

**CENTRO PAULA SOUZA
FACULDADE DE TECNOLOGIA DE FRANCA
“Dr. THOMAZ NOVELINO”**

TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

RAFAEL ALBANO DA SILVA

LICITASP:

Sistema para Extração de Licitações do Diário Oficial do Estado de São Paulo

Trabalho de Graduação apresentado à Faculdade de Tecnologia de Franca - “Dr. Thomaz Novelino”, como parte dos requisitos obrigatórios para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador: Prof. Me. Carlos Eduardo de França
Roland

FRANCA/SP

2025

LICITASP:

Sistema para Extração de Licitações do Diário Oficial do Estado de São Paulo

Rafael Albano da Silva¹

Carlos Eduardo de França Roland²

Resumo

Este trabalho apresenta o Licitasp, um protótipo de sistema *web* para automação da extração, filtragem e notificação sobre licitações publicadas no Diário Oficial do Estado de São Paulo. O problema abordado foi a dificuldade de pequenas e médias empresas em acompanhar diariamente editais e licitações oficiais, o que reduz sua participação em processos de contratação pública. O objetivo foi projetar e dimensionar uma solução que raspagem de dados automática das edições do Diário Oficial, e notificar as empresas usuárias do sistema. O projeto aplicou métodos e boas práticas propostas pela Engenharia de Software no Ciclo de Vida do Desenvolvimento de Software, como elicitação de requisitos, modelagem de processos, diagramas de projeto e prototipagem da interface. Espera-se reduzir o tempo de identificação de oportunidades, ampliar a participação das PMEs em licitações e demonstrar a aplicabilidade técnica e operacional da solução.

Palavras-chave: contratos públicos; diário oficial; licitações; pequenas e médias empresas; raspagem de dados.

Abstract

This work presents Licitasp, a prototype web system for automating the extraction, filtering, and notification of tenders published in the Official Gazette of the State of São Paulo. The problem addressed was the difficulty faced by small and medium-sized enterprises in monitoring official tenders and bidding processes daily, which reduces their participation in public procurement processes. The objective was to design and scale a solution that automatically scrapes data from the editions of the Official Gazette and notifies the companies using the system. The project applied methods and best practices proposed by Software Engineering in the Software Development Life Cycle, such as requirements elicitation, process modeling, design diagrams, and interface prototyping. It is expected to reduce the time for identifying opportunities, increase the participation of SMEs in tenders, and demonstrate the technical and operational applicability of the solution.

Keywords: bidding processes; data scraping; official gazette; public contracts; small and medium-sized enterprises.

¹ Graduando em Análise e Desenvolvimento de Sistemas pela Fatec Dr Thomaz Novelino – Franca/SP.
Endereço eletrônico: albano_rafael15@hotmail.com

² Docente em Análise e Desenvolvimento de Sistemas pela Fatec Dr Thomaz Novelino – Franca/SP.
Endereço eletrônico: carlos.roland@fatec.sp.gov.br.

1 Introdução

No cenário de contratações públicas, Pequenas e Médias Empresas (PME) enfrentam um desafio diário: a perda de oportunidades devido à dificuldade de monitorar manualmente as publicações do Diário Oficial do Estado de São Paulo (DOE-SP). A partir da atuação como colaborador de empresa de prestação de serviços na área da saúde, foi identificada, a partir de relatos de clientes, a dificuldade de encontrar licitações no diário oficial que causou a perda de oportunidade de participar de licitações ou começar projetos estratégicos. Esse problema também existe quando se considera a diversidade de editais, portarias e agendas públicas disponibilizadas diariamente, muitas vezes em formatos pouco amigáveis.

Diante desse contexto, o problema central do projeto foi: como monitorar eficientemente as licitações publicadas no DOE-SP e notificar pequenos e microempresários as oportunidades? A ausência de uma solução automatizada para esse processo gera desigualdade competitiva, já que empresas maiores contam com equipes dedicadas, enquanto PME dependem de esforços manuais e sujeitos a falhas.

A solução proposta do LictaSP, foi a prototipagem e o dimensionamento de um sistema *web* que automatiza a captura, a busca e a notificação de dados do DOE-SP. A ferramenta deveria utilizar técnicas de raspagem de dados para extrair informações diárias, processá-las com base em palavras-chave definidas pelos usuários e enviar alertas via canais verificados como e-mail ou SMS. Para implementar a arquitetura do sistema foram adotadas tecnologias consolidadas: Python com Selenium para *scraping*, PostgreSQL para armazenamento de dados e Bootstrap para a interface responsiva.

