pesquisa por dois dados do traje, o código do traje e o modelo do traje, ambos possuem *autocomplete* para facilitar a busca e essa busca vai trazer somente trajes disponíveis, ou seja, somente que já não está alocado e “ativo” no banco de dados, quando é clicado no botão “Adicionar” o traje que realizou a busca que vai para a tabela que está logo abaixo, clicando na linha do traje adicionado na tabela, o mesmo é selecionado e ao clicar no botão “Remover” ele é removido da tabela. Também é possível marcar que o traje adicionado se precisar de ajuste ou não, clicando no botão “Não precisa”, a escrita muda para “Precisa” e aparece um botão “Medidas”, ao clicar nesse botão, abre o *modal* para criar a ficha de medidas.

Abaixo da tabela dos trajes adicionados, temos o *inputbox* da previsão de devolução, este input é um que seleciona a data que está previsto a devolução da locação.

O último botão da tela, que está na parte inferior é o botão Locar, esse botão faz uma série de verificações para certificar que o usuário não colocou algo de errado (CPF inválido ou algum campo faltante) ou que quebre alguma regra de negócio, passando por todas as verificações, realizará o salvamento da locação.

Figura 3 – Preenchimento de locação.

JL Control Sistema * Pesquisar * Admin *

☐ Cliente Estrangeiro Encontrar cliente

CPF: 907.745.730-58 RG: 28.620.019-1

Nome: Robson Leonardo Telefone: (11)98544-4212

Endereço: Rua das OPEs Email: robson@robson.com

Código: Modelo: Adicionar Remover

Código	Nome	Modelo	Corte	Valor	Precisa ajuste?
TRAJE2	Traje de gala	Smoking Americano	Australiano	300.00	Não precisa

Total: R\$300

Previsão de Devolução: 01/04/2021 21:03 Local

Fonte: Os autores.

2.3.4. Ajustes/costura

A tela costura é a mais importante e mais utilizada para a consulta da equipe de ajustes. Nessa tela é possível identificar quais são os trajes que aguardam os ajustes, é possível abrir o *modal* para consultar as medidas clicando no botão “Abrir Ficha de Medidas” e ao término dos ajustes é possível realizar a finalização clicando no botão “Finalizar Ajuste”. Assim deixando o traje pronto para ser retirado pelo cliente.

Figura 4 – Ajustes/Costura.

#	Status	Código	Modelo	Cliente	Ações
1	Aguardando	TRAJE2	Smoking Americano	Robson Leonardo	Abrir ficha de medidas Finalizar ajustes

Total: 1

Fonte: Os autores.

2.3.5. Relatórios de gestão

O sistema possui quatro relatórios, “Menos/Mais alocados”, esse relatório retorna a quantidade que cada traje foi alocado, “Relatório de locações feitas no intervalo de datas:”, esse relatório recebe como parâmetro de entrada duas datas e aqui mostra a quantidade de locações separadas por “Total de Locações”, “Locações Atrasadas”, “Locações Alocadas”, “Locações Devolvidas” e “Locações canceladas”, o “Relatório por pesquisa de traje” recebe como entrada alguma informação do traje e pesquisa no banco todos os trajes que tem essa informação em comum, por exemplo, pesquisando “*Smoking Americano*” no modelo, vai mostrar todos os trajes que possui esse modelo, “Relatório financeiro nos intervalos de datas:” recebe como parâmetro um intervalo de data e mostra os dados de valores financeiros nesse período.

Figura 5 – Relatórios de gestão.

The dashboard titled 'JL Control' features a top navigation bar with 'Sistema', 'Pesquisar', and 'Admin' links. It contains four report cards:

- Menos/Mais Alocados:** A card with a title, a description, and a 'Mostrar' button.
- Relatório de locações feitas no intervalo de datas:** A card with a title, a description, and two date input fields (Data Inicial and Data Final) with a 'Pesquisar' button.
- Relatório por pesquisa de traje:** A card with a title and four input fields (Código, Nome, Modelo, Corte) with a 'Pesquisar' button.
- Relatório financeiro nos intervalos de datas:** A card with a title, a description, and two date input fields (Data Inicial and Data Final) with a 'Pesquisar' button.

Fonte: Os autores.

