

Fatec Guarulhos

Integrantes

Amanda Yasmin Botto
Danilo Zamai
João Gabriel Lima de Moraes
Lucas Silva Contieri
Mateus Figueiras Araújo
Mateus Picoli dos Santos
Paulo César Perez Percinoto Júnior
Priscila Diniz Nascimento
Silas Castro Lima
Vinícius Feitosa de Araújo

Objetivo

Projeto de desenvolvimento de software apresentado ao curso de Análise e Desenvolvimento de Sistemas da Faculdade de Tecnologia de Guarulhos sob orientação dos professores Jane Maria dos Santos Eberson e Milton Francisco de Brito.

Este trabalho tem como objetivo a criação de um sistema funcional de controle de estoque de produtos para microempresas direcionado, mais especificamente, a uma loja de artigos para animais domésticos (pet shop). Para este objetivo ser realizado serão utilizados os conteúdos apreendidos durante as aulas de nosso curso durante os quatro primeiros semestres, recursos tecnológicos disponíveis e necessários para tal tarefa. Além disso, contaremos com a orientação dos professores acima citados.

Criação do Documento

Elaborado	Vinícius Feitosa de Araújo	Data:	26/06/2026
por:	Villicius i ellosa de Aradjo	Data.	20/00/2020

Controle de Versão

Versão	Data	Páginas	Mudanças	Comentários
3.0	24/03/2024		Acrescentadas novas telas; Novas informações do banco de dados; Novos planos de teste; DAS e o SQA Plan; Dicionário de dados e termos; Considerações finais e Referências	

Composição da Equipe de desenvolvimento

Número de ordem	Função	Descrição das responsabilidades	Nome
1	Analistas	Análise e especificação de requisitos	Danilo Zamai e Silas Castro Lima
2	Desenvolvedores	Desenho e Implementação	Amanda Yasmin Botto, João Gabriel Lima de Moraes, Lucas Silva Contieri e Mateus Filgueiras Araujo
3	Testadores	Testes e gestão da qualidade	Paulo César Perez Percinoto Júnior
4	Projetista de BD	Documentação e projeto de Banco de dados	Mateus Picoli dos Santos e Priscila Diniz Nascimento
5	Documentador	Desenvolvimento e integração de toda documentação constante nesse arquivo	Vinícius Feitosa de Araújo

INTRODUÇÃO

Em uma reunião realizada na data de 26/03/2024, nossa equipe teve a oportunidade de conhecer nosso cliente Milton Francisco de Brito. Desta forma viemos a saber que ele está interessado em um sistema **ERP** para **controle de estoque** de sua loja digital de artigos para animais de estimação (pets). O cliente nos contou que já possui um sistema para fazer tal controle, mas gostaria de ter algo mais eficiente.

• Identidade visual:

Foi-nos apresentado o site já existente de sua loja *AgentePet* para que, desta forma, pudéssemos ter uma referência para a identidade visual de nosso software. A cor predominante do site é azul.

Se possível a utilização de imagens, ícones, abas com o tema animais de estimação. Na separação dos produtos, por exemplo, a segmentação das categorias por cachorros, gatos, peixes etc.

Temos o lema da loja que também pode ser trabalhado em alguma etapa do projeto: "Para gente ou para pet, conte com AgentePet".

• Desempenho:

O cliente busca, entre outras coisas, a redução do estoque por meio do desempenho do software, da previsão de tendências por meio do histórico de vendas e da geração de **relatórios** por este sistema. Desta forma, a loja poderia reduzir seu estoque que hoje conta com cerca de 450 para 50 itens.

Outra característica que o cliente gostaria que estivesse presente no sistema é o controle de produtos de acordo com a **Curva ABC**, que consiste na categorização dos itens de acordo com a demanda e preço desses produtos: a categoria A seria a categoria dos produtos mais caros, com menos fornecedores, mas com menor demanda, por exemplo.

Controle do **lead time** (tempo de esper<mark>a) de entre</mark>ga dos fornecedores, para que um produto não esteja em falta enquanto aguarda sua chegada.

Segurança:

Login obrigatório para acesso ao software, com hierarquia de permissões. Controle de usuários quanto à edição de informações dentro da plataforma. Por tratar-se de um sistema de controle de estoque, é necessário evitar o risco de prejuízo por parte da empresa por questões de segurança.

• Funções:

Controle de entrada e saída de produtos. Separação de produtos por preço, tipos, cores, ou seja, por suas características. Possibilidade de gerar relatórios com histórico e tendências.

Vale destacar que o cliente citou diversas vezes que procura por um sistema **intuitivo**, para que não haja gastos com treinamento de pessoal e que, desta forma, sua implementação seja imediata.

PROJETO CONCEITUAL

Versão de Orientação a Objetos - UML Projeto DENNA

1. Documento de Visão

Este documento tem por finalidade expressar uma visão geral da solução proposta para aprimorar o controle e organização do estoque de uma loja digital de artigos para animais de estimação (AgentePet). A proposta visa desenvolver um sistema ERP simples, intuitivo e eficiente, capaz de atender às necessidades do cliente em relação à gestão de mercadorias, previsões de demanda e redução de estoque.

1.1. Resultado do Projeto

O sistema a ser desenvolvido irá fornecer uma solução completa de controle de estoque, voltada para microempresas do setor pet shop, com foco na automação de processos, melhoria da organização e redução de perdas. Através de funcionalidades como login com hierarquia de permissões, relatórios de entrada/saída e Curva ABC, o sistema permitirá uma gestão mais inteligente e estratégica do estoque.

1.1.1. Finalidade do Projeto

Criar um sistema ERP para controle de estoque que seja mais eficiente e funcional que o utilizado atualmente, proporcionando ao cliente maior controle, praticidade e confiabilidade nas operações diárias da loja.

1.1.2. Diagnóstico da Situação Atual

		~		1		
	I Decria	ח חבי	\cap	nrol	n	ama.
ᆫ	escrio	,au u		$\mathbf{D} \mathbf{I} \mathbf{D} \mathbf{I}$	U	CITIC

Falta de previsibilidade e controle de demanda;

Descrição do problema

Estoque elevado, gerando custos e desperdícios;

Descrição do problema

Interface pouco intuitiva, exigindo tempo e custo com treinamentos;

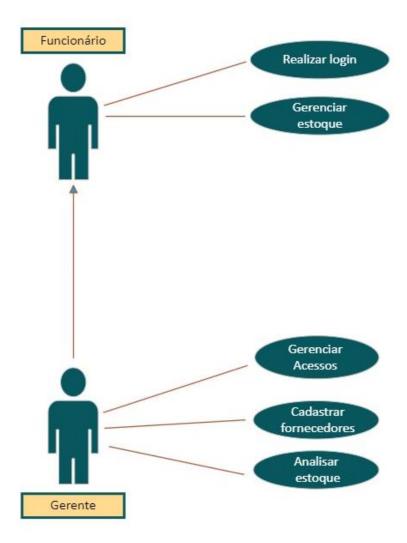
Descrição do problema

Falta de relatórios inteligentes que auxiliem na tomada de decisão;

Descrição do problema

Ausência de categorização eficiente dos produtos, como Curva ABC.

1.2. Diagrama de caso de uso



1.3. Requisitos do Projeto

1.3.1. Requisitos não funcionais

Identificador	Requisito		Classificação	
RF01	Realizar cadastro	o de usuário/login	Essencial	
RF02	Pesquisa simplifi	icada por código de produto	Essencial	
RF03	Entrada e baixa	de itens do estoque	Essencial	
RF04	Gerenciamento d	de usuário.	Essencial	
RF05	Realizar cadastro	o de mercadoria.	Essencial	
RF06	Gerenciamento d	Essencial		
RF07	Visualização do	Essencial		
RF08	Gerar relatório de entrada e saída de estoque, com Importante filtros por data			
RF09	Gerar relatório d	e curva ABC	Importante	
Identificador	Propriedade	Descrição		
RNF01	Confiabilidade	Segurança das informações da loja e de	e seus clientes.	
RNF02	Disponibilidade	Garantir que o sistema e os dados por ele gerados estejam disponíveis e acessíveis para usuários autorizados quando eles precisarem.		
RNF03	Usabilidade Interface agradável seguindo princípios Gestalt.			
RNF04	Acessibilidade	Desempenho e rapidez no acesso às in	formações.	
RNF05	Portabilidade	Capacidade de ser desenvolvido ou ex arquiteturas (seja de hardware ou de so		

RNF06	Interoperabilidade	Fácil acesso e comunicação de dados e informações exportados de outros sistemas (Excel, BI, SQL, Acess).
RNF07	Usabilidade	Interface intuitiva e que de a impressão de continuidade.
RNF08	Usabilidade	Interface com ícones sugestivos, que sejam segregados
RNF09	Desempenho	O sistema deve ser capaz processar grande volume de dados
RNF10	Manutenibilidade	O código deve ser o máximo limpo para futuras mudanças

1.4. Tabela de Testes Plano de Testes

Objetivo

Os planos de testes a seguir têm como objetivo mostrar os testes realizados no aplicativo de controle de estoque para se certificar que todos os componentes do aplicativo estão funcionando devidamente, usando um roteiro simples e objetivo para aumentar a reprodutibilidade dos testes.

