

CENTRO PAULA SOUZA
FACULDADE DE TECNOLOGIA DE ITAPETININGA
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS

BEATRIZ MARTINS DE OLIVEIRA
CAYOS DA SILVA SOUZA
DIOGO LIMA GOMES DE OLIVEIRA
KEROLIN FERREIRA DE OLIVEIRA
YURI YAMAGUCHI DE AGUIAR

DOCUMENTAÇÃO

Itapetininga, SP
2º Semestre/2025

BEATRIZ MARTINS DE OLIVEIRA
CAYOS DA SILVA SOUZA
DIOGO LIMA GOMES DE OLIVEIRA
KEROLIN FERREIRA DE OLIVEIRA
YURI YAMAGUCHI DE AGUIAR

DOCUMENTAÇÃO

Projeto submetido a disciplina de Laboratório de Engenharia de Software do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas da Faculdade de Tecnologia de Itapetininga sob orientação do Prof.^a Maylon Pires Macedo

Itapetininga, SP
2º Semestre/2025

LISTA DE FIGURAS

Figura 1 - Rotina MATA030	5
Figura 2 - Diagrama de atividade apresentando o fluxo atual do processo	10
Figura 3 - Protótipo de Tela de Cadastro	13
Figura 4 - Diagrama de abertura de pedidos.....	16
Figura 5 - Backlog na data de 24 de setembro extraído do JIRA.....	20
Figura 6 - Backlog na data de 12 de novembro extraído do JIRA	21
Figura 7 - Protótipo de tela de cadastro de cliente	28
Figura 8 - Protótipo de tela de cadastro de produto	29
Figura 9 - Protótipo de tela de cadastro de pedido	30
Figura 10 - Diagrama de Atividade do novo fluxo de processo.....	36

LISTA DE QUADROS

Quadro 1 - Anotações após primeira observação	8
Quadro 2 - Questionário aplicado	10
Quadro 3 - Requisitos Funcionais: RF-01	13
Quadro 4 - Requisitos Funcionais: RF-02	14
Quadro 5 - Requisitos não funcionais sistêmico	14
Quadro 6 - Documentação de caso de uso: CRUD Cliente	16
Quadro 7 - Documentação de CRUD Produto	17
Quadro 8 - Documentação de CRUD Pedido	18
Quadro 9 - Documentação de caso de uso: Envio de Informativo	19
Quadro 10 - Diário de Bordo de 03 de setembro	22
Quadro 11 - Diário de Bordo de 11 de setembro	23
Quadro 12 - Diário de Bordo de 17 de setembro	24
Quadro 13 - Diário de Bordo de 24 de setembro	25
Quadro 14 - Diário de Bordo de 29 de outubro	26
Quadro 15 - Diário de bordo de 05 de novembro	26
Quadro 16 - Diário de bordo de 12 de novembro	26
Quadro 17 – Descrição de arquitetura	33
Quadro 18 - Teste de qualidade aplicados no sistema	34

SUMÁRIO

1 PROBLEMA MOTIVO	5
1.1 DETALHAMENTO DO CENÁRIO	6
2 LEVANTAMENTO DE REQUISITOS	8
2.1 TÉCNICA 1: OBSERVAÇÃO	8
2.2 TÉCNICA 2: DESENHO DO PROCESSO	9
2.3 TÉCNICA 3: QUESTIONÁRIO	10
2.4 TÉCNICA 4: ENTREVISTA	12
2.5 TÉCNICA 5: DESENHO DE INTERFACE	12
2.6 REQUISITOS COLETADOS	13
3 DIAGRAMAS DE CASO USO	16
4 BACKLOG	20
5 DIÁRIOS DE BORDO	22
6 DISPONIBILIZAÇÃO DO MVP	28
7 ASPECTOS DO SISTEMA	31
7.1 ARQUITETURA UTILIZADA	31
7.2 TECNOLOGIAS E INOVAÇÕES	31
7.3 DIAGRAMA DE ARQUITETURA DO SISTEMA	33
7.3.1 Design System	33
7.4 TESTES DE QUALIDADE	34
7.5 FLUXO DA APLICAÇÃO	35

1 PROBLEMA MOTIVO

O diretor da MaicoSoft Importados está com um sério problema com sua equipe de oportunidades de vendas:

Todos os dias, um dos 3 funcionários reclamam que levam muito tempo para cadastrar um novo cliente, pois o Protheus tem muitos campos (espalhados) e eles não os encontram com facilidade;

Eles também disseram que tem dificuldade de encontrar os campos obrigatórios e mandatórios, o que tem gerado muitas falhas por cadastro incompleto;

Eles reclamam que a tela é muito ruim e não ajusta conforme a necessidade deles, sendo necessário o uso constante do mouse para conseguir trabalhar nessa tela.

Além das reclamações dos funcionários, o diretor da MaicoSoft também tem outras duas demandas:

Ele ouviu dizer que utilizando API, ele não precisa de licença TOTVS para todos os funcionários, então ele quer se beneficiar dessa possibilidade;

Ele quer receber um e-mail após cada cadastro efetuado no Protheus para saber quais clientes ele já pode solicitar a emissão de um pedido.

A tela em questão é da rotina MATA030 do módulo de materiais do Protheus (segue abaixo):

Figura 1 - Rotina MATA030

The screenshot displays the 'TOTVS | Compras' interface for the 'Clientes - INCLUIR' (Client Registration) routine. The form is organized into several columns and rows, each containing specific data entry fields. Key sections include:

- Header:** 'TOTVS | Compras' and 'Clientes - INCLUIR' with a status bar showing 'TOTVS Construção e Projetos MSSQL Totvs_pr1212310_rest', 'Administrador', '08/10/2024', and 'Tela / Matriz'.
- Navigation:** A top bar with tabs for 'Cadastro', 'Adm/Fin.', 'Fiscalis', 'Vendas', and 'Outros'.
- Form Fields:** Numerous input fields for client data, including 'Codigo', 'Nome', 'Endereço', 'Telefone', 'CNPJ', 'CEP', 'Município', 'Estado', 'País', 'Status', 'Data Cadastro', 'Descrição', 'Nome Fantasia', 'Insc. Rural', 'Complemento', 'Loja C&P', 'Cod. Test', 'Tipo', 'RG/Cad. Extr.', 'DL/Identific.', 'Cod. CBO', 'Cod. Segmento', and 'Tipo Membro'.
- Buttons:** 'Outras Ações', 'Cancelar', and 'Salvar' buttons are visible at the top right.

1.1 DETALHAMENTO DO CENÁRIO

Há uma perda de tempo e, por consequência, um gasto e uma quebra de produtividade, na rotina de cadastro de clientes da empresa MaicoSoft. Hoje, a equipe de vendas, que atua em campo, faz a prospecção de clientes e encaminha os dados que consegue para a equipe de oportunidade de vendas, usuários efetivos da tela de cadastro; eles, então, buscam as informações faltantes para o cadastro no sistema. O cadastro acontece com o preenchimento manual de diversos campos; copiando informações de outros locais e preenchendo informações que, muitas vezes, são redundantes. Por derradeiro, nota-se que a equipe de vendas depende das informações recolhidas pela equipe de campo; se as informações forem incorretas ou parcas, o cadastrar pode resultar incorreto ou ser impossível.

Quanto aos campos do sistema, mesmo os campos considerados obrigatórios pela documentação podem estar sendo redundantes e gerando um excesso de trabalho. Por exemplo, o campo de ‘Código’ é incremental, não precisando ser digitado manualmente; os itens ‘Loja’ e ‘Pessoa Física ou Jurídica’ são fixos, não variam independentemente do cadastro, pois é sempre usada a mesma loja como referência e as vendas são feitas apenas para pessoas jurídicas.

Outros campos podem ser otimizados para auxiliar o processo de cadastro, transformando a rotina de cadastro em algo mais ágil. Os dados do cliente que forem vinculados ao CNPJ podem ser obtidos dinamicamente da internet (usando de uma API para recolher os dados da Receita Federal ou de um banco próprio), assim como os dados de endereço, normalmente vinculados ao CEP, podem ser parcialmente preenchidos automaticamente (usando uma API para recolher os dados dos Correios ou de banco próprio), reduzindo os campos a serem preenchidos manualmente no total.

Conforme visto na observação, o tempo para cadastro de um novo cliente pode ser estimado em 03 minutos e 45 segundos, sendo a processo dividido entre consulta de informações, cadastros e correções de erros de digitação ou de preenchimento do sistema.

Em um cenário ideal, seria possível a otimização do processo reduzindo as consultas em outras telas, minimizando os erros de cadastro e melhorando a forma de preenchimento das informações. Serve como exemplo dessa última o cadastro do número do CNPJ: o campo rejeita constantemente valores copiados e não permite mudanças dinâmicas para correção do erro, além disso a base não reconhece como distintos CNPJ's que usam a mesma raiz.

Um novo tempo para cadastrado, considerando a aplicação exitosa das atualizações proposta, seria de menos 02 minutos; isso seria possível com a aplicação de campos atualizados dinamicamente e redução das consultas de outras informações. O usuário, então, poderia se dedicar ao processo de venda e comunicação, reduzindo o tempo perdido na tela de cadastro de clientes e ganhando em produtividade ao final do dia. O melhor cadastro também reduziria o retrabalho durante o funil de venda, trazendo benefícios para os demais times que participam do processo de venda.

2 LEVANTAMENTO DE REQUISITOS

O levantamento de requisitos foi realizado de a partir de cinco técnicas: observação; desenho do processo; questionário; entrevista; desenho de interface.

2.1 TÉCNICA 1: OBSERVAÇÃO

A observação foi realizada em dois momentos distintos, primeiro em discussão com o responsável para entender como é o processo para recolhimento de dados; e depois em observação do processo efetivo de cadastro das informações. Foi possível observar que uma das grandes demoras do processo de cadastro é a verificação de informações, que é consultada de fora do sistema.

Foram realizadas duas observações do processo de cadastro, a primeira junto de uma explicação e a segunda demonstrando como ocorre efetivamente o cadastramento de clientes; para fins de otimização, será usado como referência o segundo tempo por ser mais fiel a realidade: 03 minutos e 45 segundos.