2.3.6. Pesquisar locações

O *modal* de pesquisa de locações possui quatro formas de pesquisa com dados de cliente, podemos pesquisar por CPF, Telefone, Nome e por Documento Externo, o campo Nome possui *autocomplete* e é possível realizar uma busca avançada por data e status de locação. Alterando entre os botões Simples e Avançado, os *inputbox* abaixo mudarão o tipo de acordo com o tipo de pesquisa.

Figura 6 – Pesquisar locações avançada.

The 'Pesquisar locações com:' modal has a close button (X) and two tabs: 'Simples' and 'Avançado'. The 'Avançado' tab is active, showing the following fields:

- Data Inicial:** Input field with placeholder 'dd/mm/aaaa --:--' and a calendar icon.
- Data Final:** Input field with placeholder 'dd/mm/aaaa --:--' and a calendar icon.
- Status:** Dropdown menu with 'Todos' selected.
- Pesquisar:** A large blue button.
- Fechar:** A large grey button.

Fonte: Os autores.

Figura 7 – Pesquisar locações simples.

The 'Pesquisar locações com:' modal has a close button (X) and two tabs: 'Simples' and 'Avançado'. The 'Simples' tab is active, showing the following fields:

- CPF:** Input field with placeholder 'Digite o CPF'.
- Telefone:** Input field with placeholder 'Digite o Telefone'.
- Nome:** Input field with placeholder 'Digite o Nome'.
- Externo:** Input field with placeholder 'Digite o Documento Externo'.
- Pesquisar:** A large blue button.
- Fechar:** A large grey button.

Fonte: Os autores.

Preenchendo os campos, podemos clicar em pesquisar para abrir a tela de “Consultar” já com todas as locações do cliente pesquisado ou do período pesquisado. Na tela de “Consultar”, as locações do cliente vão aparecer em azul-claro, como se fossem botões, que ao clicar, abre em baixo todos os itens alocados por essa locação, existe um botão “Opções” ao lado da locação, que ao clicar nesse botão, abre um *dropdown* com as ações de “Devolver” e “Cancelar”, ambas precisam de confirmação para efetivar a ação. Na área expandida ao clicar na locação, os itens que possuem ficha de medidas vão aparecer o botão “Abrir ficha de medidas”, ao clicar nesse botão será aberto o *modal* com as medidas do traje, nesse *modal* é possível realizar a visualização ou a edição das medidas. Os itens com status “Aguardando” ou seja, ainda não foram realizados os ajustes da costura, é possível clicar no botão “Finalizar ajustes” para deixar o traje pronto.

Figura 8 – Pesquisar locações.

JL Control					Sistema ▾ Pesquisar ▾ Admin ▾
Nome	ID	Status	Data	Opções ▾	
Robson Leonardo	10	Alocado	19 de Maio de 2021 às 21:07		
ID	Status	Nome Do Traje	Data de Entrega		
10	Aguardando	Traje de gala		<div>Devolver</div> <div>Cancelar</div> <div>Abre ficha de locação</div> <div>Finalizar ajustes</div>	

Fonte: Os autores.

2.3.7. Pesquisa de trajes

O *modal* de pesquisa trajes possui as seguintes formas de consulta: Código, Nome, Modelo e Corte. Todos os campos possuem *autocomplete*. Existe o *switch* de “Mostrar os alocados” que vai buscar os trajes que estão alocados ao invés dos disponíveis que é o padrão. Quando é realizada a busca, exibe em forma de tabela os trajes com o filtro da busca. Caso não inserido nenhuma informação nos campos de busca a mesma retorna todos os trajes de acordo com o *switch*.

Figura 9 – Pesquisa de trajes.

Pesquisar Trajes:

Código:

Nome:

Modelo:

Corte:

☒
Mostrar os alocados

Pesquisar

Fechar

Fonte: Os autores.

Figura 10 – Resultado da pesquisa de trajes.