Tela de Login

- O usuário deverá abrir o aplicativo
- O usuário deverá ter um cadastro no Banco de dados

Código	Descrição	Passo a Passo	Resultado Esperado	Resultado Obtido	Testado Por:
LO-001	Login bem- sucedido	1-Clicar no campo "Usuário" e digitar um usuário válido 2-Clicar no campo "Senha" e digitar uma senha válida. 4-Clicar no botão "Entrar".	Após clicar no botão "Entrar", a tela de Login se fechará e o aplicativo principal será aberto.	Login feito com sucesso	Paulo César
LO-002	Usuário e/ou senha incorretas (erro)	1-Clicar no campo "Usuário" e digitar um usuário válido 2-Clicar no campo "Senha" e digitar uma senha inválida. 4-Clicar no botão "Entrar". 5-Preencher novamente os campos, desta vez digitando um usuário inválido e uma senha qualquer. 6-Clicar no botão "Entrar".	Ambas as vezes, após clicar no botão "Entrar", aparecerá uma mensagem informando o usuário que o usuário e/ou a senha estão incorretos.	Mensagem Aparece como esperado	Paulo César
LO-003	Ambos os campos de Usuário e Senha vazios (erro)	1-Preencher apenas um dos campos. 2-Clicar no botão "Entrar" 3-Repetir passos 1 e 2, desta vez preenchendo só o outro campo	Após clicar no botão "Entrar", aparecerá uma mensagem informando o usuário que todos os campos devem ser preenchidos.	Mensagem aparece como esperado	Paulo César

Tela de Cadastro de Mercadorias

- O usuário deve já estar logado com uma conta válida
- O usuário deve ter clicado no botão "Mercadoria", que se encontra abaixo de "Cadastro" no menu da lateral esquerda
- Deve haver produtos registrados no banco de dados

Código	Descrição	Passo a Passo	Resultado Esperado	Resultado Obtido	Testad o Por:
ME-001	Adicionar Mercadoria	1-Adicionar todas as informações de um produto 2-Clicar no botão "Salvar	Após clicar no botão "Salvar", todos os campos se tornarão vazios, e uma mensagem informando o usuário de que o produto foi adicionado com sucesso.	Mercadoria foi cadastrada com sucesso	Paulo César
ME-002	Adicionar produto com informações faltando (erro)	1-Preencher todos os campos, exceto um 2-Clicar no botão "Salvar". 3-Repetir passos 1 e 2 até que todos os campos tenham sido testados ao menos uma vez	Após clicar no botão "Salvar", aparecerá uma mensagem avisando que nem todos os campos foram preenchidos	Mensagem apareceu como o esperado	Paulo César
ME-003	Adicionar produto com Código repetido (erro)	1-Digitar todos os campos, fazendo questão de que o código digitado já seja de um produto existente 2-Clicar no botão "Salvar"	Após clicar no botão "Salvar", aparecerá uma mensagem avisando que já existe um produto com o código digitado.	Mensagem apareceu como o esperado	Paulo César

Pré-Condições:

- O usuário deve já estar logado com uma conta válida
- O usuário deve ter clicado no botão "Controle de Estoque", que se encontra abaixo de "Suprimentos" no menu da lateral esquerda
- Deve haver produtos Registrados no banco de dados

Obs: Nesta tela se encontram 5 botões sem rótulos. Para facilitar nos testes cada botão terá um nome, sendo estes da esquerda para direita: "Pesquisar", "Carregar", "Editar", "Limpar" e "Excluir".

Código	Descrição	Passo a Passo	Resultado Esperado	Resultado Obtido	Testado Por:
ES-001	Carregar todos os produtos	1-Clicar no Botão "Carregar"	Após clicar no botão "Carregar", aparecerá todos os produtos registrados	Produtos Carregados com sucesso	Paulo César
ES-002	Carregar um produto específico	1-Digitar um código existente no campo a esquerda do botão "Pesquisar" 2-Clicar no botão "Pesquisar"	Após clicar no botão "Pesquisar", aparecerá apenas o produto pesquisado	Produto carregado com sucesso	Paulo César
ES-003	Excluir um produto	1-Clicar no Botão "Carregar" 2-Clicar em um produto 3-Clicar no botão "Excluir" e confirmar. 3-Realizar os mesmos passos do teste ES-002, porém usando o código do produto excluído	Após clicar no botão " <u>Pesquisar</u> ", o produto excluído não aparecerá	Produto excluído com sucesso	Paulo César
ES-004	Limpar a tela	1-Clicar no botão "Carregar" 2-Clicar no botão "Limpar"	Após clicar no botão "Limpar", todos os produtos serão retirados da tela	Tela limpa com sucesso	Paulo César
ES-004	Editar um produto	1-Clicar no botão "Carregar" 2-Clicar em um produto 3-Clicar no botão "Editar" 4-Alterar um dado qualquer 5-Clicar em "Salvar Alterações"	Após clicar no botão "Salvar Alterações", os dados do produto	Ambos foram atualizados com sucesso	Paulo César

		6-Clicar no botão "Carregar 7-Repetir até que todos os campos sejam alterados	devem ser alterados no banco de dados e no próprio programa ao carregar novamente		
ES-005	Editar um produto com informações faltando (erro)	1-Clicar no botão "Carregar" 2-Clicar em um produto 3- Clicar no botão "Editar" 3-Apagar o conteúdo de um campo qualquer 4-Clicar no botão "Salvar alterações	Após clicar no botão "Salvar Alterações", aparecerá uma mensagem dizendo que algum campo não foi preenchido	Mensagem apareceu como o esperado	Paulo César

Tela de Entrada de mercadoria

- O usuário deve já estar logado com uma conta válida
- O usuário deve ter clicado no botão "Registrar entrada", que se encontra abaixo de "Controle de estoque" no menu da lateral esquerda
- Deve haver produtos Registrados no banco de dados
- Deverá ter ocorrido entrada e saída de produtos

Código	Descrição	Passo a Passo	Resultado Esperado	Resultado Obtido	Testado Por:
EN-001	Mostrar todas as entradas e saídas	1-Clicar no Botão "Buscar"	Após clicar no botão "Buscar", aparecerá todas as entradas e saídas	Informação Apareceu como o esperado	Paulo César
EN-002	Mostrar Entradas e saídas com filtro	1-Digitar um valor válido no campo do Código do Produto 2-Clicar no botão "Buscar" 3-Repetir o processo com cada campo de busca, até que todos sejam testados.	Cada vez que o botão "Buscar" for pressionado, se houver um produto que corresponda com os filtros ele aparecerá na tabela	O filtro funcionou como o esperado	Paulo César
EN-003	Registrar entrada de Produto	1-Clicar no botão "Buscar" 2-Clicar em uma mercadoria 3-Selecionar "Entrada" como tipo de movimento	Após clicar no botão "Buscar", pela segunda vez, a entrada que o usuário registrou aparecerá na	Entrada registrada com sucesso	Paulo César

		4-Preencher os campos "Quantidade" e "Preço Unitário" com valores válidos 5-Clicar no botão "Atualizar Hora" 6-Clicar no botão "Registrar Entrada" 7-Clicar no botão "Buscar"	tabela, junto com as anteriores		
EN-004	Registrar saída de produto	1-Clicar no botão "Buscar" 2-Clicar em uma mercadoria 3-Selecionar "Saída" como tipo de movimento 4-Preencher os campos "Quantidade" e "Preço Unitário" com valores válidos 5-Clicar no botão "Atualizar Hora" 6-Clicar no botão "Registrar Entrada" 7-Clicar no botão "Buscar"	Após clicar no botão "Buscar", pela segunda vez, a Saída que o usuário registrou aparecerá na tabela, junto com as anteriores	Saída registrada com sucesso	Paulo César
EN-005	Registrar saída de produto maior do que estoque (erro)	1-Clicar no botão "Buscar" 2-Clicar em uma mercadoria 3-Selecionar "Saída" como tipo de movimento 4-Preencher os campos "Quantidade" e "Preço Unitário", sendo que o valor do campo "Quantidade" deve ser maior do que o estoque do produto 5-Clicar no botão "Atualizar Hora" 6-Clicar no botão "Registrar Entrada"	Após clicar no botão "Registrar Entrada", haverá um aviso alertando o usuário de que não há estoque suficiente para realizar a saída	Alerta funciona como o esperado	Paulo César

Tela de Relatório de movimento

- O usuário deve já estar logado com uma conta válida
- O usuário deve ter clicado no botão "Movimentação", que se encontra abaixo de "Relatórios" no menu da lateral esquerda
- Deve haver produtos Registrados no banco de dados
- Deverá ter ocorrido entrada e saída de produtos

Código	Descrição	Passo a Passo	Resultado	Resultado	Testado
			Esperado	Obtido	Por:

MO-001	Mostrar	1-Clicar no Botão "Gerar	Após clicar no	Funcionan	Paulo
	Movimentação completa	Relatório"	botão " <u>Gerar</u> <u>Relatório</u> ", aparecerá toda a Entrada e Saída de Mercadorias	do como o esperado	César
MO-002	Mostrar movimentação de um produto específico	1-Digitar um código existente no campo "Código do Produto" 2- Clicar no botão "Gerar Relatório"	Após clicar no botão "Gerar Relatório", aparecerá a Entrada e Saída do produto selecionado	Funcionan do como o esperado	Paulo César
MO-003	Mostrar movimentação do Mês	1-Selecionar a opção "Do Mês", dentro da área de pesquisa por data 2-Clicar no botão "Gerar Relatório	Após clicar no botão "Gerar Relatório", aparecerá todas as entradas e saídas do Mês	Funcionan do como o esperado	Paulo César
MO-004	Mostrar Movimentação de um período específico	1-Selecionar a opção "De um Período", dentro da área de pesquisa por data 2-Selecionar uma data Inicial qualquer 3-Selecionar uma data final qualquer, desde que seja após a data inicial 4-Clicar no botão "Gerar Relatório"	A pós clicar no botão "Gerar Relatório", aparecerá todas as entradas e saídas dentro do Período selecionado	Funcionan do como o esperado	Paulo César
MO-005	Limpar a tela e desmarcar escolhas	1-Selecionar a opção "Do Mês", dentro da área de pesquisa por data 2-Clicar no botão "Gerar Relatório" 3-Clicar no botão "Limpar", que se encontra abaixo do botão "Gerar Relatório"	Após clicar no botão "Limpar" a tela deve estar sem nenhum produto, e os botões de Data devem estar desmarcados	Funcionan do como o esperado	Paulo César
MO-006	Exportar Relatório para um arquivo	1-Clicar no botão "Gerar relatório" 2-Clicar no botão "Exportar"	Após clicar nos botões "Exportar", um arquivo com os dados da tabela	Funcionan do como o esperado	Paulo César

	deve ser	
	baixado	

Tela de Relatório de curva ABC

- O usuário deve já estar logado com uma conta válida
- O usuário deve ter clicado no botão "Curva ABC", que se encontra abaixo de "Movimentação" no menu da lateral esquerda
- Deve haver produtos Registrados no banco de dados
- Deverá ter ocorrido entrada e saída de produtos