No quadro abaixo são apresentadas as anotações iniciais, recolhidas durante a observação.

Quadro 1 - Anotações após primeira observação

<p>Equipe de campo recolhe contatos para a oportunidade de vendas, os vendedores mandam a informação como é possível;</p> <p>Muitas vezes, faltam informações ou estão incorretas;</p> <p>A equipe de oportunidade de vendas recolhe os dados antes do processo de vendas de fato, buscando agilizar a tarefa futura. A equipe normalmente fica no escritório mesmo;</p> <p>A tela não é possível, ela apenas corta as informações.</p> <p>Cadastro 1:</p> <p>Tempo: 05:20 – Durante a explicação do uso;</p> <p>Consulta de CNPJ em uma tela externa para recolher algumas informações (CNPJ.ws com API);</p> <p>Código é livre, mas não é sequencial como deveria; usuário perde tempo confirmando o código;</p> <p>Copia a razão social;</p> <p>Confirma a natureza do cliente;</p> <p>Insere endereço;</p> <p>Nome fantasia;</p>
--

Tipo é um dado constante, 'F';
Abre consulta do município dentro do Totvs, para então extrair o código;
Depois volta a copiar o as coisas do CNPJ.ws;
Campo de CNPJ não funciona corretamente;
Questiona se pode usar a mesma base que outros CNPJ's?
Ele pergunta se autoriza cadastro errado novamente.
Cadastro 2: 03:45;
Tempo:
Joga o CNPJ no site;
Conta o número atual do cliente;
Loja é sempre 01;
Questiona se pode usar a mesma base que outros CNPJ's?
Tipo é um dado constante;

Fonte: Autoria própria (2025)

É possível perceber que durante o cadastro múltiplas consultas a informações externas são realizadas ao longo processo, aproximadamente 01 minutos e 01 segundos, o que seria em torno de um terço do tempo integral. Uma otimização na busca de informações poderia reduzir esse tempo ou até mesmo extinguir.

Algumas dificuldades relativas ao processo e a interface foram encontradas na observação, são elas:

Inserção do CNPJ, o sistema recusa constantemente a escrita dos números, dificultando o cadastro rápido desse dado;

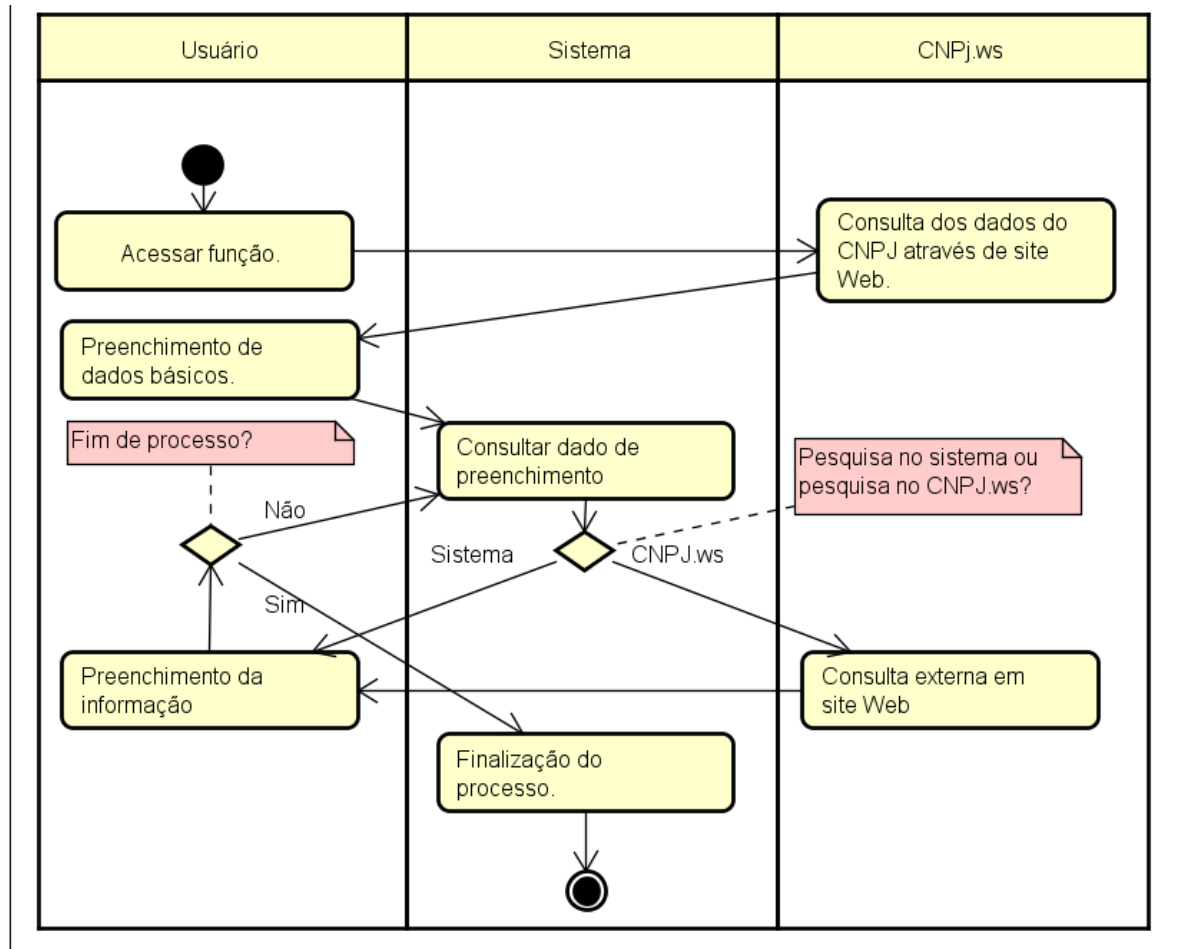
Aceitação do CNPJ, o sistema questiona constantemente se o CNPJ é o mesmo quando a raiz é mesma;

Extração do código de município, o sistema dificulta a inserção do município do através da lista interna.

2.2 TÉCNICA 2: DESENHO DO PROCESSO

O processo, conforme observado, é padrão e natural. Ao longo do cadastro, o usuário vai consultando informações externas e preenchendo no sistema ou buscando no próprio sistema. O problema real está nas informações recolhidas, pois são dados que poderiam ser extraídas automaticamente, reduzindo o tempo final de preenchimento. Outra questão, conforme apresentado na observação, é a frequência de recusas do sistema ao CNPJ e a dificuldade para buscar a cidade dentro da base externa.

Figura 2 - Diagrama de atividade apresentando o fluxo atual do processo



powered by Astah

Fonte: Autoria própria (2025)

2.3 TÉCNICA 3: QUESTIONÁRIO

Com base no processo observado e buscando criar pauta para as entrevistas necessárias para assimilar os requisitos do projeto. O questionário e as respostas estão apresentados no quadro 2:

Quadro 2 - Questionário aplicado

1. Poderia descrever em poucas palavras qual a dinâmica de trabalho dos usuários que farão uso do sistema?
R.: Como indicado durante a aula, a equipe de oportunidades de vendas recebe as informações da equipe de vendas que está em campo prospectando novos clientes.
2. Quantos usuário farão uso recorrente desse cadastro? O estudo apresenta 3 usuários que reclamam do tempo de cadastro, mas outros funcionários também usam essa parte do sistema?
R.: 3. Somente 3.

3. Alguns campos foram marcados como obrigatórios, os demais campos são opcionais ou descartáveis? O campo Home-Page, por exemplo.
R.: Os campos destacados na documentação enviada são obrigatórios. Atenha-se a documentação.
4. O cadastro será realizado em um ambiente com algum impedimento de hardware; por exemplo, será usado em um local sem mouse?
R.: Será no mesmo local de trabalho atual
5. O diretor da empresa gostaria de saber sobre atualizações de cadastros também ou apenas o surgimento de novos cadastros?
R.: Sim, conforme indicado na documentação.
6. Existem outros funcionários interessados em receber essas informações?
R.: Somente o que foi indicado.
7. Qual a demanda para o uso de celular, mobile, para a tela de cadastro?
R.: Você pode propor novas ideias.
8. Existe demanda para o treinamento dos usuários atualmente ou após a criação dessa nova tela?
R.: Seria bom. Vocês oferecem isso nesse projeto?
9. Qual a frequência para cadastro de novos clientes?
R.: Diário. Cerca de 20 novos cadastros por dia.
10. Quanto aos dados em JSON:
a. "loja:";"01"; existe plano de expansão para a loja?
R.: Sua pergunta não ficou clara.
- b. O CNPJ, CEP e telefone, precisam ser processados sem traço ou existe alguma demanda quanto a formatação?
R.: Precisa ver isso com o cara do banco de dados.
- c. "abertura": "2020-1-1", precisa respeitar algum formato para outro processamento?
R.: Precisa ver isso com o cara do banco de dados.
11. São feitas vendas para outros países?
R.: No momento não
12. É interessante a apresentação de uma tela de protótipo para o projeto?
R.: Não precisa.
13. Quais os benefícios esperados de uma solução bem-sucedida?
R.: Que diminua o tempo do cadastro e os erros por cadastros errados.

Fonte: Autoria própria (2025)

Algumas informações fundamentais para o dimensionamento foram extraídas do questionário. O escopo apresentado na documentação parece ser muito fiel as necessidades reais

da empresa, não alterando o número de usuários do sistema e nem de gestores que precisam receber e-mail. Foi informado também que a quantidade de cadastros, e por consequência de requisições da API, são apenas 20 por dia; o que torna possível o uso de funcionalidades gratuitas que tem uma restrição na quantidade de usos.

2.4 TÉCNICA 4: ENTREVISTA

Com base no questionário apresentado ao cliente, surgiram alguns questionamentos a respeito da aplicação a ser desenvolvidas. As novas perguntas que emergiram foram:

1. No momento as vendas ocorrem apenas para clientes PJ ou atende também pessoas físicas? Existe alguma possibilidade de expansão?

Não.

2. Existe algum plano de abertura de novas lojas que usarão a mesma tela de cadastro?

Não.

3. Tem algum campo que gostariam de sugerir para o cadastro dos clientes?

Não.