JL Control							Sistema ▾ Pesquisar ▾ Admin ▾ Sair
#	Codigo	Nome	Modelo	Corte	Valor	Disponibilidade	
1	TRAJE01	Traje Esporte Fino	Tennue de Ville	Australiano	200,00	Disponível	
2	TRAJE2	Traje de gala	Smoking Americano	Australiano	300,00	Disponível	
3	TRAJE9	Traje Americano	Smoking Americano	Australiano Simples	220,00	Disponível	
4	TRAJE4	Traje de gala	Smoking Americano	Australiano	200,00	Disponível	
5	TERNO02	TERNO AZUL MARINHO PANAMA	TERNO	ITALIANO	250,00	Disponível	
6	TERNO03	APOLLO INGLÊS	TERNO	INGLÊS	350,00	Disponível	
Total:	6						

Fonte: Os autores.

2.3.8. Admin trajes

É possível abrir essa tela via Menu: Admin > Trajes (tem acesso somente *login* administrador), essa tela é gerada pelo Django, de acordo com o *models*, aqui que o

administrador do sistema vai realizar as adições e edições dos trajes de todo o sistema. Ao clicar no botão adicionar traje, abrirá um formulário com os campos: Código, Nome, Modelo, Corte, Valor e Ativo, preenchendo esses campos é possível adicionar um novo traje clicando no botão salvar.

Figura 11 – Admin trajes.

Fonte: Os autores.

2.3.9. Admin usuários

É possível abrir essa tela via Menu: Admin > Usuários (tem acesso somente *login* administrador), essa tela é gerada pelo Django e é possível adicionar um novo usuário clicando no botão “Adicionar Usuário”.

Figura 12 – Admin usuários.

Fonte: Os autores.

Ao clicar em salvar, é direcionado para as informações complementares do usuário, como informações pessoais e permissões.

Figura 13 – Admin usuários permissões.

Permissões

☒ **Ativo**
Indica que o usuário será tratado como ativo. Ao invés de excluir contas de usuário, desmarque isso.

☐ **Membro da equipe**
Indica que usuário consegue acessar este site de administração.

☐ **Status de superusuário**
Indica que este usuário tem todas as permissões sem atribuí-las explicitamente.

Grupos:

grupos disponíveis

Escolher todos

grupos escolhido(s)

Os grupos que este usuário pertence. Um usuário terá todas as permissões concedidas a cada um dos seus grupos. Pressione "Control", ou "Command" no Mac, para selecionar mais de um.

permissões do usuário disponíveis

admin | entrada de log | Can add log entry
admin | entrada de log | Can change log entry
admin | entrada de log | Can delete log entry
admin | entrada de log | Can view log entry
auth | grupo | Can add group
auth | grupo | Can change group
auth | grupo | Can delete group
auth | grupo | Can view group
auth | permissão | Can add permission
auth | permissão | Can change permission
auth | permissão | Can delete permission
auth | permissão | Can view permission
auth | usuário | Can add user

Escolher todos

permissões do usuário escolhido(s)

Remover todos

Fonte: Os autores.

Em Permissões é possível indicar as permissões do usuário, colocando somente as permissões de controle, o usuário não tem acesso as ferramentas de administradores.