Código	Descrição	Passo a Passo	Resultado Esperado	Resultado Obtido	Testado Por:
AB-001	Mostrar Curva ABC	1-Clicar no Botão "Gerar Relatório"	Após clicar no botão "Gerar Relatório", aparecerá toda a Curva ABC	Funcionando como o esperado	Paulo César
AB-002	Limpar tabela	1-Clicar no Botão "Gerar Relatório" 2-Clicar no botão "Limpar", que se encontra abaixo do botão "Gerar Relatório"	Após clicar no botão "Limpar", a tabela da Curva ABC será esvaziada	Funcionando como o esperado	Paulo César
AB-003	Exportar Curva ABC para um arquivo	1-Clicar no botão "Gerar relatório" 2-Clicar no botão "Exportar"	Após clicar nos botões "Exportar", um arquivo com os dados da tabela deve ser baixado	Funcionando como o esperado	Paulo César

1.5. Estrutura de Dados no SGBD MySQL

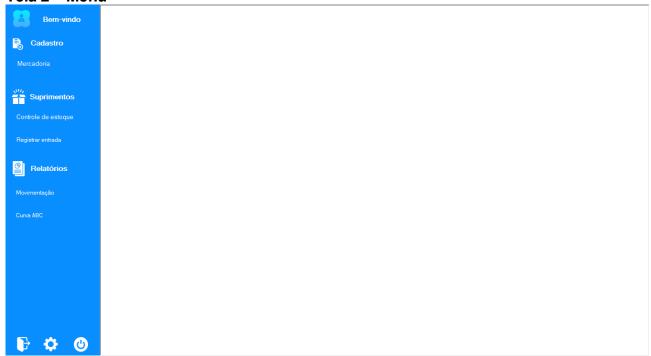
Mercadoria	Estoque	Movimentos_Estoque
ID(pk)	codigo_mercadoria	id_movimento
Nome	quantidade_atual	codigo_mercadoria
Quantidade	data_ultima_atualizacao	tipo movimento
Itens_caixa		quantidade
estoque_minimo		preco_unitario_movimento
estoque_maximo		data_hora_movimento
Volumes		observacoes
Peso_liquido		
Peso_bruto		
Largura		
Altura		
Profundidade		
Preco_venda		
Departamento		
codigo		
Formato		
Condicao		
Tipo		
Situacao		
Marca		
Producao		
Data_validade		
Frete_gratis		
Unidade_Medida		
Variacao_Atributo		
Variacao_Opcao		
GTIN		
Localizacao		

1.6 Telas

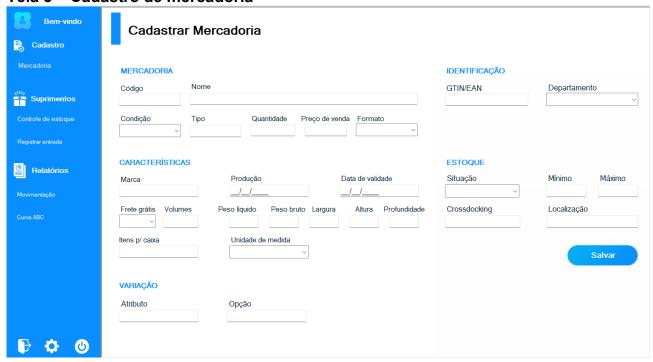
Tela 1 - Login



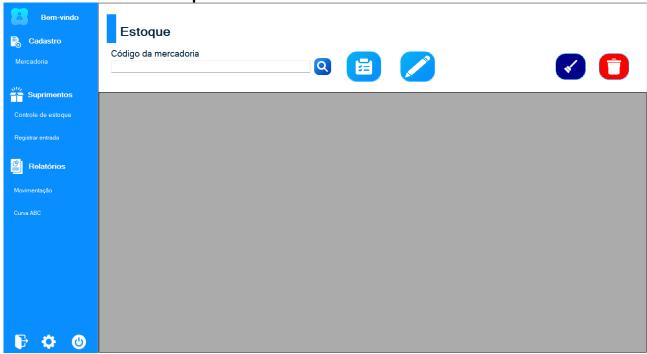
Tela 2 – Menu



Tela 3 - Cadastro de mercadoria



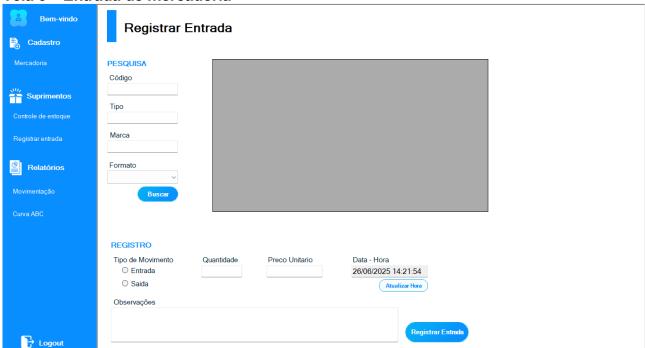
Tela 4 - Controle de estoque



Tela 5 - Edição de mercadoria



Tela 6 - Entrada de mercadoria



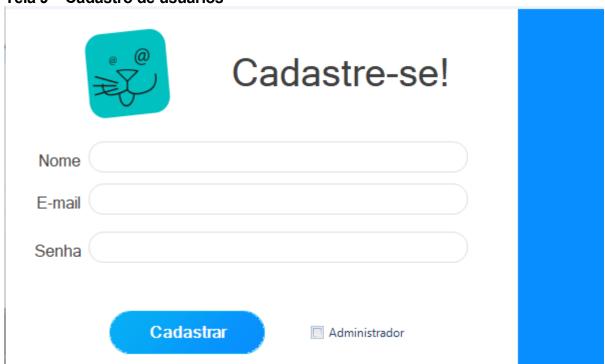
Tela 7 - Relatório de movimento



Tela 8 - Relatório de curva ABC



Tela 9 - Cadastro de usuários



1.7 Relação de Classes de Objetos persistentes

Nome da Classe de Objetos	Conceito da Classe de Objetos	Principais atributos
Mercadoria	Representa qualquer item físico armazenado, comercializado ou movimentado no estoque, contendo informações sobre suas características, quantidade disponível, dimensões, preço, validade e localização no armazém	quantidade, itens_caixa, estoque_minimo, estoque_maximo volumes, peso_liquido, peso_bruto, largura, altura, profundidade, codigo, nome, formato, condicao, tipo, situacao, preco_venda, marca, producao, data_validade, frete_gratis, unidade_medida, variacao_atributo, variacao_opcao, gtin, departamento, crossdocking, localizacao.

1.8. Modelagem e projeto de Banco de Dados

Ao desenvolvermos o software de gestão de estoque, para nós, uma etapa muito importante foi a criação de seu banco de dados. A modelagem cuidadosa da estrutura de dados foi fundamental para garantirmos a eficiência, a integridade e a segurança das informações. Estruturas essenciais para o controle de um estoque empresarial.

Nossa principal decisão foi adotar um modelo relacional, que se mostrou a abordagem mais adequada para lidar com as entidades e seus relacionamentos de forma estruturada e flexível. Para o gerenciamento a princípio, optamos pelo **SGBD Access** e, posteriormente, utilizamos também o **MySQL**, escolhas que se alinhavam com a necessidade de uma solução robusta, porém acessível, para a empresa de nosso cliente.

A estrutura do banco de dados foi projetada para refletir diretamente os principais elementos do sistema, conforme mostramos no nosso Diagrama de Classes. A tabela central que compõe nosso modelo é:

• Mercadorias: Tabela que representa o coração do controle de estoque. Nela, armazenamos detalhes fundamentais sobre cada produto, como *Id, Nome, Código, Tipo, Preço de Venda, Estoque, Quantidade a Encomendar, Falta e Enviar*. Incluímos também campos para *Categoria, Variação, Marca, Produção e Validade*, o que permite uma segregação e busca de produtos mais detalhada – um requisito do cliente para categorizar itens por tipo (cachorros, gatos, peixes, etc.) e aplicar a Curva ABC. A inclusão de Lead Time de entrega dos fornecedores garante que o sistema possa prever e evitar a falta de produtos.

Além dessas entidades primárias, o banco de dados foi criado para suportar a geração de relatórios detalhados, um dos pilares para a redução de estoque e previsão de tendências solicitadas pelo cliente. Isso se concretiza através da capacidade de registrar entradas e saídas de produtos e suas movimentações.

A organização dessas tabelas e seus atributos foi pensada para otimizar o desempenho do software, um ponto crucial para o cliente, que busca algo mais eficiente que seu sistema atualmente utilizado. A normalização dos dados foi aplicada para minimizar redundâncias e garantir a integridade, facilitando futuras manutenções e o processamento de grandes volumes de dados.

Nosso projeto de banco de dados, estabeleceu um equilíbrio entre a eficiência e a simplicidade permitindo sua fácil implementação e uso, sem a necessidade de grandes treinamentos, tratando-se, portanto, de um sistema intuitivo.

1.8.1 Dicionário de Dados

Este dicionário de dados tem como objetivo documentar as tabelas, campos, seus respectivos tipos, tamanhos e descrições, fornecendo uma visão clara e organizada da estrutura do banco de dados do sistema de controle de estoque. Ele é fundamental para o entendimento, manutenção e futuras expansões do sistema.