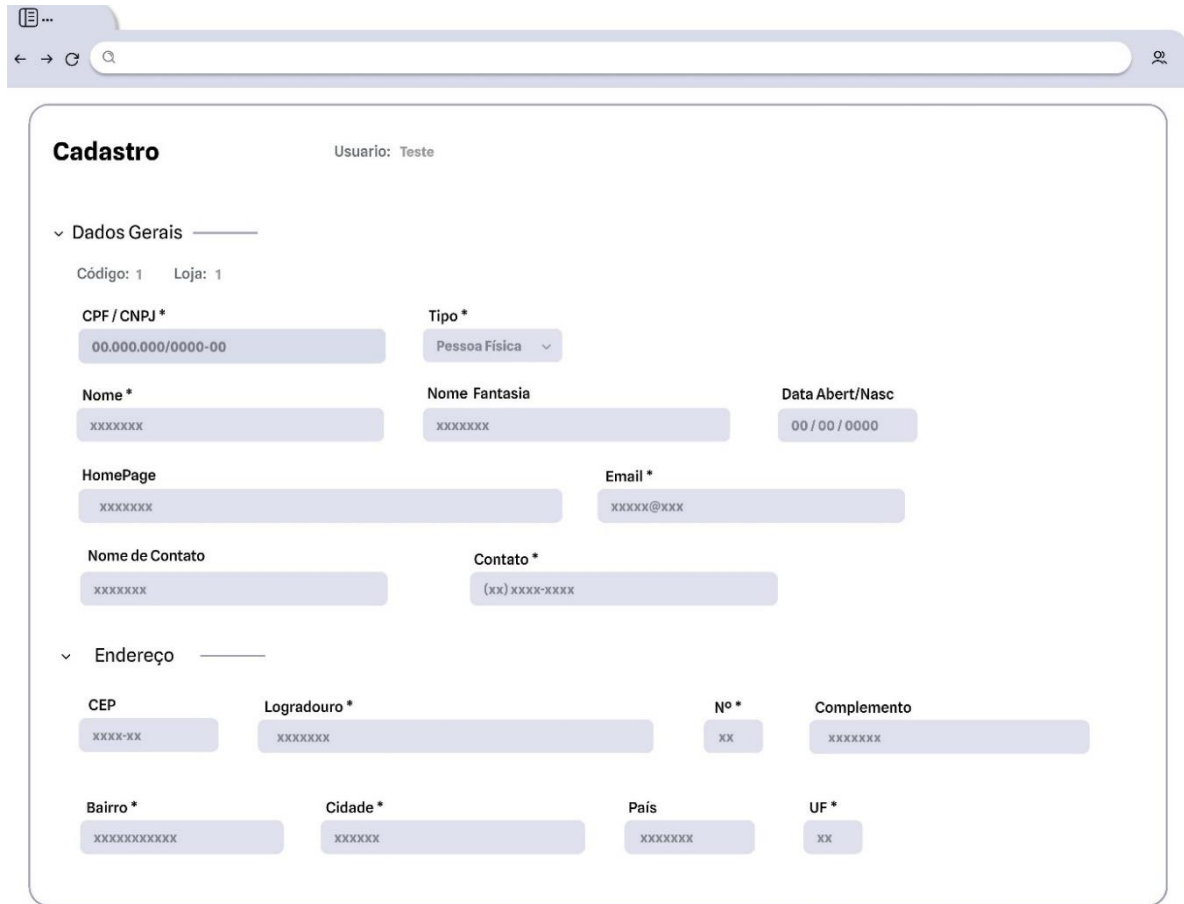
4. Quais meios de contato precisam ser obrigatórios?

O maior contato é por telefone, mas o e-mail é obrigatório para informações oficiais.

2.5 TÉCNICA 5: DESENHO DE INTERFACE

Foi desenvolvido um desenho de interface para servir como esboço para o desenvolvimento da tela final. A figura 2 abaixo, então, é o modelo de tela a ser desenvolvido em HTML, CSS e JavaScript, já estão delimitados os campos obrigatórios, com o asterisco, e os campos opcionais; tendo ainda dois campos permanentes que referentes ao código do cadastro e a loja.

Figura 3 - Protótipo de Tela de Cadastro



O protótipo da tela de cadastro apresenta uma interface limpa com uma barra de navegação superior. O título principal é 'Cadastro', e o usuário logado é 'Usuário: Teste'. A seção 'Dados Gerais' contém campos para 'Código' e 'Loja', seguidos por 'CPF / CNPJ *' e 'Tipo *' (com uma lista suspensa selecionando 'Pessoa Física'). Abaixo, há campos para 'Nome *', 'Nome Fantasia', 'Data Abert/Nasc' (formato 00/00/0000), 'HomePage' e 'Email *'. A seção 'Endereço' inclui campos para 'Nome de Contato', 'Contato *' (formato (xx) xxxx-xxxx), 'CEP', 'Logradouro *', 'Nº *', 'Complemento', 'Bairro *', 'Cidade *', 'País' e 'UF *'. Todos os campos de texto possuem máscaras de entrada para garantir o formato correto dos dados.

Fonte: Autoria própria (2025)

2.6 REQUISITOS COLETADOS

A nova aplicação tem dois requisitos funcionais básicos, o primeiro referente ao cadastro do cliente e o segundo referente ao envio do informativo por e-mail ao diretor. O quadro 3 detalha o ‘Requisitos Funcionais RF-01’, apresentado sua descrição e restrições.

Quadro 3 - Requisitos Funcionais: RF-01

REQUISITOS FUNCIONAIS		
RF-01	Criar, listar, atualizar e deletar cliente.	
Descrição detalhada do Requisito Funcional:		
Ele deve permitir o cadastro de um novo cliente sem necessidade de acesso e/ou busca de informações externas, buscando de maneira automática os dados relacionados ao endereço, através do CEP, e os dados da empresa através do CNPJ; posteriormente, o sistema deve permitir que a leitura, a criação e a limpeza da mesma.		
REQUISITOS NÃO FUNCIONAIS RELACIONADOS		
Descrição detalhada da regra/restrição	Natureza	Tipo
RNFR-01.1 O cadastro será feito pela equipe de oportunidade de vendas.	Implementação	Obrigatório

RNFR-01.2	O sistema de realizar o preenchimento automático de dados relacionados a empresa e CNPJ.	Entrega	Desejável
RNFR-01.3	Usuários deverão operar o sistema após um determinado tempo de treinamento.	Facilidade de uso	Obrigatório
RNFR-01.4	A solução deverá ser desenvolvido como uma API em Java com SpringBoot..	Implementação	Obrigatório

Fonte: Autoria própria (2025)

Logo abaixo, no quadro 4, segue o requisito funcional referente ao envio do e-mail para o diretor da MaicoSoft, ‘Requisitos Funcionais RF-02’, sua descrição e as restrições.

Quadro 4 - Requisitos Funcionais: RF-02

REQUISITOS FUNCIONAIS			
RF-01	Envio de Informativo.		
Descrição detalhada do Requisito Funcional:			
Ele deve realizar o envio de um e-mail ao endereço cadastrado que contenha os dados colocados no cadastro.			
REQUISITOS NÃO FUNCIONAIS RELACIONADOS			
Descrição detalhada da regra/restrrição		Natureza	Tipo
RNFR-01.1	O envio deverá ser feito automaticamente, após o final de um novo cadastro.	Implementação	Obrigatório
RNFR-01.2	O envio deverá ser feito 99% das vezes.	Confiabilidade	Obrigatório
RNFR-01.3	O processo de envio deve ser feito através de API JavaScript..	Implementação	Obrigatório

Fonte: Autoria própria (2025)

Por fim, no quadro 5, estão apresentados os requisitos não-funcionais sistêmicos; eles totalizam nove requisitos e descrevem algumas características referentes ao funcionamento do sistema como um todo.

Quadro 5 - Requisitos não funcionais sistêmico

REQUISITOS NÃO FUNCIONAIS SISTÊMICOS			
Descrição detalhada da regra/restrição		Natureza	Tipo
RNFS-01	O sistema deve ser utilizável em qualquer navegador ou dispositivo	Portabilidade	Obrigatório
RNFS-02	O sistema deve respeitar a LGPD	Legal	Obrigatório
RNFS-03	O sistema deverá seguir todas as regras de acessibilidade digital	Acessibilidade	Obrigatório
RNFS-04	O sistema não deverá demorar mais de 5 segundos para trazer os resultados das buscas	Desempenho	Desejável
RNFS-05	O sistema deverá se comunicar com o ERP Protheus	Implementação	Desejável
RNFS-06	O sistema deverá ser feito em plataforma aberta	Implementação	Desejável
RNFS-07	O sistema será desenvolvido usando: Javascript, CSS, HTML5 e Java com SpringBoot.	Interoperabilidade	Obrigatório
RNFS-08	O sistema deverá estar disponível 98% do tempo	Disponibilidade	Desejável