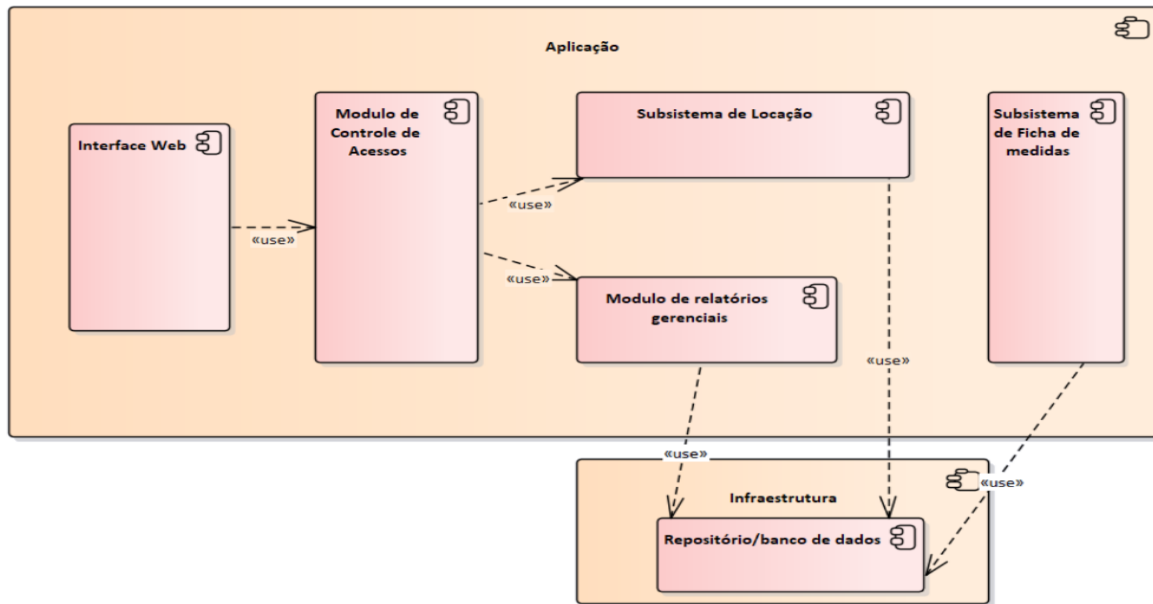
3. Projeto, análise e implementação

A solução apresentada foi uma aplicação *web* desenvolvida utilizando o framework Django que é escrito em Python, com o *front-end* em HTML/CSS e Bootstrap com recursos em Jinja2 com banco em PostgreSQL e a aplicação está hospedada na plataforma Heroku.

3.1. Arquitetura, módulos e subsistemas

A composição da solução é formada por um banco de dados relacional que interage instantaneamente junto a um a solução *web* responsiva, conforme diagrama de componentes abaixo:

Figura 14 – Arquitetura, módulos e subsistemas.



Fonte: Os autores.

3.1.1. Módulo de interface web

Esse módulo é composto por todo o HTML, CSS, JavaScript e Jinja2 da aplicação, é responsável por mostrar os dados que o *back-end* devolve para a interface web, este módulo é o *front-end* da aplicação, como estamos utilizando o *framework* Django esses dados são enviados por *template-view*, que a interface web recebe e trata esses dados via Jinja2, que então são mostrados para o usuário, esse módulo também é importante para enviar dados para o *back-end*, por exemplo na hora de realizar uma locação, o JavaScript trata os dados que o usuário adicionou nos formulários e envia para a *view* do Django fazendo requisições *Fetch*, que é respondido por um código, que representa sucesso ou alguma falha.

A maioria da comunicação que se trata de um recebimento de informações do *back-end*, exemplo: Consulta Locações, Consulta Trajes, Consulta Cliente e Relatórios são via *template-view* do Django, o Jinja2 é responsável por receber esses dados e mostrar de forma organização para os usuários, geralmente colocando em alguma tabela ou preenchendo um formulário. As comunicações que o módulo de interface web envia para o *back-end* operar, são via requisições *Fetch* do JavaScript por exemplo, Devolver Locação, Realizar Locações, Cadastrar Clientes, Criar Ficha de Medidas, Finalizar ajustes do traje e algumas requisições mais simples.

Todas as funções de autocompletar um campo desse módulo, é via jQuery utilizando o *widget* de *autocomplete* que comunica diretamente com o *view* do Django e retorna os dados necessários para autocompletar.

3.1.2. Módulo de controle de acesso

Esse módulo é responsável por controle de acesso a aplicação, fornece uma tela de *login* e controla todas as requisições certificando que a solicitação está com um usuário logado, se alguém que não estiver logado tentar acessar uma tela do sistema o mesmo será redirecionado para a tela de *login*.

Estamos utilizando uma ferramenta do *framework* Django *Authentication System*, nesse módulo é oferecida a segurança necessária para o sistema, gerenciará o cadastro de usuários e permissões para acessar determinada função do sistema. Com esse módulo, conseguiremos

bloquear ou permitir funções do sistema de acordo com permissão, por exemplo, o botão do Menu: Sistema > Admin ficará visível somente se o usuário logado tem permissão "admin".

Caso um usuário tente acessar via URL uma tela/função que não tenha permissão, o sistema redirecionará para a tela de *login* prevenindo a utilização não permitida.