1.8.1.1 Tabela: mercadorias

Esta tabela armazena informações detalhadas sobre os produtos ou itens que compõem o estoque. É o cadastro principal de todos os produtos comercializados.

Campo	Tipo de Dado	Tamanho	Chave	Descrição
ID	INT	-	PK, Auto- Inc.	Identificador único e sequencial da mercadoria.
Nome	VARCHAR	100	-	Nome completo da mercadoria.
Quantidade	INT	30	-	Quantidade total de itens desta mercadoria em estoque (considerando o estoque físico inicial e não os movimentos)
Itens_caixa	INT	-	-	Quantidade de itens contidos em uma caixa ou embalagem.
Estoque_minimo	INT	-	-	Limite mínimo de estoque para esta mercadoria, a partir do qual alertas podem ser gerados.
Estoque_maximo	INT	-	-	Limite máximo de estoque desejável para esta mercadoria.
Volumes	FLOAT	6, 2	-	Número de volumes que a mercadoria ocupa.
Peso_liquido	FLOAT	6, 2	-	Peso da mercadoria sem a embalagem.
Peso_bruto	FLOAT	6, 2	-	Peso da mercadoria com a embalagem.
Largura	FLOAT	6, 2	-	Largura da mercadoria (em uma unidade de medida definida, ex: cm).
Altura	FLOAT	6, 2	-	Altura da mercadoria (em uma unidade de medida definida, ex: cm).

Profundidade	FLOAT	6, 2	-	Profundidade da mercadoria (em uma unidade de medida definida, ex: cm).
Preco_venda	FLOAT	6, 2	ı	Preço de venda unitário da mercadoria.
Codigo	VARCHAR	30	Unique Key	Código alfanumérico exclusivo para identificar a mercadoria. Utilizado como chave de ligação para movimentos_estoqu e.
Formato	VARCHAR	30	-	Formato ou tipo físico da mercadoria (ex: "líquido", "sólido", "granel").
Condicao	VARCHAR	30	-	Condição da mercadoria (ex: "novo", "usado", "remanufaturado").
Tipo	VARCHAR	30	-	Classificação do tipo de mercadoria (ex: "eletrônico", "alimento", "vestuário").
Situacao	VARCHAR	30	-	Situação atual da mercadoria (ex: "ativo", "inativo", "em promoção").
Marca	VARCHAR	50	-	Marca do fabricante da mercadoria.
Producao	VARCHAR	30	-	Informações sobre a produção da mercadoria (ex: "nacional", "importada").
Data_validade	VARCHAR	30	-	Data de validade da mercadoria (formato de texto,).
Frete_gratis	VARCHAR	10	-	Indica se a mercadoria possui frete grátis (ex: "sim", "não").
Unidade_medida	VARCHAR	30	-	Unidade de medida da mercadoria (ex: "un", "kg", "m", "litro").
Variacao_tributari a	VARCHAR	30	-	Atributo de variação da mercadoria (ex: "cor", "tamanho").

Variacao_opcao	VARCHAR	30	-	Opção da variação do atributo (ex: "azul", "P", "M").
GTIN	VARCHAR	30	-	Global Trade Item Number (antigo EAN/UPC), código de barras internacional.
Departamento	VARCHAR	30	-	Departamento ao qual a mercadoria pertence.
Crossdocking	VARCHAR	30	-	Indica se a mercadoria é elegível para cross-docking (ex: "sim", "não").
Localizacao	VARCHAR	30	-	Localização física da mercadoria no estoque (ex: "Corredor A - Prateleira 3").

1.8.1.2. Tabela: estoque

Esta tabela mantém o registro do saldo atual de cada mercadoria no estoque, com a última data de atualização. É a representação em tempo real do volume de cada produto disponível.

Campo	Tipo de Dado	Tamanho	Chave	Descrição
Codigo_mercado ria	INT	-	PK, Auto- Inc.	Identificador único e sequencial da mercadoria.
quantidade_atual	VARCHAR	100	-	Nome completo da mercadoria.
data_ultima_atual izacao	INT	30	-	Quantidade total de itens desta mercadoria em estoque (considerando o estoque físico inicial e não os movimentos)

1.8.1.2. Tabela: movimentos_estoque

Esta tabela registra todas as movimentações (entradas e saídas) de mercadorias no estoque. É o histórico de todas as transações que alteram o saldo do estoque.

Campo	Tipo de Dado	Tamanho	Chave	Descrição
id_movimento	INT	-	PK, Auto- Inc	Identificador único e sequencial do movimento de estoque.
codigo_mercadoria	VARCHAR	30	FK	Código da mercadoria que foi movimentada, referenciando o campo codigo da tabela mercadorias.
tipo_movimento	ENUM	-	1	Tipo de movimento realizado, podendo ser 'ENTRADA' ou 'SAIDA'.
quantidade	INT	10	-	Quantidade de itens movimentados nesta transação.
preco_unitario_mov imento	DECIMAL	8,2	-	Preço unitário da mercadoria no momento em que o movimento ocorreu.

data_hora_movime nto	DATETIME	-	-	Data e hora exata em que o movimento foi registrado.
observacoes	VARCHAR	255	1	Campo para observações adicionais sobre o movimento.

1.8.1.3. .Gatilhos (Triggers)

Os gatilhos são procedimentos armazenados que são executados automaticamente em resposta a eventos específicos (INSERT, UPDATE, DELETE) em uma tabela. Eles garantem a consistência dos dados, especialmente na tabela estoque.

• **after_insert_movimento_estoque**: Este gatilho é acionado após a inserção de um novo registro na tabela movimentos_estoque.

Funcionalidade: Se a mercadoria movimentada ainda não existir na tabela estoque, ele a insere com quantidade zero. Em seguida, atualiza a quantidade_atual na tabela estoque somando ou subtraindo a quantidade do movimento, dependendo do tipo_movimento (ENTRADA ou SAIDA). Se for uma SAÍDA e a quantidade em estoque for insuficiente, ele impede a operação e emite uma mensagem de erro.

• after_update_movimento_estoque: Este gatilho é acionado após a atualização de um registro na tabela movimentos estoque.

Funcionalidade: Primeiramente, ele reverte o impacto do registro antigo (OLD) na tabela estoque. Em seguida, aplica o impacto do registro atualizado (NEW) na quantidade_atual da tabela estoque. Isso garante que a quantidade no estoque seja recalculada corretamente após uma alteração. Assim como no INSERT, ele verifica a disponibilidade para saídas.

• **after_delete_movimento_estoque**: Este gatilho é acionado após a exclusão de um registro na tabela movimentos_estoque.

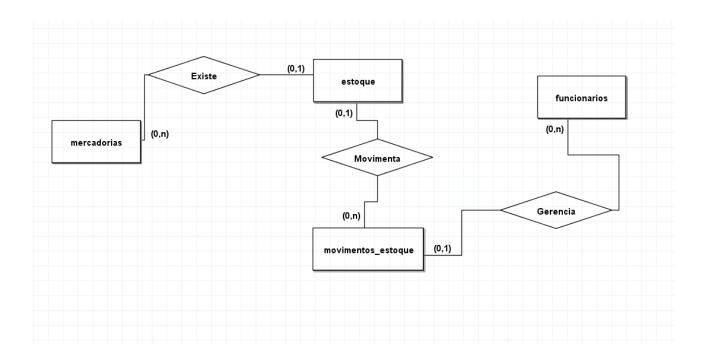
Funcionalidade: Ele reverte o impacto da movimentação que foi excluída na quantidade_atual da tabela estoque. Se a movimentação excluída era uma 'ENTRADA', a quantidade é subtraída do estoque. Se era uma 'SAIDA', a quantidade é adicionada de volta ao estoque.

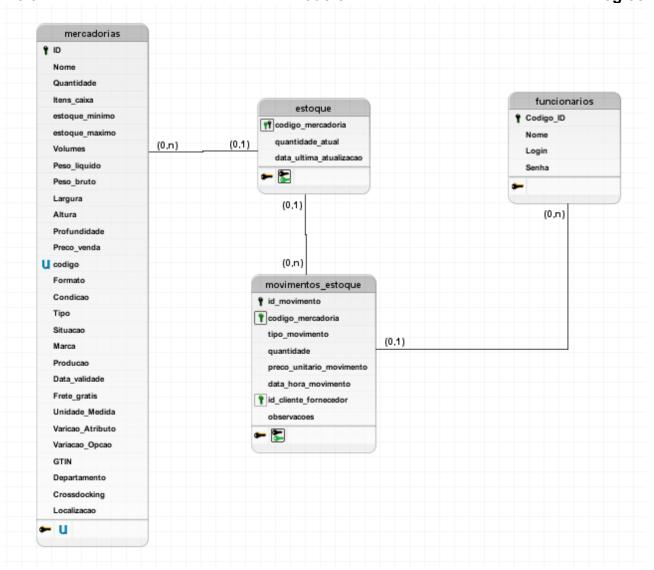
1.8.1.4. Visão (View): view_curva_abc

Esta visão fornece uma análise da Curva ABC dos produtos baseada no valor total das saídas. A Curva ABC é uma ferramenta de gestão de estoque que classifica os itens em categorias (A, B, C) de acordo com sua importância, geralmente definida pelo valor de consumo ou saída.