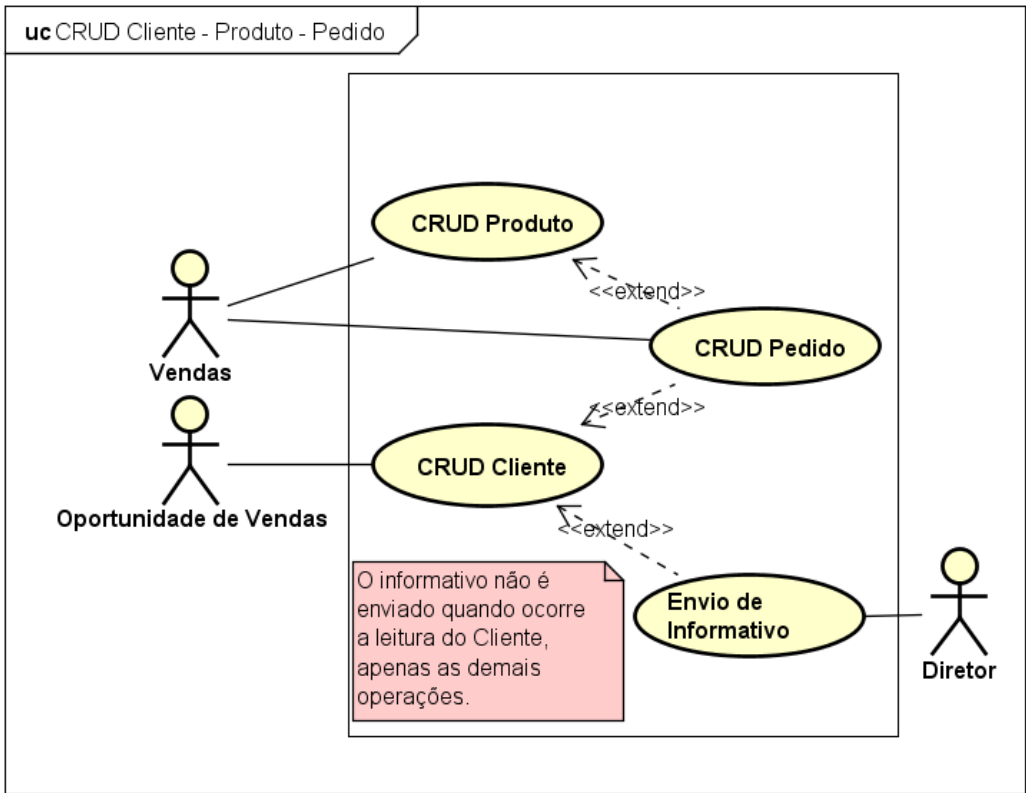
RNFS.09	O sistema deverá seguir o clean code durante sua progressão, manutenção implementação	Padrão	Obrigatório
----------------	---	--------	-------------

Fonte: Aatoria própria (2025)

3 DIAGRAMAS DE CASO USO

Na figura 4 é apresentado o Diagrama de Caso de Uso da aplicação que abrange a Manutenção dos Clientes pela equipe de Oportunidade de Vendas e a Manutenção de Produtos e Pedidos pela equipe de Vendas. Nesse contexto, é importante apresentar que foi respeitada a demanda da empresa de encaminhar um informativo a cada alteração no cadastro de um cliente e quando eles são apagados.

Figura 4 - Diagrama de abertura de pedidos



powered by Astah

Fonte: Autoria Própria (2025)

Logo em seguida, é apresentada a documentação referente a cada caso de uso apresentado no diagrama: ‘CRUD’ Cliente, ‘CRUD’ Pedido, ‘CRUD’ Produto e Envio de Informativo.

No CRUD de Cliente, conforme quadro 6, é explicado o funcionamento da API que realiza o preenchimento das informações básicas de cadastro.

Quadro 6 - Documentação de caso de uso: CRUD Cliente

Nome do Caso de Uso	CRUD Cliente
Ator Principal	Usuário (Oportunidade de Vendas)

Atores Secundários	
Resumo	Esta tela realiza todo o processo de cadastro de cliente
Pré-Condições	Possuir acesso a API
Pós-Condições	Registrar no banco novo cliente
Fluxo Principal	
Ações dos Atores	Ações do Sistema
Abertura da função	Criação do ID, e alocação dos dados default loja e país do cliente.
Inserção do CNPJ	Busca do CNPJ através de API
Inserção de CEP	Busca do CNPJ através de API
Inserção de dados opcionais e não extraídos da API	
Finalização	Registro dos dados no banco de dados, chamada do caso de uso Envio de Informativo
Restrições / Validações	Preenchimentos dos campos obrigatórios
Fluxo Alternativo I – Campos obrigatórios em branco	
Ações do Atores	Ações do Sistema
	Recusa do cadastro pelo sistema

Fonte: Autoria Própria (2025)

O sistema ainda abrange um CRUD de produto destinado aos produtos produzidos e vendidos pela empresa, conforme quadro 7, que usa de recursos básicos de Inteligência Artificial para gerar as melhores descrições para o produto.

Quadro 7 - Documentação de CRUD Produto

Nome do Caso de Uso	CRUD Produto
Ator Principal	Usuário (Vendas)
Atores Secundários	
Resumo	Este tela realiza todo o processo de cadastro de produto
Pré-Condições	Acesso a API do Gemini
Pós-Condições	Registrar no banco novo produto
Fluxo Principal	
Ações dos Atores	Ações do Sistema
Abertura da função	Criação do ID incremental
Preenchimento dos dados básicos do produto	Criação da descrição do produto
Aceitação da descrição	Finalização do cadastro
Finalização	Registro dos dados no banco de dados
Restrições / Validações	Preenchimentos dos campos obrigatórios

Fluxo Alternativo I – Campos obrigatórios em branco	
Abertura da função	Criação do ID incremental
Preenchimento dos dados básicos do produto	Criação da descrição do produto
Alteração do código de descrição	

Fonte: Autoria Própria (2025)

Usando como base o cadastro de cliente e de pedido é possível realizar o cadastro de um pedido com múltiplos itens, conforme documentação apresentado no quadro 8.

Quadro 8 - Documentação de CRUD Pedido

Nome do Caso de Uso	CRUD Pedido
Ator Principal	Usuário (Oportunidade de Vendas)
Atores Secundários	
Resumo	Esta tela realiza todo o processo de cadastro de pedido
Pré-Condições	Cadastro de clientes e produtos
Pós-Condições	Registrar no banco novo pedido
Fluxo Principal	
Ações dos Atores	Ações do Sistema
Abertura da função	Criação do ID.
Busca de cliente	Resgate das informações no banco de dados
Busca de produto	Resgate das informações no banco de dados
Seleção de forma de pagamento e prazo de entrega e gravação	Validação dos campos obrigatórios e gravação no banco
Finalização	Registro dos dados no banco de dados
Restrições / Validações	Preenchimentos dos campos obrigatórios.
Fluxo Alternativo I – Campos obrigatórios em branco	
Ações do Atores	Ações do Sistema
Abertura da função	Criação do ID.
Busca de cliente	Resgate das informações no banco de dados
Busca de produto	Resgate das informações no banco de dados
Seleção de forma de pagamento e prazo de entrega e gravação	Validação dos campos obrigatórios e recusa no cadastro
Preenchimento de campos obrigatórios	

Fonte: Autoria Própria (2025)

O caso de Envio de Informativo é mais simples, pois se tratar apenas do envio dos dados do cliente como um informativo por e-mail para um endereço registrado, vide quadro 9; é importante registrar que nesse modelo o ator principal é o próprio sistema.

Quadro 9 - Documentação de caso de uso: Envio de Informativo

Nome do Caso de Uso	Envio de Informativo
Ator Principal	Sistema
Atores Secundários	
Resumo	Este diagrama realiza todo o processo de envio dos dados do novo cliente para o endereço de e-mail pré-registrado.
Pré-Condições	Registro de um e-mail.
Pós-Condições	Recebimento da mensagem.
Fluxo Principal	
Ações dos Atores	Ações do Sistema
Consulta do e-mail	Retorno do E-mail
Envio do e-mail.	
Restrições / Validações	

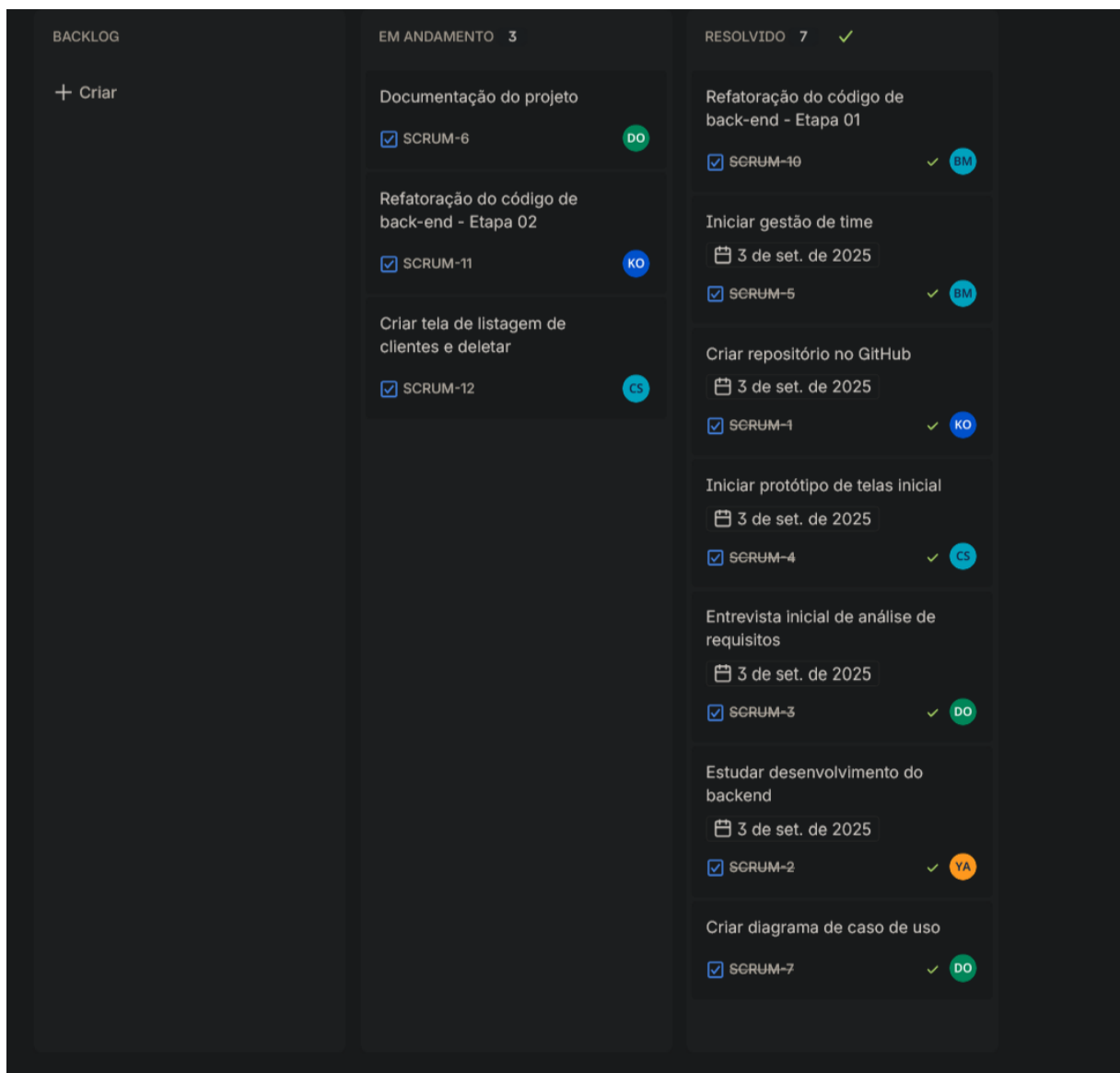
Fonte: Autoria Própria (2025)

4 BACKLOG

O backlog, assim como seu histórico e resumo, pode ser acessado pela plataforma Jira, através do link: <https://fatec-team-bai5w8kf.atlassian.net/jira/software/projects/SCRUM/boards/1?atlOrigin=eyJpIjoiNmY3YzhiNmE4NzdiNDhlNGIwYjU4OTMyODRjZmRkYWUiLCJwIjoiajI9>.