3.1.3. Módulo de locação

O subsistema de locação é responsável por gerir toda a etapa de locação de trajes, cancelamento e devolução da locação e a consulta da locação, esse subsistema é o mais complexo e com mais regras de negócios, utiliza todos os módulos do sistema, gerencia as telas de consulta locações e locar do módulo de interface *web*. É responsável por receber todos os dados para a criação de locação, verificar as regras de negócios, conferir se os dados estão corretos, se os trajes estão disponíveis para locação e efetuar a chamada para persistência dos dados no banco de dados, responsável por receber informações de busca da tela de consultar locações e realizar a busca das locações no banco de dados e retornar esses dados para o módulo de interface *web* tratar. Recebe requisições *Fetch* para devolver e cancelar locação, que realiza as verificações necessárias e persiste a informação no banco de dados.

3.1.4. Módulo de ficha de medidas

O subsistema de ficha de medidas é o responsável pela criação e edição das fichas de medidas, funciona no mesmo conceito do subsistema de locação, gerencia a tela de Costura e todas as operações *Fetch* de criação/edição/consulta de ficha de medidas.

3.1.5. Módulo de relatórios gerenciais

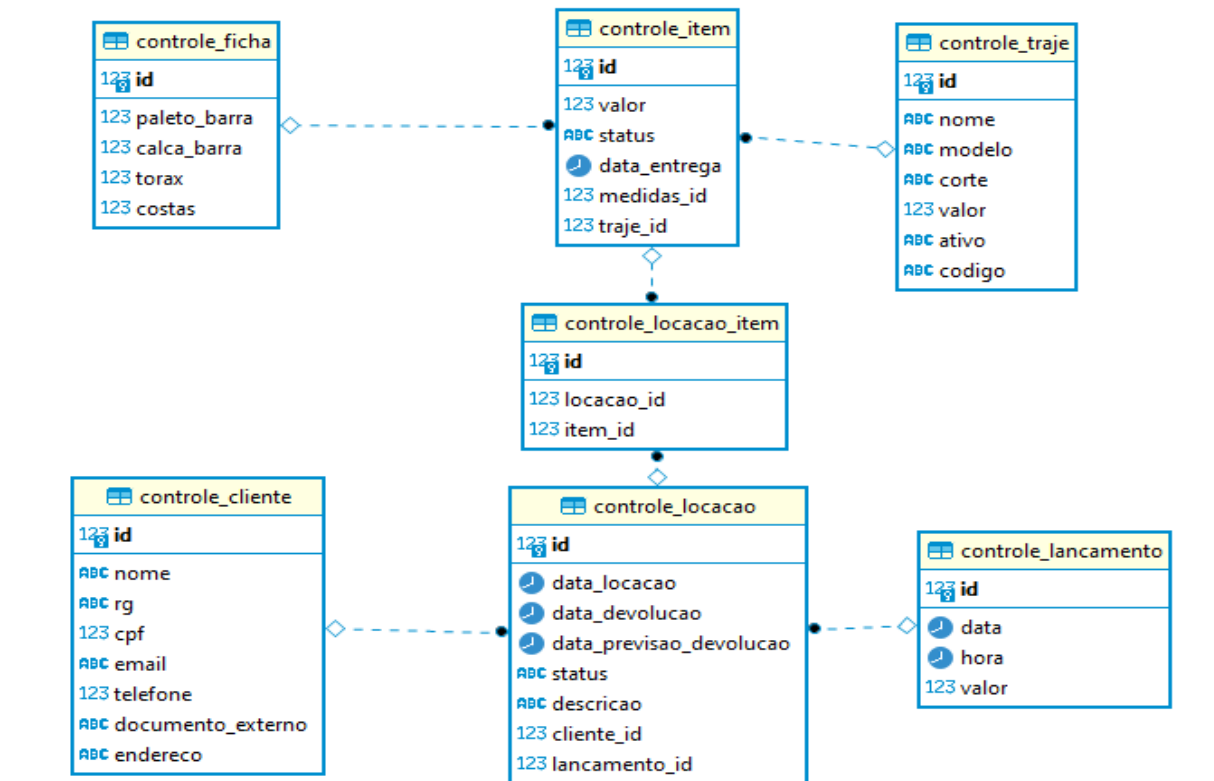
Nesse módulo, disponível somente para os administradores do sistema, possui relatórios gerenciais, a interface *web* manda qual é a página do relatório solicitado e as informações de pesquisa, por exemplo, se o usuário solicitar o relatório de locações efetuadas em um intervalo de datas, o modulo de interface *web* vai enviar para o *back-end* um formulário com uma *action* /busca-por-data/ com os dados de dPrevInicial e dPrevFinal, como data inicial e data final respectivamente, com essa informação, o *back-end* realizará as devidas buscas no modulo de banco de dados e retorna as informações para o módulo de interface *web* e o Jinja2 vai ser responsável para exibir esses dados.