Campo	Tipo de Dado	Descrição
codigo_mercadoria	VARCHAR	Código da mercadoria.
Marca	VARCHAR	Marca da mercadoria, vinda da tabela mercadorias.
Tipo	VARCHAR	Tipo da mercadoria, vinda da tabela mercadorias.
valor_total	DECIMAL	Valor total das saídas da mercadoria (quantidade * preço unitário).
percentual	DECIMAL	Percentual que o valor_total desta mercadoria representa sobre o total_geral de todas as saídas.
percentual_acumul ado	DECIMAL	Percentual acumulado do valor_total em relação ao total_geral, usado para classificar a curva ABC.
classificacao_ABC	VARCHAR	Classificação da mercadoria na curva ABC ('A', 'B' ou 'C'). Regra: - 'A': Acumulado até 70% - 'B': Acumulado entre 70.01% e 90% - 'C': Acumulado acima de 90%

1.8.2. Modelo Conceitual - Diagrama de Entidade-Relacionamento





1.8.4. Modelo Físico

```
CREATE TABLE IF NOT EXISTS mercadorias (
       ID INT NOT NULL AUTO INCREMENT PRIMARY KEY,
       Nome VARCHAR(100) NOT NULL,
       Quantidade int(30) NOT NULL,
      Itens_caixa int NOT NULL,
       estoque_minimo int NOT NULL,
       estoque_maximo int NOT NULL,
       Volumes float(6,2) NOT NULL,
      Peso_liquido float(6,2) NOT NULL,
       Peso_bruto float(6,2) NOT NULL,
       Largura float(6,2) NOT NULL,
       Altura float(6,2) NOT NULL,
       Profundidade float(6,2) NOT NULL,
       Preco_venda float(6,2) NOT NULL,
       codigo varchar(30) NOT NULL,
       Formato varchar(30) NOT NULL,
       Condicao varchar(30) NOT NULL,
       Tipo varchar(30) NOT NULL,
       Situacao varchar(30) NOT NULL,
       Marca varchar(50) NOT NULL,
       Producao varchar(30) NOT NULL,
       Data_validade varchar(30) NOT NULL,
      Frete_gratis varchar(10) NOT NULL,
       Unidade Medida varchar(30) NOT NULL,
       Variacao Atributo varchar(30) NOT NULL,
       Variacao_Opcao varchar(30) NOT NULL,
       GTIN varchar(30) NOT NULL,
       Departamento varchar(30) NOT NULL,
       Crossdocking varchar(30) NOT NULL,
       Localizacao varchar(30) NOT NULL,
       UNIQUE KEY unique_codigo (codigo)
  ) DEFAULT CHARSET=utf8;
```

```
    ○ CREATE TABLE IF NOT EXISTS estoque (
     codigo_mercadoria INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
      quantidade_atual INT NOT NULL DEFAULT 0,
       data_ultima_atualizacao DATETIME NOT NULL,
      FOREIGN KEY (codigo_mercadoria) REFERENCES mercadorias(Id)
 DEFAULT CHARSET = utf8;
CREATE TABLE IF NOT EXISTS movimentos_estoque (
      id_movimento INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
      codigo_mercadoria VARCHAR(30) NOT NULL,
      tipo_movimento ENUM('ENTRADA', 'SAIDA') NOT NULL,
      quantidade INT(10) NOT NULL,
     preco_unitario_movimento DECIMAL(8, 2) NOT NULL,
      data_hora_movimento DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
      observacoes VARCHAR(255),
      FOREIGN KEY (codigo_mercadoria) REFERENCES mercadorias(codigo)
 DEFAULT CHARSET = utf8;
```

2. Documento de Arquitetura de Software

Questão: Funcionamento - Software de Gestão de Estoque

Fatores

- Alinhamento e comunicação eficiente
- Qualidade no desenvolvimento

Solução

Este documento procura demonstrar o funcionamento do sistema de controle de estoque, trazendo junto dele o padrão de arquitetura estabelecido no projeto e suas visões arquiteturais. O objetivo é facilitar o entendimento para aqueles envolvidos no mesmo.

Produziremos, portanto, um artefato que irá auxiliar na concepção, visualização, implementação e manutenção deste software. Servirá como guia durante todo o processo de criação, nos diferentes papéis exercidos para os envolvidos.

Motivação

Um controle de estoque eficiente é auxilia na gestão de uma loja virtual para atender as demandas necessárias de controle. Evita erros no gerenciamento de mercadorias, facilita a utilização por atuais e futuros funcionários e cria um padrão para o fluxo de entradas e saídas de produtos da loja. Em outras palavras, melhora o tempo gasto nas atividades comuns de controle de estoque e o faz com maior precisão. Para que o projeto tenha esse resultado, é preciso que a equipe tenha claro entendimento dos requisitos, arquitetura, visões e funcionamento do software, tornando o time muito mais coordenado e eficiente.

Representação arquitetural

O modelo de arquitetura estabelecido para o projeto é o de arquitetura em camadas, pois dentre as necessidades mais importantes do cliente, a eficiência e segurança, é o que mais se adequa. Uma maior segurança é estabelecida quando a interface gráfica não tem acesso direto ao banco de dados e com níveis de permissão de usuários (por exemplo, somente um administrador poder alterar dados). Quanto à eficiência, na arquitetura em camadas não precisa escrever o mesmo código várias vezes, tendo menos processamento desnecessário. Outros pontos vantajosos desta arquitetura são: organização e separação de atividades (diferentes interfaces), facilidade de manutenção (corrigir erros somente no código de uma camada) e reutilização de código (caso haja necessidade de criar versão web ou mobile).

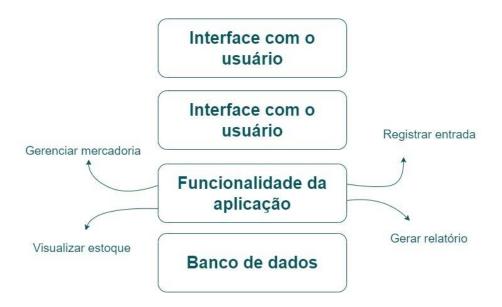


Figura 1 - Padrão de Arquitetura em Camadas

Visão Lógica

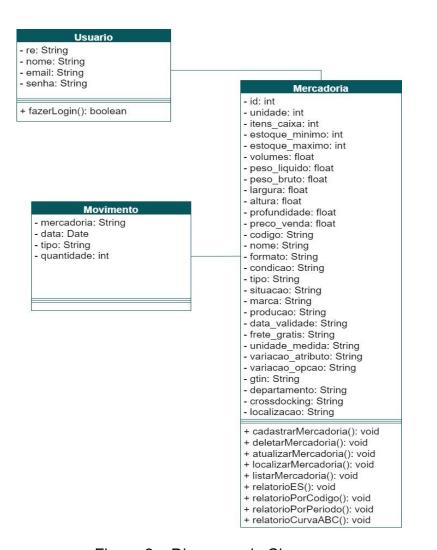


Figura 2 – Diagrama de Classes

Na figura 2 podemos observar todas as classes contidas no projeto, onde a classe 'Usuario' interage diretamente com a classe 'Mercadoria'.

A classe 'Movimento' não possui relação direta com o 'Usuario', isso porque o movimento só é registrado quando o usuário solicita um relatório e os dados do movimento são passados a ele, quando há mudança na quantidade de mercadorias:

- Entrada: feita pelo usuário ao registrar entrada de mercadoria, aumentando sua quantidade em estoque;
- Saída: quando um cliente compra da loja que também está ligada ao banco de dados do software e a quantidade decresce.

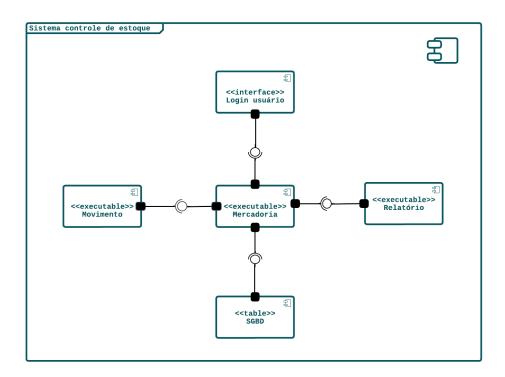


Figura 3 - Diagrama de Componentes

Na figura 3 podemos observar os principais componentes que compõem o sistema, a começar pela interface do <<usuario>> que acessa os componentes de <<Mercadoria>>, <<Movimento>> e <<Relatório>>, todos eles fornecidos pelo componente <<SGBD>>. Este diagrama auxiliará em modelo compreensível da estrutura do software.

As principais vantagens que trazem o diagrama de componentes, são a facilidade que traz em relação à manutenção do sistema e melhorias no software, isso é proporcionado pelo claro mapeamento físico dos elementos do arranjo dos elementos do software.

CADASTRO DE MERCADORIA

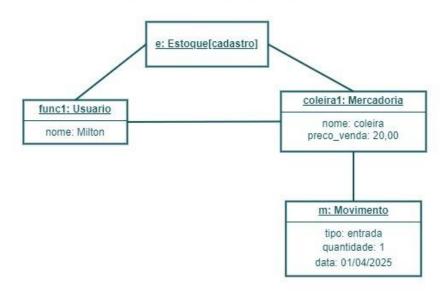


Figura 4 - Diagrama de Objetos

Neste tipo de diagrama, o enfoque é em um determinado momento do sistema, mostrando como os objetos instanciados das classes, interagem no momento daquela ação, e quais atributos de cada objeto participam nesse momento, assim como os valores que recebem esses atributos.

Para melhor entendermos melhor esse diagrama podemos perceber na figura 4 esse padrão, foram instanciados objetos nas classes Estoque, Usuario, Movimento e Mercadoria, o momento que detalhamos a interação foi no Cadastro de mercadorias, e para essa ação foram usados os atributos necessários de cada objeto de classe para efetuar essa ação.