A seguir segue imagem do Backlog para na data de 24 de setembro, antes da finalização do projeto e, por consequência, antes da finalização desta documentação.

Figura 5 - Backlog na data de 24 de setembro extraído do JIRA



Fonte: Autoria Própria (2025)

Na figura 6 é apresentada a tela da segunda Sprint que desenvolveu as telas de cadastro de produto e pedido, ela apresenta muitas outras atividades uma vez que compreende a finalização do projeto.

Figura 6 - Backlog na data de 12 de novembro extraído do JIRA

The screenshot displays a JIRA backlog interface. On the left, there is a large empty column labeled '+ Criar'. The main area shows a list of 15 tasks, each represented by a card. The tasks are as follows:

Task Title	Checkmark	JIRA ID	Status Icon	Priority Icon
Documentação de casos de uso	<input checked="" type="checkbox"/>	SCRUM-20	✓	DO
Montagem de diagrama de arquitetura	<input checked="" type="checkbox"/>	SCRUM-23	✓	DO
Desenvolver o back-end da tela de pedidos	<input checked="" type="checkbox"/>	SCRUM-17	✓	Person
Desenvolver o back-end da tela de produtos	<input checked="" type="checkbox"/>	SCRUM-16	✓	Person
Iniciar gestão de time	<input checked="" type="checkbox"/>	SCRUM-5	✓	BM
Criar repositório no GitHub	<input checked="" type="checkbox"/>	SCRUM-1	✓	KO
Iniciar protótipo de telas inicial	<input checked="" type="checkbox"/>	SCRUM-4	✓	GS
Desenvolver tela de pedidos	<input checked="" type="checkbox"/>	SCRUM-25	✓	Person
Desenvolver tela de produtos	<input checked="" type="checkbox"/>	SCRUM-18	✓	Person
Documentação do design system	<input checked="" type="checkbox"/>	SCRUM-24	✓	DO
Entrevista inicial de análise de requisitos	<input checked="" type="checkbox"/>	SCRUM-3	✓	DO
Estudar desenvolvimento do backend	<input checked="" type="checkbox"/>	SCRUM-2	✓	YA
Criar diagrama de caso de uso	<input checked="" type="checkbox"/>	SCRUM-7	✓	DO
Documentação do projeto	<input checked="" type="checkbox"/>	SCRUM-6	✓	DO

Fonte: Autoria Própria (2025)

5 DIÁRIOS DE BORDO

Foi criado um diário de bordo que detalha o acompanhamento do projeto, a seguir serão apresentados os relatórios referentes aos dias 03, 11 e 17 de setembro.

O dia 03 de setembro, quadro 10, marca o início do projeto, o dimensionamento das tarefas e a criação do backlog.

Quadro 10 - Diário de Bordo de 03 de setembro

Logbook (Daily)					
Nº Backlog	Data	Anotações	Riscos	Oportunidades	Status
Sprint 1 - Task SCRUM 5	03/09/2025	Foi efetuado a criação de cards no Jira para a sprint 1 e alinhamento de objetivos iniciais com a equipe	Atraso no projeto	Controle de atividades	Concluído
Sprint 1 - Task SCRUM 6	03/09/2025	Iniciar documentação do projeto	A tarefa é incremental e será perpetuada durante praticamente todas as sprints	Processo de documentação do desenvolvimento do projeto	Em desenvolvimento
Sprint 1 - Task SCRUM 1	03/09/2025	Entrega do repositório do GitHub	-	Tornará possível o versionamento e o compartilhamento de desenvolvimento de código pela equipe	Concluído
Sprint 1 - Task SCRUM 4	03/09/2025	Iniciar prototipação da tela. Criar a tela em um modelo inicial	Apesar da tarefa iniciar no dia de hoje, pode se estender por ser o ponto central do projeto. Necessário para iniciar desenvolvimento do frontend	-	Em desenvolvimento
Sprint 1 - Task SCRUM 3	03/09/2025	Entrevista com o cliente para levantamento de requisitos	-	Mapeamento de requisitos e necessidades para o sistema.	Em desenvolvimento
Sprint 1 - Task SCRUM 2	03/09/2025	Está sendo feita uma pesquisa para a escolha de linguagens de programação utilizadas no projeto, verificando a viabilidade de acordo com os requisitos levantados	-	-	Em desenvolvimento

Sprint 1 - Task SCRUM 7	03/09/2025	Iniciar diagrama de casos de uso	-	-	Backlog
-------------------------	------------	----------------------------------	---	---	---------

Fonte: Autoria Própria (2025)

O dia 11 de setembro é onde ocorreu o início do desenvolvimento e o levantamento de requisitos, conforme quadro 11.

Quadro 11 - Diário de Bordo de 11 de setembro

Logbook (Daily)					
Nº Backlog	Data	Anotações	Riscos	Oportunidades	Status
Sprint 1 - Task SCRUM 6	11/09/2025	Inicar documentação do projeto	A tarefa é incremental e será perpetuada durante praticamente todas as sprints	Elaborado Detalhamento do Cenário e adaptando o levantamento de requisitos.	Em desenvolvimento
Sprint 1 - Task SCRUM 4	11/09/2025	Iniciar prototipação da tela. Criar a tela em um modelo inicial	-	A partir do protótipo feito, foi possível iniciar o desenvolvimento do front-end	Concluído
Sprint 1 - Task SCRUM 3	11/09/2025	Entrevista com o cliente para levantamento de requisitos	-	Após o levantamento de requisitos e detalhamento do escopo, o projeto pôde iniciar o desenvolvimento do back-end e front-end que atendessem as expectativas do cliente.	Concluído
Sprint 1 - Task SCRUM 2	11/09/2025	Está sendo feita uma pesquisa para a escolha de linguagens de programação utilizadas no projeto, verificando a viabilidade de acordo com os requisitos levantados	-	Após a pesquisa, foi elencado o uso de HTML/CSS para o front-end, JavaScript como comunicador com o back-end e Java como back-end	Concluído
Sprint 1 - Task SCRUM 7	11/09/2025	Iniciar diagrama de casos de uso	-	-	Backlog
Sprint 2 - Task SCRUM 8	11/09/2025	Desenvolvimento inicial de tela no HTML e estilização no CSS	Com base no protótipo, está sendo modelado a tela de cadastro.	-	Em desenvolvimento
Sprint 2 - Task SCRUM 9	11/09/2025	Criação de endpoints iniciais de POST referente a criar o cliente	Criar o modelo inicial da API do back-end para	-	Em desenvolvimento

			comunicação com o banco de dados		
--	--	--	----------------------------------	--	--

Fonte: Autoria Própria (2025)

O dia 17 de setembro, conforme apresentado no quadro 12, foi focado no desenvolvimento do backend e criação das interfaces para o usuário, assim como criação do JavaScript para uso de API's externas.

Quadro 12 - Diário de Bordo de 17 de setembro

Logbook (Daily)					
Nº Backlog	Data	Anotações	Riscos	Oportunidades	Status
Sprint 1 - Task SCRUM 6	17/09/2025	Iniciar documentação do projeto	A tarefa é incremental e será perpetuada durante praticamente todas as sprints	Realizado todo o levantamento de requisitos, incluindo Observação, Desenho do Processo, Entrevista e Desenho de Interface, culminando nos requisitos funcionais e não funcionais.	Em desenvolvimento
Sprint 1 - Task SCRUM 7	17/09/2025	Iniciar diagrama de casos de uso	-	Desenvolvido em Astah, conforme 3º semestre, em um processo simples com os casos de cadastro de cliente e envio de informativo.	Concluído
Sprint 2 - Task SCRUM 8	17/09/2025	Desenvolvimento inicial de tela no HTML e estilização no CSS	Com base no protótipo, está sendo modelado a tela de cadastro.	Foi concluído a estilização e desenvolvimento do front-end	Concluído
Sprint 2 - Task SCRUM 9	17/09/2025	Criação de endpoints iniciais de POST referente a criar o cliente	Criar o modelo inicial da API do back-end para comunicação com o banco de dados	A tarefa foi dividida em duas tarefas subsequentes: SCRUM-10 e SCRUM-11	Em desenvolvimento
Sprint 3 - Task SCRUM 10	17/09/2025	Refatoração do back-end - Etapa 01	Alteração da application-properties, entidade e repository	Foi criado banco de dados estabelecido em memória, entidade de cliente e de endereço, repositório para	Concluído

				ambas as entidades	
Sprint 2 - Task SCRUM 11	17/09/2025	Refatoração do back-end - Etapa 02	Alteração de service, dtos e controller		Em desenvolvimento
Sprint 3 - Task SCRUM 12	17/09/2025	Criar tela de listagem de clientes e deletar	-	-	Em desenvolvimento

Fonte: Autoria Própria (2025)

O dia 24 de setembro foi marcado pela finalização da Sprint e das atividades realizadas até aquele ponto, conforme quadro 13.