3.2. Projeto de banco de dados

Para o início do projeto, foi necessário realizar a estruturação da base de dados com o objetivo de obter uma visão mais profunda e possibilitar maior entendimento dos dados disponíveis. Nessa estruturação foi criado o modelo lógico, que possibilitou identificar com mais clareza os dados e definir como o projeto poderia ser desenvolvido e aplicado às necessidades da empresa.

Na Figura 15, é possível analisar quais tabelas possuem informações de forma ampla e visual sobre a arquitetura da base de dados.

Figura 15 – Projeto de banco de dados.



Fonte: Os autores.

Quadro 4: Tabelas e campos.

Requisitos do sistema		
Tabela	Campos	Descrição
controle_ficha	Tabela para controle da ficha de medidas.	
	id	Chave da tabela.
	peleto_barra	Valor da medida da barra do paletó.
	calca_barra	Valor da medida da barra da calça.
	torax	Valor da medida do tórax do paletó.
	costas	Valor da medida das costas do paletó.
controle_item	Tabela para controle da ficha de medidas.	
	id	Chave da tabela.
	valor	Valor do item.
	status	Status “Pronto” para retirada e “Aguardando”.

	data_entrega	Data de entrega estipulada.
	medidas_id	Chave estrangeira da tabela controle_ficha.
	traje_id	Chave estrangeira da tabela controle_traje.
controle_traje	Tabela de controle de trajes por unidade e tipos.	
	id	Chave da tabela.
	nome	Nome do traje.
	corte	Tipo de corte do traje.
	valor	Valor da peça unitária.
	ativo	Ativo para trajes disponíveis, Inativo para trajes não-disponíveis.
	codigo	Código externo do traje que vincula com o item do banco de dados.
controle_locacao_item	Tabela de relacionamento entre controle_item e controle_locacao.	
	id	Chave da tabela.
	locacao_id	Chave estrangeira da tabela controle_locacao.
	item_id	Chave estrangeira da tabela controle_item.
controle_cliente	Tabela de cadastro de clientes.	
	id	Chave da tabela.
	nome	Nome do cliente.
	rg	Documento Registro Geral.
	cpf	Documento de cadastro de pessoa física.
	e-mail	<i>E-mail</i> do cliente.
	telefone	Telefone do cliente.
	documento_externo	Documento de estrangeiro.
	endereco	Endereço do cliente.

controle_locacao	Tabela de cadastro de pedido.	
	id	Chave da tabela.
	data_locacao	Data da locação.
	data_devolucao	Data de previsão para devolução.
	status	Alocado, devolvido, atraso e cancelado.
	descricao	Campo de observações.
	cliente_id	Chave estrangeira da tabela controle_cliente.
	lançamento_id	Chave estrangeira da tabela controle_lançamento.
controle_lancamento	Tabela de controle de pagamentos.	
	id	Chave da tabela.
	data	Data do lançamento.
	hora	Data do lançamento.
	valor	Valor total do lançamento.

Fonte: Os autores.

3.3. Estrutura do sistema

Optamos por uma única aplicação de *software* em camadas no qual a interface de usuário e código de acesso aos dados é combinado em um único programa a partir de uma única plataforma. A filosofia do projeto consiste em um aplicativo que não é responsável apenas por uma determinada tarefa, mas que também pode executar todos os passos necessários para completar uma determinada função.

3.4. Tecnologias utilizadas

Quadro 5 – Tecnologias utilizadas.

Tecnologia	Justificativa
Git	Sistema de versionamento de código distribuído entre todos os desenvolvedores.
GitHub	Site onde o código-fonte é hospedado e versionado. https://github.com/caikecolantonio/JLControl
Trello	Sistema <i>web</i> grátis de controle de tarefas e organização de <i>sprints</i> no modelo <i>Kanban</i> .