Visão de processo

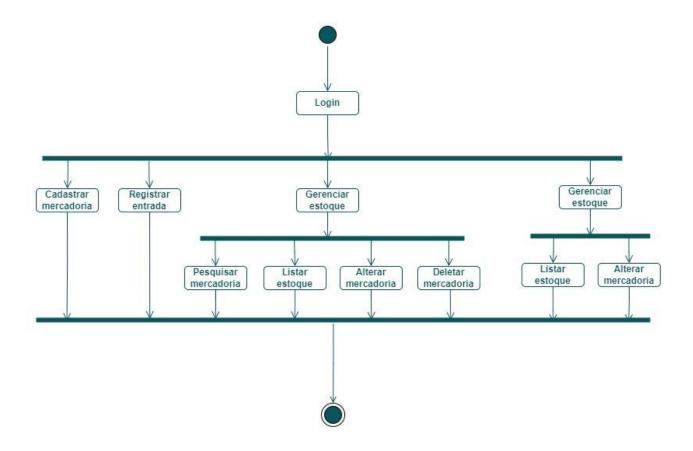


Figura 5 - Diagrama de Atividades

Quando falamos em diagramas comportamentais, um dos principais é o diagrama de atividades, ele é usado para modelar as atividades de um sistema, muito utilizado para mostrar essas atividades em sistemas complexos, melhorando assim a compreensão, do comportamento, excelente para entender a lógica comportamental do sistema.

Como podemos observar na figura 4, é descrito o funcionamento da movimentação do Estoque, como cadastro e entrada de mercadorias, gerenciamento do Estoque usando CRUD, e a geração de relatórios para análise, mostrando assim uma visão global dos comportamentos do sistema.

Diagrama de Sequência Denna

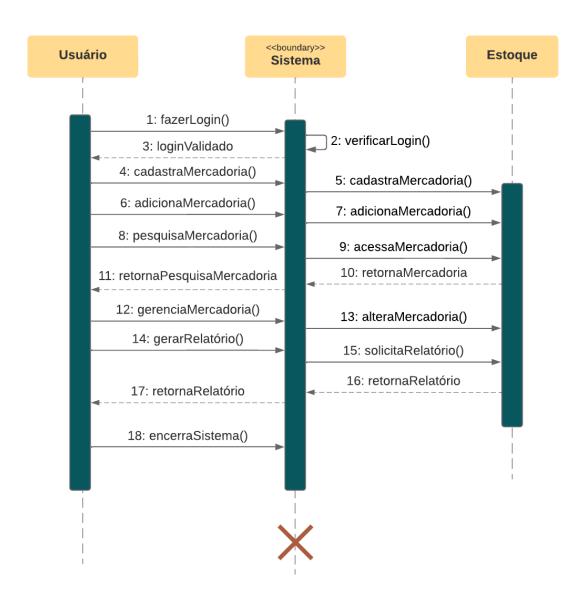


Figura 6 - Diagrama de Sequência

Na figura 6 temos o diagrama de sequência representando a interação do Usuário e as respostas do Sistemas para cada tipo de interação. Também observamos a interação do Sistema com o Estoque (banco de dados), com o objetivo de retornar as requisições do usuário.

Este diagrama nos ajuda a entender o fluxo de processos do sistema, isto é, a resposta esperada após cada interação do Usuário. Ele fornece diretrizes para os processos de testes que deverão ser implementados no sistema.

Visão de implantação

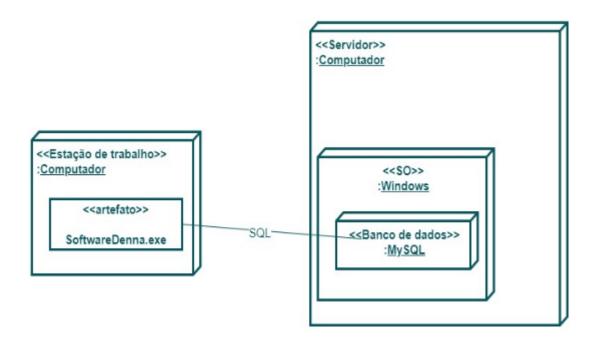


Figura 7 - Diagrama de Implantação

O diagrama de implantação descreve a implementação física de informações geradas pelo programa de software em componentes de hardware. A informação gerada pelo software é chamada de artefato. As principais características desse diagrama são: mostrar quais elementos de software são implementados por quais elementos de hardware, ilustrar o processamento do tempo de execução do hardware, propiciar uma visão da topologia do sistema do hardware.

Na Figura 7 podemos ver a maneira como os componentes de software e hardware se relacionam e se distribuem fisicamente. O software de gestão de estoque está instalado no sistema operacional da máquina do cliente, já o banco de dados está rodando no sistema operacional de outra máquina, em um servidor na nuvem. Ocorre então a ligação entre eles por meio da porta e então os dados contidos no banco de dados (MySQL) são passados para o software para a gestão do estoque da loja.

3. Planejamento da Qualidade de Software

Projeto: Sistema de Gestão de Estoque DENNA

Equipe Desenvolvedora: DENNA

Instituição: Fatec Guarulhos

Data da Versão: 14 de junho de 2025

Versão do Documento: 1.0

3.1 Introdução e Visão Geral do Projeto

3.1.1. Objetivo do Plano

Este Plano de Qualidade Software (PQS) tem como objetivo estabelecer os critérios, as atividades, os padrões e as responsabilidades visando garantir a qualidade do sistema de Gestão de Estoque DENNA.

O propósito é assegurar que o software, desenvolvido pela equipe, atenda ou supere os requisitos funcionais e não funcionais estabelecidos, entregando um produto confiável, eficiente, seguro e intuitivo que agregue valor à microempresa do setor de pet shop.

3.1.2. Escopo do Projeto

O software DENNA é um sistema funcional de controle de estoque de produtos, desenvolvido especificamente para uma loja de artigos para animais domésticos (pet shop). Ele visa otimizar a gestão de estoque, proporcionando:

- Controle de entrada e saída de produtos;
- Geração de relatórios com histórico e tendências;
- Gerenciamento de produtos por Curva ABC
- Sistema de login com hierarquia de permissões e controle de usuários;
- Emissão de alertas;
- Pesquisa e consulta simplificada de produtos e fornecedores;
- Relatórios de movimentação financeira, entrada e saída por depósito/data, e saldo de estoque;

3.1.3. Público-Alvo do Plano

Este documento é direcionado a todos os membros da equipe DENNA (Analistas, Desenvolvedores, Testadores, Projetistas de Banco de Dados, Documentadores), à professora orientadora Jane Maria dos Santos Eberson e ao nosso cliente (professor Milton Francisco de Brito) para fins de validação e entendimento das abordagens de qualidade.

3.2. Objetivos de Qualidade

Os objetivos de qualidade para o sistema DENNA são derivados dos requisitos do cliente e dos requisitos não funcionais identificados.

Confiabilidade:

- Garantir a segurança das informações da loja e de seus clientes;
- Minimizar o risco de prejuízo para a empresa por questões de segurança;
- Assegurar a integridade e consistência dos dados de estoque e vendas;
- Operar sem falhas críticas que resultem em perda de dados de estoque.

Disponibilidade: Garantir que o sistema e os dados gerados estejam disponíveis e acessíveis para usuários autorizados quando eles precisarem.

Usabilidade:

- Oferecer uma interface agradável seguindo princípios Gestalt;
- Apresentar uma interface intuitiva que dê a impressão de continuidade;
- Utilizar ícones sugestivos e segregados na interface;
- Reduzir gastos com treinamento de pessoal, permitindo implementação imediata.

Desempenho:

- Ser capaz de processar grande volume de dados;
- Reduzir o estoque através do desempenho do software, previsão de tendências e geração de relatórios;
- Garantir rapidez no acesso às informações.

Segurança:

- Implementar login obrigatório com hierarquia de permissões;
- Controlar usuários quanto à edição de informações dentro da plataforma.

Manutenibilidade: O código deve ser o mais limpo possível para futuras mudanças.

Portabilidade: Capacidade de ser desenvolvido ou executado em diferentes arquiteturas (hardware/software).

Interoperabilidade: Fácil acesso e comunicação de dados e informações exportados de outros sistemas (Excel, BI, SQL, Access).

3.3. Organização e Responsabilidades da Qualidade

A equipe DENNA é organizada em setores com responsabilidades específicas que contribuem para a qualidade do software.

Análise de Sistemas

Integrantes: Danilo Zamai e Silas Castro Lima.

Responsabilidade: Análise e especificação de requisitos. Garantir que os requisitos funcionais e não funcionais estejam claros, completos, consistentes e testáveis. Realizar validação dos requisitos com o cliente e, na fase final, atuar nos testes de aceitação (RFs e RNFs).

Desenvolvimento

Integrantes: Amanda Yasmin Botto, João Gabriel Lima de Moraes, Lucas Silva Contieri e Mateus Filgueiras Araújo.

Responsabilidade: Desenho e Implementação. Escrever código limpo e seguir padrões de codificação. Realizar testes de unidade em seus próprios desenvolvimentos. Participar de revisões de código e corrigir defeitos.

Projetista de Banco de Dados

Integrantes: Mateus Picoli dos Santos e Priscila Diniz Nascimento.

Responsabilidade: Projeto de Banco de Dados. Garantir a modelagem de dados correta e eficiente para o SGBD Access. Assegurar a integridade, consistência e performance do banco de dados (consultas, armazenamento). Participar das revisões do esquema de banco de dados e auxiliar na investigação de defeitos relacionados a dados.

Testes

Integrante: Paulo César Perez Percinoto Júnior

Responsabilidade: Testes e gestão da qualidade. Elaborar casos de teste com base nos requisitos. Executar testes de integração e sistema. Registrar e acompanhar o ciclo de vida dos defeitos. Validar correções de defeitos.

Documentação

Integrante: Vinícius Feitosa de Araújo

Responsabilidade: Desenvolvimento e integração de toda documentação constante nesse arquivo. Assegurar a clareza, completude e atualização da documentação técnica e de usuário (manual). A documentação de qualidade (este plano) também é de sua responsabilidade na elaboração.