Quadro 13 - Diário de Bordo de 24 de setembro

Logbook (Daily)					
Nº Backlog	Data	Anotações	Riscos	Oportunidades	Status
Sprint 1 - Task SCRUM 6	24/09/2025	Iniciar documentação do projeto	A tarefa é incremental e será perpetuada durante praticamente todas as sprints	-	Concluído
Sprint 3 - Task SCRUM 11	24/09/2025	Refatoração do back-end - Etapa 02	Alteração de service, dtos e controller		Concluído
<u>Sprint 3 - Task SCRUM 12</u>	24/09/2025	Criar tela de listagem de clientes e deletar	-	-	Em desenvolvimento
Sprint 4 - Task SCRUM 13	24/09/2025	Código javascript da tela de listagem de clientes e de deletar			Em desenvolvimento
<u>Sprint 4 - Task SCRUM 14</u>	24/09/2025	Integrar front-end completo com o back-end da tela principal de cadastro de clientes	-	-	Concluído
Sprint 4 - Task SCRUM 15	24/09/2025	Integrar front-end completo com o back-end da tela de listagem de clientes e de deletar/alterar			Em desenvolvimento
<u>Sprint 4 - Task SCRUM 16</u>	24/09/2025	Realizar testes de CRUD nas três telas	Para realização de todos os testes, é necessário que as telas envolvidas no projeto sejam concluídas	-	Em desenvolvimento

Fonte: Autoria Própria (2025)

O quadro 14 detalha a abertura da Sprint 2 para criação das telas de cadastro de produto e pedido.

Quadro 14 - Diário de Bordo de 29 de outubro

Logbook (Daily)					
Nº Backlog	Data	Anotações	Riscos	Oportunidades	Status
SCRUM 15	29/10/2025	Resolver problema de tamanho do banco de dados dos campos de cliente e endereço	Os campos (como e-mail, logradouro, razão social) estavam com tamanho menor do que o aceitável		Concluído
SCRUM 16	29/10/2025	Desenvolver o back-end da tela de produtos	A tarefa deve perdurar por mais de uma sprint		Em andamento
SCRUM 17	29/10/2025	Desenvolver o back-end da tela de pedidos	A tarefa deve perdurar por mais de uma sprint		Em andamento
SCRUM 18	29/10/2025	Desenvolver tela de produtos	A tarefa deve perdurar por mais de uma sprint		Em andamento
SCRUM 20	29/10/2025	Documentação de casos de uso	Mudanças posteriores no funcionamento base do sistema		Em andamento
SCRUM 23	29/10/2025	Montagem de diagrama de arquitetura			Não Iniciado
SCRUM 24	29/10/2025	Documentação do design system	-		Não Iniciado

Fonte: Autoria Própria (2025)

O quadro 15 descreve as tarefas realizadas no dia 05 de novembro, durante a realização desse diário o sistema já estava no final do desenvolvimento.

Quadro 15 - Diário de bordo de 05 de novembro

Logbook (Daily)					
Nº Backlog	Data	Anotações	Riscos	Oportunidades	Status
SCRUM 16	05/11/2025	Desenvolver o back-end da tela de produtos	-	-	Concluído
SCRUM 17	05/11/2025	Desenvolver o back-end da tela de pedidos	-	-	Em Andamento
SCRUM 18	05/11/2025	Desenvolver tela de produtos	-	Com base na conclusão da tela de produtos (html, css e javascript) foi possível iniciar o desenvolvimento da tela de pedidos.	Concluído
SCRUM 20	05/11/2025	Documentação de casos de uso	-	Documentação agora engloba as atualizações na arquitetura e funcionalidades novas do sistema.	Concluído
SCRUM 23	05/11/2025	Montagem de diagrama de arquitetura	-	-	Em Andamento
SCRUM 24	05/11/2025	Documentação do design system	-	-	Concluído

Fonte: Autoria Própria (2025)

O último diário de bordo está apresentado no quadro 16, descrevendo como foi a finalização das tarefas e do sistema como um todo.

Quadro 16 - Diário de bordo de 12 de novembro

Logbook (Daily)					
Nº Backlog	Data	Anotações	Riscos	Oportunidades	Status
SCRUM 23	12/11/2025	Montagem de diagrama de arquitetura	-	-	Concluído

SCRUM 17	12/11/2025	Desenvolver o back-end da tela de pedidos	-	-	Concluído
SCRUM 18	12/11/2025	Desenvolver a tela de pedidos	-	-	Concluído
SCRUM 26	12/11/2025	Adicionar no github a documentação completa do projeto	-	-	Concluído

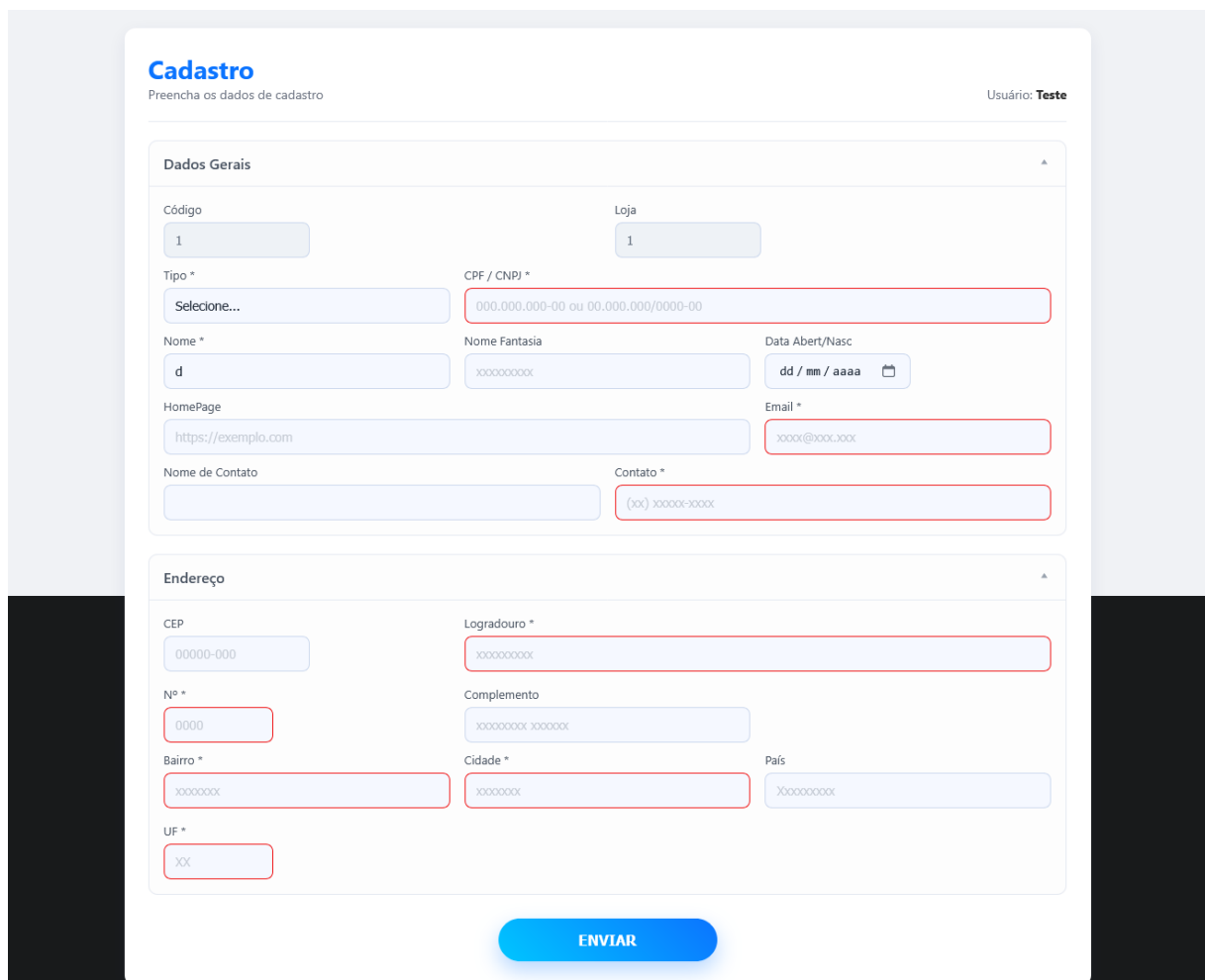
Fonte: Autoria Própria (2025)

6 DISPONIBILIZAÇÃO DO MVP

O código-fonte do projeto final está disponível e público no GitHub, através do link: https://github.com/Kerolinferreira/projeto_lab_engenharia_software.git. A partir do link, é possível a clonagem do repositório para uso e teste.

Para fins de documentação, é apresentada, na figura 6, o protótipo da tela de cadastro de cliente desenvolvido durante a primeira Sprint.