O projeto alinhou competências desenvolvidas no curso de Análise e Desenvolvimento de Sistemas (ADS), como prototipagem de telas, documentação técnica, e verificação da viabilidade de otimização e automação de processos. Socialmente, o LictaSP democratiza o acesso a informações públicas, reduzindo desigualdades competitivas no mercado e fortalecendo a participação de PME em licitações estaduais. Isso não apenas amplia o leque de fornecedores para a administração pública, mas também fomenta a economia local.

A viabilidade técnica é se dá pelo uso de tecnologias de código aberto e infraestrutura de baixo custo. O investimento inicial é baixo, focando em desenvolvimento e manutenção simplificada, com potencial de escalar o produto

futuramente. O alinhamento com metodologias ágeis e ferramentas consolidadas e *open-source*, como Git para versionamento e Docker para conteinerização, garante a sustentabilidade do projeto.

O LictaSP não só visa resolver um problema operacional identificado no mercado, mas também exemplifica a aplicação prática de conhecimentos de análise e desenvolvimento de sistemas em projetos social e economicamente importantes. Ao dimensionar uma solução acessível e eficiente, o projeto contribui para a transparência pública e a competitividade empresarial, cumprindo um duplo papel: técnico e social.

1.1 Termo da Abertura do Projeto (TAP)

O TAP é um documento essencial na fase inicial de qualquer projeto, responsável por formalizar sua existência, definir seus objetivos gerais e conceder autoridade ao gerente de projeto para mobilizar recursos organizacionais. Segundo Varolo e Fiirst (2020), o TAP constitui uma ferramenta estratégica que possibilita a estruturação de propostas e planos de trabalho com base em informações claras e verificáveis, assegurando que todos os envolvidos compreendam a finalidade e as diretrizes do projeto desde o início.

Ele contém informações essenciais como objetivos, justificativa, escopo preliminar, partes interessadas, riscos iniciais e recursos previstos. Ainda conforme Varolo e Fiirst (2020), sua aplicação prática em instituições públicas, como observado no município de Guaíra-PR, demonstrou que o TAP funciona não apenas como instrumento de autorização, mas como guia de alinhamento técnico, administrativo e estratégico.

O detalhamento do TAP do projeto LictaSP está apresentado no Apêndice 1.

2 Viabilidade do Projeto

Para avaliar a viabilidade do projeto foi criado o *Business Model Canvas* (BMC), que é uma ferramenta visual que ajuda a compreender como uma empresa cria e entrega valor. Ele organiza os principais elementos de um negócio em nove blocos que mostram clientes, proposta de valor, canais, recursos e custos. Seu uso facilita o planejamento e a comunicação entre empreendedores e equipes.

No contexto brasileiro, o BMC tem sido usado para apoiar decisões estratégicas e promover uma visão mais clara dos processos empresariais. Segundo Ismail e

Cintra (2021), a ferramenta permite alinhar a criação de valor e a estrutura operacional de forma simples e eficiente. Os autores destacam que o uso do BMC contribui para o planejamento e a inovação dentro das pequenas empresas, servindo como apoio na formulação de estratégias. A Figura 1 mostra o BMC do LictaSP.

2.2 Matriz SWOT

Segundo Speth (2023), a análise SWOT, também conhecida como matriz SWOT, é uma ferramenta essencial de planejamento estratégico que permite às organizações avaliar de forma estruturada os fatores internos e externos que influenciam seu desempenho. Essa metodologia é composta por quatro dimensões: Forças (*Strengths*), Fraquezas (*Weaknesses*), Oportunidades (*Opportunities*) e Ameaças (*Threats*).

A ferramenta foi desenvolvida originalmente na década de 1960 por Albert Humphrey e, tornou-se amplamente utilizada em diferentes contextos empresariais para apoiar o diagnóstico estratégico e a formulação de planos de ação mais eficazes (Speth, 2023).

Ao aplicar a matriz SWOT, as empresas podem definir estratégias mais coerentes, priorizar investimentos, otimizar a alocação de recursos e antecipar contingências no ambiente de negócios, o que fortalece sua capacidade de adaptação e sustentabilidade no mercado.

No Quadro 1 estão listadas as características estabelecidas para o LictaSP.

2.3 Plano de Ação 5W2H do Projeto

A sigla deriva das iniciais, em inglês, de sete questões fundamentais: *What* (o que será feito?), *Why* (por que será feito?), *Where* (onde será feito?), *When* (quando será feito?), *Who* (por quem será feito?), *How* (como será feito?) e *How much* (quanto custará?).