Python	Linguagem de programação utilizada para desenvolver o <i>back-end</i> do projeto. Linguagem que os integrantes do grupo possuem mais conhecimento.
PostgreSQL	Sistema gerenciador de banco de dados relacional utilizado para persistirem-se os dados cadastrais do projeto. O Heroku já fornece uma integração automatizada.
Django	Django é um <i>framework</i> da <i>web</i> gratuito e de código aberto baseado em Python que segue o padrão arquitetônico <i>model-template-views</i> . É utilizado devido à facilidade de integração com páginas <i>web</i> , realiza todo controle de acesso de usuários e gerência o banco de dados por ORM.
Jinja 2	Biblioteca do Python utilizada para, em conjunto com o Django, montar-se no <i>back-end</i> as páginas HTML que serão servidas ao navegador cliente. Já possui nativamente a integração do <i>template view</i> do Django.
HTML	Padrão no qual as páginas servidas ao navegador cliente estão codificadas.
CSS	Padrão utilizado pelo navegador para estilizar e formatar as páginas clientes adequadamente.
JavaScript	Linguagem de programação utilizada para desenvolver o <i>front-end</i> do projeto. Utilizado em <i>Vanilla</i> para menor consumo de memória e para não utilizar bibliotecas para soluções de pequena escala.
Bootstrap	Biblioteca utilizada para simplificar e padronizar a estilização do <i>front-end</i> .
jQuery	Biblioteca em Javascript utilizada para simplificar e agilizar o desenvolvimento de diversas funcionalidades do <i>front-end</i> .
Heroku	Utilizado para hospedar o <i>back-end</i> e o banco de dados.

Fonte: Os autores.

4. Considerações finais

Expandiremos este projeto aos demais departamentos da empresa como compras, financeiro, contas a pagar, entre outros, em busca do melhor desempenho na organização e clareza das informações aplicadas para tomadas de decisões de forma mais assertivas. A ferramenta JLControl será implementada para que seja possível acompanhar o gerenciamento de forma mais rápida e online.

O projeto desenvolvido pode ser acessado por meio da URL:

<https://jlima-control.herokuapp.com/>

Login: professores

Senha: senha12345

Referências bibliográficas

BARBALHO, P. **Descubra o Data Warehouse: Produtividade e Rapidez.** SQL Magazine, Rio de Janeiro, n. 03, p. 34-38, 2003.

CASTRO, Elisabeth, and Bruce HYSLOP. **HTML5 e CSS3 – Guia Prático e Visual.**” (2013).

KIMBALL, R. **Data Warehouse Toolkit: Guia Completo para Modelagem Dimensional.** 2ª ed. Rio de Janeiro: Campus, 2002. 494p.

MAZZA, Lucas. **HTML5 e CSS3. Domine a web do futuro.** São Paulo, Casa do código (2015)

THOMSEN, E. **OLAP: Construindo Sistemas de Informações Multidimensionais.** Rio de Janeiro: Campus, 2002. 708p.

TURBAN, E.; SHARDA, R.; ARONSON, J. E.; KING, D. **Business Intelligence: Um Enfoque Gerencial Para A Inteligência Do Negócio.** São Paulo: Artmed, 2008. 254p.

PRESSMAN, R. S. **Engenharia de Software: uma abordagem profissional.** New York: The McGraw-Hill Companies, Inc., 2011.

Agradecimentos

Agradecemos, em primeiro lugar, a Deus, pela força e coragem durante toda esta longa caminhada. A todos os professores do curso, que foram muito importantes na nossa vida acadêmica e no desenvolvimento do projeto. Em especial aos Professores Fabio Furia Silva, Osvaldo Kotaro Takai e Victor Williams Stafusa da Silva, por todo apoio e orientação na realização deste trabalho.

As nossas famílias, por acreditarem e investirem em nós e por estarem sempre por perto nos incentivando, dando alegria e motivação a cada dia, fácil ou difícil, nesta jornada.

E, finalmente, aos nossos amigos que, ao longo do curso, se mostraram verdadeiros companheiros, fizeram parte da nossa formação e continuarão presentes em nossas vidas.