3.4. Atividades por Setor

Análise de Sistemas

Responsabilidade Principal: Fazer o levantamento junto ao cliente do escopo do projeto; requisitos funcionais e não funcionais do software. Criar, com auxílio das outras áreas envolvidas os diagramas do software: diagramas de classe, objeto, sequência, componentes, entre outros. Agendar e realizar as reuniões necessárias para o andamento do projeto junto ao cliente, a fim de que as necessidades do cliente sejam atendidas e atualizadas guando necessário.

Atividades Chave para Qualidade:

Entrevista com Cliente: Agendar entrevista com cliente e todos os participantes do grupo (desenvolvedores, analistas, documentador, banco de dados e teste) com o objetivo de ouvir a proposta de software, bem como da viabilidade do projeto.

Levantamento de Requisitos: Elencar e documentar os requisitos funcionais e não funcionais do software por meio das reuniões e entrevistas com o cliente. Repassar os requisitos com a equipe, em todas as áreas, para que sejam implementados na aplicação.

Criação dos Diagramas: Elaborar os diagramas necessários para concepção e documentação do projeto, utilizando a UML. Fazer alterações nestes artefatos de acordo com as necessidades e alterações. Os diagramas também são relevantes como concepções visuais de diferentes camadas e funcionalidades do software para apresentação ao cliente.

Reuniões de Grupo: Agendamento e realização das reuniões dos envolvidos no projeto (exceto cliente), a fim de verificar o andamento do projeto para cada área envolvida. Verificar as etapas realizadas, as etapas em andamento e as próximas etapas a serem realizadas do software. Estas reuniões foram realizadas, na maior parte das vezes, de forma remota com utilização do programa *Discord*.

Comunicação com o Cliente: Manter o cliente informado quanto ao andamento do projeto, das funcionalidades implementadas, das telas geradas pelos desenvolvedores, da criação e modificação do banco de dados e dos testes realizados. Demonstração visual do projeto para a avaliação do cliente e garantia de que os requisitos são seguidos.

Desenvolvimento

Responsabilidade Principal: Desenvolvimento do código fonte do software.

Atividades Chave para Qualidade:

Boas práticas de codificação: seguir padrões de codificação estabelecidos para facilitar manutenção e reduzir erros.

Revisões de código: verificar o código desenvolvido por outros membros da equipe

Testes unitários: verificar o funcionamento das menores unidades de código (como funções ou métodos) de forma isolada, para garantir que cada unidade funcione como o esperado antes de integrar com outros componentes.

Controle de versão: utilizar sistemas de versionamento para que haja rastreabilidade nas versões e reversões seguras (se necessário).

Documentação do código: Comentar o funcionamento dos principais trechos do código para que outros membros entendam e possam dar continuidade.

Design: Utilizando fundamentos da interação humano computador (IHC), construir interfaces gráficas que sejam intuitivas para o usuário e forneçam uma boa experiência.

Correção de falhas reportadas: Resolver os defeitos identificados por testes ou feedbacks e registrar correções.

Projetista de Banco de Dados

Responsabilidade Principal: Prover e manter a estrutura de dados robusta e eficiente para o sistema DENNA, garantindo a integridade, consistência, segurança e desempenho das informações.

Atividades Chave para Qualidade:

Modelagem de Dados: Elaborar e revisar o modelo de dados para garantir que ele reflita com precisão os requisitos de negócio do pet shop e otimize o armazenamento e recuperação de dados.

Implementação e Otimização: Desenvolver e manter o esquema do banco de dados (tabelas, índices, relacionamentos) no SGBD MySQL, focando na eficiência das operações de leitura e escrita.

Integridade de Dados: Definir e implementar regras de integridade (chaves primárias, chaves estrangeiras, validações) para prevenir dados inconsistentes ou inválidos.

Controle de Versão do Esquema: Gerenciar as evoluções do esquema do banco de dados, garantindo a compatibilidade e a rastreabilidade das mudanças.

Performance: Monitorar e otimizar o desempenho das consultas SQL e operações de banco de dados, assegurando a rapidez exigida pelos requisitos não funcionais.

Segurança: Implementar mecanismos de segurança no banco de dados para controle de acesso e proteção das informações sensíveis.

Suporte a Testes: Colaborar ativamente com os Testadores e Desenvolvedores, se necessário, na criação de dados de teste, na investigação de defeitos relacionados a dados e na validação da persistência correta das informações.

Garantia de Integridade e Consistência dos Dados:

- Chaves Primárias e Estrangeiras: Implementação obrigatória de chaves primárias para identificação única de registros e chaves estrangeiras para garantir a integridade referencial entre as tabelas (Ex: Uma mercadoria não pode existir sem um fornecedor válido, se for o caso).
- Restrições de Coluna: Aplicação de restrições NOT NULL onde os dados são obrigatórios e restrições CHECK (se suportado pelo MySQL para seus tipos de

dados) para garantir valores dentro de um intervalo válido (Ex: Estoque de mercadoria não pode ser negativo).

Testes

Responsabilidade Principal: Se assegurar de que todos os componentes do sistema e sua integração estão funcionando de acordo com o esperado.

Atividades Chave para Qualidade:

Teste de caixa preta (teste de sistema): Criação de tabela de testes ditando todos os testes a ser feito. Estes testes terão como objetivo "simular" várias situações de entradas de dados, incluindo dados válidos e inválidos.

Avaliação do visual do programa, se ele é visualmente agradável e se é fácil de se usar.

Documentação

Responsabilidade Principal: registrar de forma clara todas as etapas do desenvolvimento.

Atividades Chave para Qualidade:

Elaboração dos documentos do projeto: registrar requisitos, manuais e atas de reunião.

Controle de versão e atualização de documentos: atualizar o que for necessário a cada entrega do projeto.

Revisões periódicas: garantir que os documentos estejam corretos, completos e atualizados.



3.5. Processo de Verificação e Validação

Para assegurar um software de qualidade, a equipe deve adotar um processo de verificação e validação que auxiliarão na identificação de falhas em diferentes momentos do desenvolvimento.

- Verificação: tarefas que vão garantir que o software está sendo desenvolvido corretamente, como: revisão de código-fonte, testes de unidade, testes estáticos, revisões técnicas.
- Validação: tarefas que vão garantir se o software realiza ou não aquilo que foi solicitado, como: testes de sistema, testes de aceitação, testes de usabilidade e testes funcionais finais.

3.6. Plano de Garantia de Qualidade

O objetivo deste plano é prevenir erros durante o processo e garantir que o produto atenda aos padrões de qualidades definidos. É a partir das seguintes estratégias que a qualidade do software será assegurada:

Estratégias de verificação: Serão adotadas atividades estáticas de verificação (ações realizadas sem executar o software), como:

- Revisões de requisitos por membros que não os redigiram;
- Inspeção de código-fonte para assegurar uso de boas práticas de codificação;
- Conferência de documentação para verificar sua formatação, clareza e conformidade;
- Checklist de verificação para assegurar que todos os pontos foram analisados.

Estratégias de Validação: Será utilizado testes de caixa preta como teste de sistema para se assegurar de que o programa funcione como desejado.

Casos de Teste: Os casos de teste serão em formato de tabela de testes, onde cada linha na tabela representa um teste específico, e apresenta o passo a passo do teste, resultado esperado, e o resultado do teste. Haverá uma tabela diferente para cada tela do programa onde é possível a entrada e saída de dados.

Responsabilidades: Como o único testador, Paulo César será responsável por realizar todos os testes. As revisões serão feitas com a presença do Documentador para assegurar que os requisitos do cliente foram atendidos

Ferramentas e técnicas utilizadas: Visual Studio para rodar o aplicativo, *Parsec* para poder acessar um banco de dados preenchido, *Discord* para a comunicação entre os membros da equipe.

3.7. Equipes de Revisões e Auditorias

Para garantir a conformidade dos processos e artefatos com os padrões definidos no plano de qualidade, serão conduzidas revisões e auditorias internas.

Revisões: conferência de documentos, requisitos, código-fonte e banco de dados antes da finalização.

Auditorias: avaliações periódicas para verificar se as atividades estão seguindo os padrões e o plano de qualidade definido.

Papéis: cada setor deve indicar um responsável para revisar os artefatos produzidos por outro setor. Isso deve ser feito para garantir a imparcialidade.

3.8. Padrões e Políticas de Qualidade

Análise de Sistemas

Convenções e Padrões de Especificação:

- Padrão: Utilizar padrões formais para a elaboração de documentos de requisitos, como: uso de linguagem clara e não ambígua; estruturação dos requisitos em funcionais e não funcionais e utilização da UML para criação de diagramas.
- Objetivo: Facilitar o entendimento dos requisitos para toda a equipe (principalmente os desenvolvedores) e garantir que tudo esteja alinhado às necessidades do cliente.

Desenvolvimento

Convenções de Nomenclatura Padrão:

- Padrão: Os desenvolvedores devem seguir padrões de codificação como: convenções de nomenclatura para variáveis, métodos, classes e arquivos; código limpo e legível; comentários; padronização do estilo de código e utilização de código sempre que possível
- Objetivo: Facilitar a leitura, compreensão e manutenção do esquema do banco de dados por toda a equipe.

Banco de Dados

Convenções de Nomenclatura Padrão:

- Padrão: Todas as tabelas, campos, índices e outros objetos de banco de dados seguirão convenções de nomenclatura predefinidas (Ex: Nomes de tabelas no plural, campos com nomes descritivos e consistentes, uso de chaves primárias com prefixo 'Id' e chaves estrangeiras com prefixo 'fk').
- ➤ Objetivo: Facilitar a leitura, compreensão e manutenção do esquema do banco de dados por toda a equipe.

Testes

Boas Práticas de Teste:

- Padrão: Seguir um plano de testes estruturado, com casos de teste bem definidos, cobrindo todos os requisitos. Os critérios de aceitação devem estar claros. Registrar os resultados de cada execução e documentar erros encontrados para notificar o setor de desenvolvimento.
- Objetivo: Assegurar um plano de testes sistemático, consistente e rastreável.