Figura 7 - Protótipo de tela de cadastro de cliente



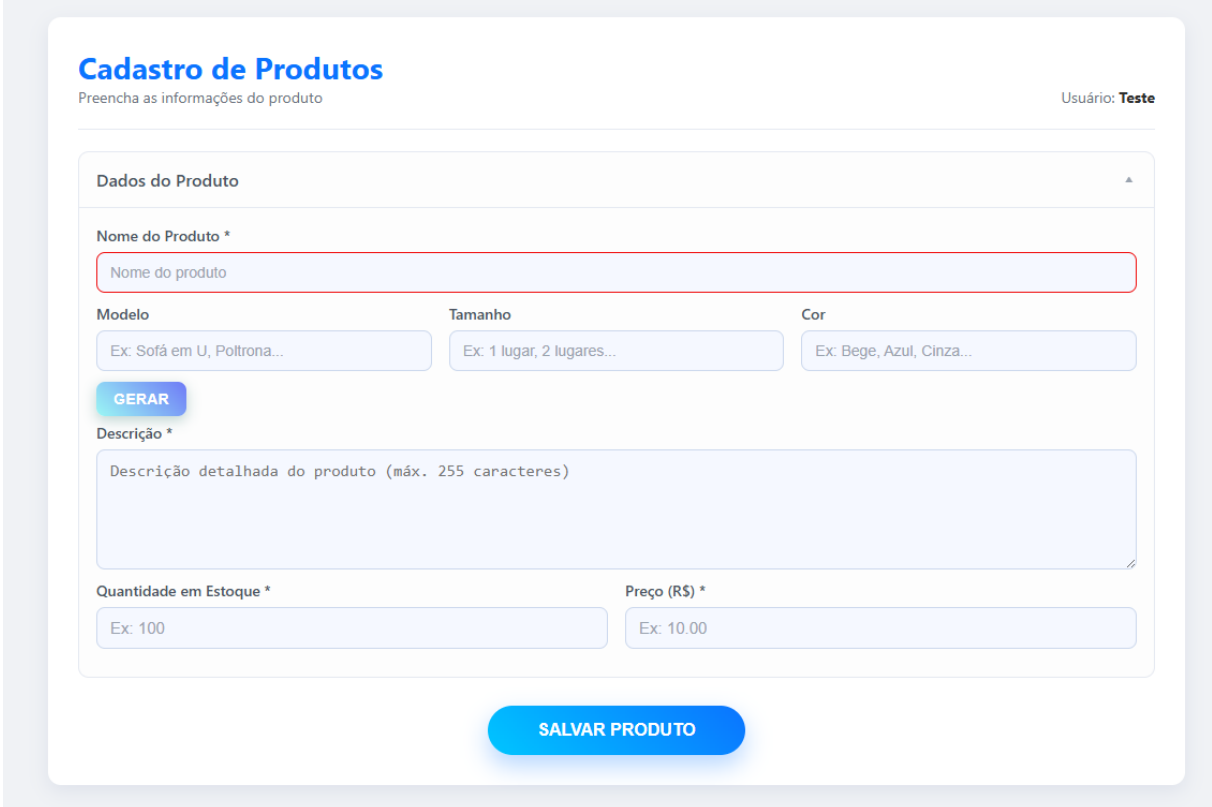
O protótipo da tela de cadastro de cliente, intitulada "Cadastro", apresenta o seguinte layout:

- Cabeçalho:** Título "Cadastro" e subtítulo "Preencha os dados de cadastro". No canto superior direito, há o texto "Usuário: **Teste**".
- Seção "Dados Gerais":**
 - Código:** Campo com o valor "1".
 - Loja:** Campo com o valor "1".
 - Tipo *:** Menu suspenso com a opção "Selecione...".
 - CPF / CNPJ *:** Campo com o formato "000.000.000-00 ou 00.000.000/0000-00".
 - Nome *:** Campo com o valor "d".
 - Nome Fantasia:** Campo com o valor "xxxxxxxxx".
 - Data Abert/Nasc:** Campo com o formato "dd / mm / aaaa" e ícone de calendário.
 - HomePage:** Campo com o valor "https://exemplo.com".
 - Email *:** Campo com o valor "xxxx@xxx.xxx".
 - Nome de Contato:** Campo em branco.
 - Contato *:** Campo com o formato "(xx) xxxxx-xxxx".
- Seção "Endereço":**
 - CEP:** Campo com o valor "00000-000".
 - Logradouro *:** Campo com o valor "xxxxxxxxx".
 - Nº *:** Campo com o valor "0000".
 - Complemento:** Campo com o valor "xxxxxxxxx xxxxxx".
 - Bairro *:** Campo com o valor "xxxxxxxxx".
 - Cidade *:** Campo com o valor "xxxxxxxxx".
 - País:** Campo com o valor "xxxxxxxxx".
 - UF *:** Campo com o valor "XX".
- Botão:** Um botão azul com o texto "ENVIAR" localizado na base da tela.

Fonte: Autoria Própria (2025)

Durante a segunda Sprint ainda foram criadas as telas de cadastro de produto e cadastro de pedido, para fins de documentação é apresentado o protótipo primeira é apresentada abaixo na forma de forma da figura 08.

Figura 8 - Protótipo de tela de cadastro de produto



O protótipo da tela de cadastro de produto apresenta o seguinte layout:

- Cadastro de Produtos**: Título principal em azul.
- Preencha as informações do produto**: Subtítulo em cinza.
- Usuário: Teste**: Texto no canto superior direito.
- Dados do Produto**: Cabeçalho da seção principal.
- Nome do Produto ***: Campo de texto com placeholder "Nome do produto".
- Modelo**: Campo de texto com placeholder "Ex: Sofá em U, Poltrona...".
- Tamanho**: Campo de texto com placeholder "Ex: 1 lugar, 2 lugares...".
- Cor**: Campo de texto com placeholder "Ex: Bege, Azul, Cinza...".
- GERAR**: Botão de ação em azul.
- Descrição ***: Campo de texto com placeholder "Descrição detalhada do produto (máx. 255 caracteres)".
- Quantidade em Estoque ***: Campo de texto com placeholder "Ex: 100".
- Preço (R\$) ***: Campo de texto com placeholder "Ex: 10.00".
- SALVAR PRODUTO**: Botão de ação em azul.

Fonte: Autoria Própria (2025)

Ainda para fins de documentação, o protótipo de tela de cadastro de pedido é apresentado na figura 9.

Figura 9 - Protótipo de tela de cadastro de pedido

Cadastro de Pedidos

Preencha os dados do pedido

Usuário: **Teste**

Dados do Pedido

Nº Pedido

Auto/Sequencial

Data *

dd / mm / aaaa

Status *

Selecione...

Cliente

Buscar cliente *

Digite nome, CNPJ, cidade...

Buscar

ID

—

Dica: integre esta busca com a sua API (GET /clientes?query=...)

Razão Social

Fantasia

CNPJ/CPF

Cidade

Itens

+ Adicionar item

#

Descrição

Frete (R\$)

0

Total (R\$)

0,00

Pagamento & Entrega

Fonte: Autoria Própria (2025)

7 ASPECTOS DO SISTEMA

7.1 ARQUITETURA UTILIZADA

O sistema pode se enquadrar em múltiplas arquiteturas, para fins dessa documentação a arquitetura utilizada é MVC (Modelo-Visão-Controlador). A arquitetura consiste em dividir o sistema em três camadas: Modelo, Visão e Controlador.

- Modelo: componente/camada que lida com os dados (usando um banco de dados quando necessário), lógicas e regras do sistema, onde são montados os objetos;
- Visão: representação gráfica dos dados e como será sua exibição aos usuários;
- Controlador: o receptor dos dados que faz uso do modelo para o processamento e da visão para a exibição.

O sistema não é muito diferente da Arquitetura em Camadas, ele separa os componentes para a montagem e desenvolvimento do sistema, mas colocando o controlador como o intermediário entre ambas as partes.

Outras duas arquiteturas podem ser usadas para descrever o sistema: Arquitetura Cliente-Servidor e Arquitetura em Camadas.

Cliente-Servidor é uma arquitetura onde o Cliente solicita serviços ou dados ao servidor, nesse modelo o servidor é responsável pela aplicação ou programa que processa os dados. A ideia desse sistema é permitir que múltiplos clientes usem um mesmo servidor, acessando remotamente ou até com cabeamento físico.

A Arquitetura em Camadas é uma arquitetura de desenvolvimento que separa a aplicação em camadas, normalmente uma camada para o visual, outra lógica de negócios e outra para o banco de dados. Ela busca delimitar responsabilidades específicas para cada nível e simplificar o desenvolvimento, limitando também como ocorre a comunicação. Como falado, o falado a arquitetura MVC também usa de camadas, mas são camadas específicas e com um escopo específico de suas atividades.

7.2 TECNOLOGIAS E INOVAÇÕES

Durante a realização do projeto algumas tecnologias e inovações foram utilizadas para otimizar o sistema em si e permitir a melhor realização das rotinas de desenvolvimento.

Para a melhor realização do processo de cadastro duas inovações foram chaves: a API Brasil API (1.0.0)¹ e a API CNPJs². Ambas as APIs buscaram otimizar o processo de cadastro

¹ <https://brasilapi.com.br/docs#tag/CEP>

² <https://docs.cnpj.ws/referencia-de-api/api-publica/consultando-cnpj>

e permitir menos consultas a links externos. A primeira busca otimizar o preenchimento dos endereços dos novos clientes através do preenchimento do CEP, a Brasil API busca otimizar o serviço dos correios através da melhor criação de *endpoints* para as requisições. A segunda otimiza o resgate dos mais variados dados da empresa que são relacionados ao CNPJ. Ambas as tecnologias usam do ‘JSON’ para otimizar o resgate das informações.

Buscando satisfazer a demanda do diretor da Maicosoft foi através do uso da API JavaMailSender³ que é integrada no Java Spring Boot. A API permite o envio de automático de e-mail e servirá para encaminhar os clientes cadastrados para o diretor da Maicosoft.

Para auxiliar no desenvolvimento da API em Java com Spring Boot foram utilizadas diversas tecnologias auxiliares, como o Maven para construção do pacote, JPA para auxiliar na criação do banco de dados e “tradução” dos processos para auxiliar na portabilidade e, para fins de apresentação desse projeto e demonstração, o banco H2 que é de fácil portabilidade e permite que ele seja iniciado em todos os computadores sem necessidade de instalações adicionais. Outra extensão importante para agilizar o processo foi o Lombok que reduz a quantidade de escrita de código por usar anotações para realizar os ‘Getters’, ‘Setters’ e ‘Constructor’. O controle de versionamento e compartilhamento do código-fonte foi feito através do GitHub com um *Workflow* próprio para esse projeto.

A prototipagem do processo foi realizada com o Figma⁴, plataforma de design de interfaces e protótipos, através dele é possível que múltiplos membros acompanhem o desenvolvimento da interface e opinem sobre ela.

O backlog foi desenvolvido com a plataforma Jira da Atlassian, desenvolvida para metodologias ágeis como Kanban, que organiza as tarefas em cartões, e Scrum, metodologia usada no projeto, que organiza as tarefas em variadas histórias. A documentação ainda foi montada com o Astah Community, versão gratuita do software Astah da ChangeVision, que auxilia no desenho de diagramas UML.

Por derradeiro, a última tecnologia implementada foi API do Gemini⁵ que foi utilizada para aprimorar e otimizar na criação de descrição de produtos. Esse micro serviço foi adotado devido a sua facilidade de aplicação e tendência de uso nos sistemas atuais de marketplace, aplicativos de delivery e similares.

³<https://docs.spring.io/spring-framework/docs/current/javadoc-api/org/springframework/mail/javamail/JavaMailSender.html>

⁴ <https://www.figma.com/design/NIX15G39DSJcVyNiNKJqv7/Projeto-Engenharia?node-id=0-1&p=f>

⁵ <https://ai.google.dev/gemini-api/docs/models?hl=pt-br>

7.3 DIAGRAMA DE ARQUITETURA DO SISTEMA

Foi desenvolvida para a melhor percepção das tecnologias utilizadas um diagrama de arquitetura que descreve onde cada tecnologia se encaixa nas arquiteturas utilizadas para o desenvolvimento do sistema. O diagrama em questão é apresentado no quadro 17 abaixo, e relaciona as tecnologias apresentadas no capítulo de tecnologias e inovações.

Quadro 17 – Descrição de arquitetura

Cliente	View	HTML5
		CSS
		JavaScript
Servidor	View	Lombok H2
	Control	
	Model	
Micros serviços	API's	API 1 - Brasil API
		API 2 - API CNPJws
		API 3 – JavaMailSender
		API 4 – Gemini API

Fonte: Autoria Própria (2025)

7.3.1 Design System

As telas foram desenvolvidas e estilizadas com base em um design system próprio focado em uma estética agradável e intuitiva para o usuário. O design system desenvolvido usa as diretrizes descritas abaixo para sua organização.

As cores do sistema se limitam ao:

- Branco para cor de fundo do formulário, *foreground*, e para o texto dos botões;
- Tons de cinza para os textos de subtítulos, nomes de secções e nomes de campo;
- Vermelho para apresentar avisos e campos obrigatórios;
- Cor única para representar a “temática da aplicação”, essa cor única será utilizada no *background* em tons claros e pastéis que deverão ser reproduzidos como cor de fundo nos campos de edição, ela ainda será usada na cor base para o título de primeiro nível de cada página, como um gradiente do branco ao tom do título para apresentar os botões e desenhando as secções do formulário.

Dessa forma, a cor temática será usada para gerar contraste na aplicação, destacando os botões e “desenhando” o formulário. A percepção do usuário é que a cor temática é a cor do sistema, mas sem prejudicar o equilíbrio das informações ou chamar atenção em excesso.

A tela usa de uma leitura no sentido ocidental, da esquerda para a direita e de cima para baixo, assim é importante que o título de primeiro nível fique no canto superior esquerdo. E,

buscando evitar possíveis confusões no uso do sistema, o título de primeiro nível deve vir acompanhado de um parágrafo que orienta a função da tela.

As secções do sistema serão colapsáveis para possibilitar a visualização de informações relevantes, mas sem precisar mudar as telas ou recarregar a aplicação. O contorno das secções será realizado na cor temática do sistema, delimitando as partes colapsáveis sem deixar a imagem sobrecarregada.

É preferível que os campos a serem preenchidos contenham máscaras para orientar sobre a forma de preenchimento e reduzir dúvidas sobre o manuseio do sistema. Durante a montagem do sistema para telas maiores, modelo *desktop*, os campos de preenchimento devem ser limitados a três para facilitar o preenchimento rápido e acompanhamento visual do usuário. Campos precisem de destaque para informações mais importantes devem ser maiores e usar da cor temática em tom mais escuro, sem chamar mais atenção que os botões e os títulos da página.

Ainda sobre os campos, é importante que as máscaras sejam uniformes e os valores sejam devidamente pontuados sem apresentar os símbolos monetários. Essas restrições buscam garantir o entendimento das informações que devem ser inseridas e reduzir erros na hora do preenchimento.

Os botões destacados da aplicação devem se limitar ao botão de finalização/salvamento, permitindo a percepção clara do usuário de quais ações finalização/alteram as informações. Demais botões devem ter a mesma cor e padrões dos campos.

O sistema deve se limitar ao uso de retângulos arredondados (campos, secções, botões e demais elementos), a única exceção é botão de finalização/salvamento que deve ter forma de pílula; também para educar o usuário da ação que está por fazer.

7.4 TESTES DE QUALIDADE

A seguir é apresentado o quadro com os testes realizados no sistema para assegurar o funcionamento das funções básicas. A seguir

Quadro 18 - Teste de qualidade aplicados no sistema

ID do Caso de Teste	Descrição do Caso de Teste	Caso de Teste	Dados do Teste	Resultado esperado	Passou / Falhou
T01	Verifique o funcionamento dos dados obrigatórios.	Uma vez que o usuário está na tela de cadastro, ele deve tentar enviar as informações com	-	Os dados não devem ser enviados.	Passou

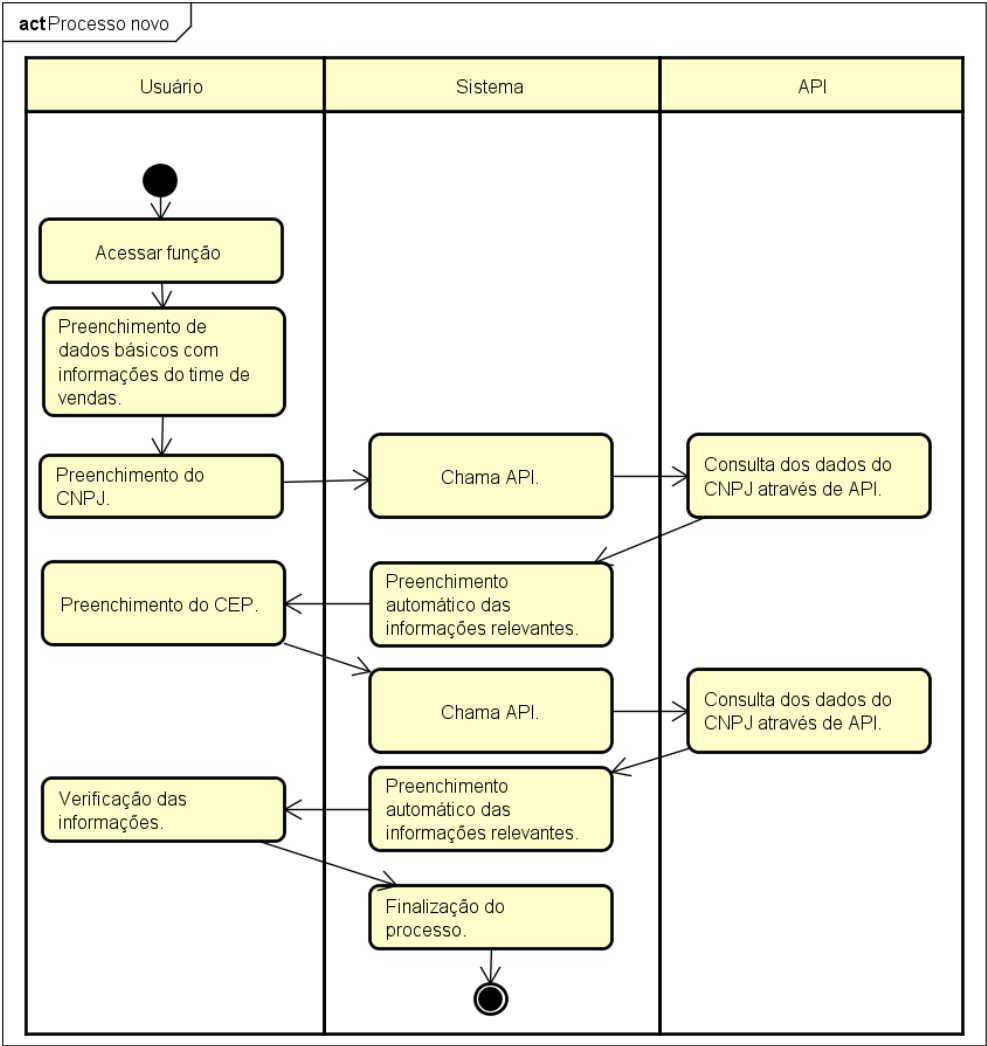
		todos os campos nulos.			
T02	Verifique envio dos dados através de JSon.	Uma vez preenchidos os dados da Fatec Itapetininga, é preciso verificar no backend a estrutura resultante.	Dados gerais da Fatec Itapetininga consultados através do CNPJws.	Os dados devem aparecer em uma estrutura JSON.	Passou
T03	Verifique o funcionamento da API de CNPJ.	Uma vez preenchido o campo de CNPJ, é preciso enviar a requisição para a API.	CNPJ Fatec Itapetininga: 62.823.257/0131-89	Os demais campos relacionados ao CNPJ devem ser preenchidos automaticamente.	Passou
T04	Verifique o funcionamento da API de CEP.	Uma vez preenchido o campo de CEP, é preciso enviar a requisição para a API.	CEP Fatec Itapetininga: 18205-600	Os demais campos relacionados ao CEP devem ser preenchidos automaticamente.	Passou
T05	Verifique a leitura dos dados do GET.	Uma vez cadastrados os clientes, é preciso verificar se eles são extraídos na tela de consulta.	-	Uma lista de clientes pré-cadastrada deve ser apresentada.	Passou
T06	Verifique a edição dos dados de um cliente já cadastrado.	Uma vez cadastrado o cliente Fatec Itapetininga, é para verificar se é possível alterar para o CNPJ de outra Fatec.	CNPJ Fatec Sorocaba: 62.823.257/0002-81	O novo CNPJ junto dos demais campos devem ser alterados e salvos dessa forma.	Passou
T07	Verifique o envio dos dados para o e-mail pré-cadastrado.	Uma vez preenchidos os dados, é preciso verificar se eles estão sendo enviados e em qual formato.	Email: diogo.oliveira27@fatec.sp.gov.br	A caixa de e-mail apresentada deve receber os dados do cliente.	Passou
T08	Verifique a deleção de um cliente.	Uma vez cadastrado o cliente Fatec Itapetininga, é preciso verificar se é possível a sua deleção.	-	O cliente Fatec Itapetininga deve ser excluído.	Passou

Fonte: Autoria Própria (2025)

7.5 FLUXO DA APLICAÇÃO

A seguir, é apresentado o novo fluxo da aplicação onde o sistema não apenas receber e envia os dados, mas consegue ativamente consultar as informações das API's para facilitar a realização do processo.

Figura 10 - Diagrama de Atividade do novo fluxo de processo



powered by Astah