Documentação

Convenções e Padrões de Documentação

- Padrão: O documentador deve seguir modelos padronizados (templates) para cada tipo de documentação, manter a linguagem técnica clara e objetiva.
- Objetivo: Facilitar a leitura, compreensão (para toda a equipe), trazer rastreabilidade e facilitar a manutenção dos documentos para o documentador.

3.9. Gerenciamento de Riscos

3.9.1. Identificação de Riscos

Os principais riscos que envolvem a migração e a implementação de um sistema de ERP no contexto de um Petshop são: não ter um plano claro, ignorar o feedback da equipe, não acompanhar as atualizações, ignorar protocolos de segurança.

3.9.2. Análise de Impacto

Esses riscos podem trazer os seguintes impactos:

 Não ter um plano claro: A falta de planejamento pode afetar desde a implementação até o uso do sistema pelos clientes e usuários, se o responsável não tiver um plano definido, acarreta vários prejuízos, tanto financeiros, quanto confiança dos clientes, que é o principal patrimônio de qualquer comercio.

- Ignorar o feedback da equipe: Ao implementar um novo sistema, é se suma importância ouvir as impressões daqueles que vão usar o mesmo, os funcionários começam a descobrir falhas que não foram detectadas pela equipe de testes, ou limitações que não foram tão perceptíveis, e que podem causar problemas operacionais.
- Não acompanhar as atualizações: Deixar de acompanhar as atualizações do software pode causar vários tipos de erros no sistema, pode interferir na acessibilidade aos clientes, divergências em produtos disponíveis bem como suas quantidades em estoque.
- Ignorar protocolos de segurança: Na implementação de um sistema ERP é
 essencial a implementação de um protocolo de segurança, em casos como perca de
 informações, invasões ao sistema, ou de login.

3.9.3. Plano de Mitigação

Não ter um plano claro

- Elaborar um documento detalhado de requisitos com todas as funcionalidades que o sistema precisa ter;
- Criar um cronograma de desenvolvimento, dividindo por etapas: planejamento, desenvolvimento, testes, implantação e suporte;
- Realizar reuniões semanais para revisão de progresso e ajustes;
- Registrar todas as decisões e modificações para manter a organização do projeto.

Ignorar o feedback da equipe

- Realizar entrevistas ou questionários com potenciais usuários (donos ou funcionários de pet shops) para entender suas necessidades;
- Convidar usuários para testar o sistema durante o desenvolvimento e coletar sugestões de melhoria;
- Disponibilizar um espaço para envio de feedback, como um formulário Google ou uma aba no próprio sistema;
- Implementar melhorias com base no feedback recebido, sempre avaliando a viabilidade dentro do escopo do projeto.

• Não acompanhar as atualizações

- Estruturar o sistema de forma modular, facilitando futuras atualizações sem comprometer o funcionamento;
- Documentar todas as funcionalidades e preparar um espaço para registrar futuras melhorias;
- Simular um cronograma de atualização, mesmo que hipotético, para demonstrar manutenção contínua;
- Acompanhar novidades tecnológicas que possam melhorar ou otimizar o sistema no futuro.

• Ignorar protocolos de segurança

- Implementar controle de acesso por meio de login e senha individual para cada usuário:
- Reforçar boas práticas de segurança: criação de senhas fortes e troca periódica de senha;
- Configurar rotinas de backup (ou criar um botão de exportação de dados como demonstração);
- Garantir que cada usuário tenha permissões específicas de acordo com o seu nível de acesso dentro do sistema.

Responsáveis: Todos os funcionários são responsáveis por todos esses aspectos levantados acima, essa é a melhor forma de garantir que todos estejam inteirados e engajados no projeto, e com isso uma maior chance de sucesso no objetivo.

3.10. Métricas e Medições de Qualidade

Serão necessárias métricas para acompanhar a evolução e garantir a qualidade. Possibilitarão avaliar a qualidade dos requisitos e modelos de projeto, o código-fonte e os casos de teste criados no desenvolvimento.

3.10.1. Métricas para Modelo de Requisitos

Após as primeiras etapas do projeto, c<mark>om</mark>o a primeira reunião entre cliente e demais participantes, serão elencados os requisitos desejáveis para o software.

Os requisitos levantados serão importantes para se ter uma ideia do escopo do projeto. Eles servirão de guia para que a equipe siga em conformidade com os objetivos de qualidade do projeto. Será importante que os requisitos sejam claros, verificáveis, que não sejam ambíguos e que sejam concisos para que não gerem dúvidas.

- Número de telas estáticas: 1 (Tela de login);
- Número de telas dinâmicas: 6 (Tela de cadastro, controle de estoque, editar mercadoria, entrada de mercadoria, cadastro de funcionário e relatório);
- Número de objetos de dados persistentes: 2 (Banco de dados e documentação);
- Número de objetos de conteúdo dinâmico: 2 (Mercadoria e Usuário);
- Número de funções executáveis: 6 (cadastrar mercadoria, listar mercadorias, pesquisar mercadoria, deletar mercadoria, editar mercadoria, exibir relatório de E/S).

3.10.2. Métricas de Projeto de Arquitetura

Para este tipo métrica os artefatos produzidos serão os diferentes diagramas do projeto, os quais demonstram a arquitetura do software de forma conceitual.

Estes modelos apresentam os componentes (classes, objetos, tabelas de banco de dados) necessários para o funcionamento, sem a necessidade de se conhecer as propriedades internas do programa ou de seus elementos isolados.

A criação de tais artefatos se dará antes mesmo da criação de seus componentes reais dentro do projeto, com o objetivo de guiar a equipe na sua concepção.

3.10.3. Métricas de Projeto de Interface de Usuário

Desde a primeira reunião com o cliente e os participantes da equipe, esta será uma preocupação para a elaboração do projeto. Desde as cores selecionadas, ícone da aplicação, até a interface serão feitas de acordo com as exigências do cliente e sempre com a sua aprovação. Para a medição deste parâmetro tão subjetivo, serão levados em conta requisitos como complexidade de layout, facilidade de utilização, intuitivo, tempo de reconhecimento, trabalho de digitação, entre outros.

Serão realizadas reuniões de demonstração e reuniões de sprint para demonstração de interface. Estes encontros também serviram para sugestão de mudanças no layout e adequação com exigências do cliente.

3.10.4. Métricas para Teste

Para as métricas de teste será proposta que cada tela do projeto seja verificada pelo membro da equipe responsável por esta atividade, bem como as funcionalidades da tela em questão.

Os erros serão anotados e repassados para a equipe para sua resolução e servirá como medição de qualidade. Esta etapa se concentra nos elementos do projeto em escala reduzida.

Após estas verificações, o software será testado de maneira global, ou seja, como um todo (transições de tela, funcionalidades, layout do programa na tela, possíveis interrupções).



4. Dicionário de Termos

Termo	Checklist		
Descrição	Lista organizada de itens ou tarefas a serem verificados ou concluídos. Ajuda a garantir que nada importante seja esquecido.		
Termo	Crossdocking		
Descrição	Processo logístico onde as mercadorias são recebidas e enviadas rapidamente, sem passar por armazenamento. Reduz tempo e custos na cadeia de distribuição.		
Termo	Design		
Descrição	Planejamento visual que define a aparência e a funcionalidade de um produto ou sistema. Inclui cores, formas e usabilidade.		
Termo	GTIN		
Descrição	Número único que identifica produtos no mercado global, usado para controle e venda. Facilita a leitura em sistemas de código de barras.		
Termo	Layout		
Descrição	Disposição organizada dos elementos visuais em uma página, tela ou ambiente. Ajuda na clareza e facilidade de uso.		
Termo	Login		
Descrição	Processo de autenticação onde o usuário entra em um sistema usando credenciais. Garante acesso seguro às informações.		
Termo	Software		
Descrição	Conjunto de programas e instruções que fazem um computador executar tarefas específicas. Inclui aplicativos, sistemas operacionais e utilitários.		
, and the second			
Termo	Template		
Descrição	Modelo pré-definido usado para facilitar a criação de documentos, sites ou interfaces. Permite manter padrão e economia de tempo.		

5. Considerações Finais

O desenvolvimento do sistema de controle de estoque DENNA representa um esforço conjunto de nossa equipe de alunos da Fatec Guarulhos para atender às necessidades reais de uma microempresa do ramo pet shop. Durante o processo de levantamento de requisitos, modelagem, implementação, testes e documentação, buscouse compreender profundamente os desafios enfrentados pelo cliente e propor uma solução eficiente, segura e de fácil utilização. O esforço da equipe foi o de atender às necessidades e expectativas do cliente e entregar um software de qualidade. Para isto foram empregados os conhecimentos acumulados até aqui no curso de Análise e Desenvolvimento de Sistemas.

O projeto teve como base os princípios da Engenharia de Software, empregando boas práticas de análise, projeto orientado a objetos, arquitetura em camadas, testes sistemáticos e controle de qualidade. Todo o sistema foi pensado para ser intuitivo e acessível, evitando a necessidade de treinamentos complexos e garantindo agilidade no uso diário.

Entre os diferenciais do sistema destacam-se: o controle detalhado de estoque, a geração de relatórios de movimentação e Curva ABC, o gerenciamento de usuários com níveis de acesso, e a integração com banco de dados relacional para garantir confiabilidade e consistência das informações.

Com isso, encerramos este ciclo de desenvolvimento com a certeza de termos entregue uma solução alinhada aos objetivos do cliente e aos princípios de qualidade do curso de Análise e Desenvolvimento de Sistemas.



6. Referências

https://www.agentepet.com.br/

https://softdesign.com.br/

https://www.umov.me/

https://www.totvs.com/blog/negocios/curva-abc/

https://www.sankhya.com.br/blog/curva-abc/

https://pdvcontent.com.br/curva-abc/

https://www.oracle.com/

https://learn.microsoft.com/pt-br/visualstudio

https://github.com/