O método 5W2H é uma ferramenta que utiliza uma série de perguntas direcionadas ao processo produtivo, ajudando a destacar as rotinas mais relevantes, identificar problemas e propor soluções (Lisbôa e Godoy, 2012).

Neste estudo o método foi aplicado visando proporcionar soluções para as fraquezas do negócio, de modo que para cada fraqueza foram propostas soluções práticas, identificando os atores responsáveis, os custos envolvidos e as datas de conclusão (Quadro 2).

Figura 1 – Canvas



Fonte: os autores

Quadro 1 – Matriz SWOT

Forças	Oportunidades
Automação eficiente com scraping agendado	Expansão para outras fontes oficiais
Interface administrativa simples e funcional	Integração com sistemas de gestão
Uso de tecnologias consolidadas (Python, MySQL)	Melhoria contínua com feedback dos usuários
Fraquezas	Ameaças
Possibilidade de alterar a fonte oficial	Bloqueios ou alterações no layout do Diário
Um mercado mais restrito de usuários (somente empresas)	Concorrência com soluções mais robustas
Por ser um sistema WEB, pode ter problemas de atingir um público maior	Riscos de mudanças legislativas que afetem os dados

Fonte: os autores

Quadro 2 – 5W2H

Fraqueza	O que (What)	Por que (Why)	Quem (Who)	Onde (Where)	Quando (When)	Como (How)	Quanto (How Much)
Possibilidade de alterar a fonte oficial	Implementar verificação de layout na raspagem de dados	Prevenir erros caso o Diário mude o formato	Equipe de desenvolvimento	No código de scraping	Fase de manutenção	Monitorando mudanças no HTML e adaptando-se automaticamente	Requer tempo adicional de desenvolvimento
Possibilidade de alterar a fonte oficial	Supor tar múltiplas fontes de dados	Evitar dependência de uma única fonte	Equipe de desenvolvimento	Back-end e banco de dados	Ao planejar o sistema (fazer métodos reutilizáveis)	Fazer arquitetura modular e flexível	Complexidade maior, mas amplia cobertura
Um mercado mais restrito de usuários (somente empresas)	Expandir público-alvo além de empresas	Aumentar a base de clientes	marketing	Campanhas e site	Após validação inicial	Criar estratégias de divulgação e parcerias	Custos de marketing e publicidade
Um mercado mais restrito de usuários (somente empresas)	Oferecer planos para diferentes tamanhos de clientes	Adequar o sistema a vários perfis de uso	Equipe de desenvolvimento	No modelo de negócios e pricing	Após MVP	Definir faixas de preços e recursos escaláveis	Investimento em estudo de mercado e suporte
Por ser um sistema WEB, pode ter problemas de atingir um público maior	Desenvolver app simples usando WebView	Atingir usuários mobile e offline	Equipe de desenvolvimento	Android	Depois do lançamento web	Empacotar a aplicação web em containers móveis	Custo de publicação e manutenção nas lojas
Por ser um sistema WEB, pode ter problemas de atingir um público maior	Otimizar SEO e responsividade	Ampliar alcance e melhorar experiência em dispositivos móveis	Equipe de desenvolvimento	No site principal	Contínuo, desde a implementação até melhorias	Aplicar boas práticas de SEO e design responsável	Horas de desenvolvimento e consultoria de SEO

Fonte: os autores

3 Elicitação e Especificação dos Requisitos

Para compreender com profundidade as necessidades e expectativas do projeto, foi realizado um processo de elicitação de requisitos junto ao cliente. A primeira etapa consistiu em reuniões presenciais e entrevistas estruturadas, nas quais

buscou-se esclarecer as principais funcionalidades desejadas e as limitações atuais que o sistema pretende resolver. O cliente demonstrou interesse em um sistema intuitivo e eficiente, que pudesse se integrar os processos de trabalho já existentes e melhorar o fluxo de atividades.

Durante as sessões, foram aplicadas técnicas de *brainstorming* e mapeamento de processos para identificar os pontos críticos e oportunidades de melhoria. O cliente destacou a importância de certos requisitos funcionais, como a necessidade de notificar o usuário com um relatório diário das licitações encontradas, além de requisitos não funcionais, como a segurança e integridade dos dados. Também foi utilizada uma análise de casos de uso para entender as interações específicas que o sistema deve ter.

A partir dessas reuniões, foi possível elaborar uma especificação detalhada dos requisitos, dividida em requisitos funcionais e não funcionais, documentando cada um de forma clara e precisa. Esse registro formal permitiu o alinhamento das expectativas do cliente com a equipe de desenvolvimento e serviu como referência para as fases de validação do projeto e implementação do sistema.

3.1 Business Process Model and Notation (BPMN)

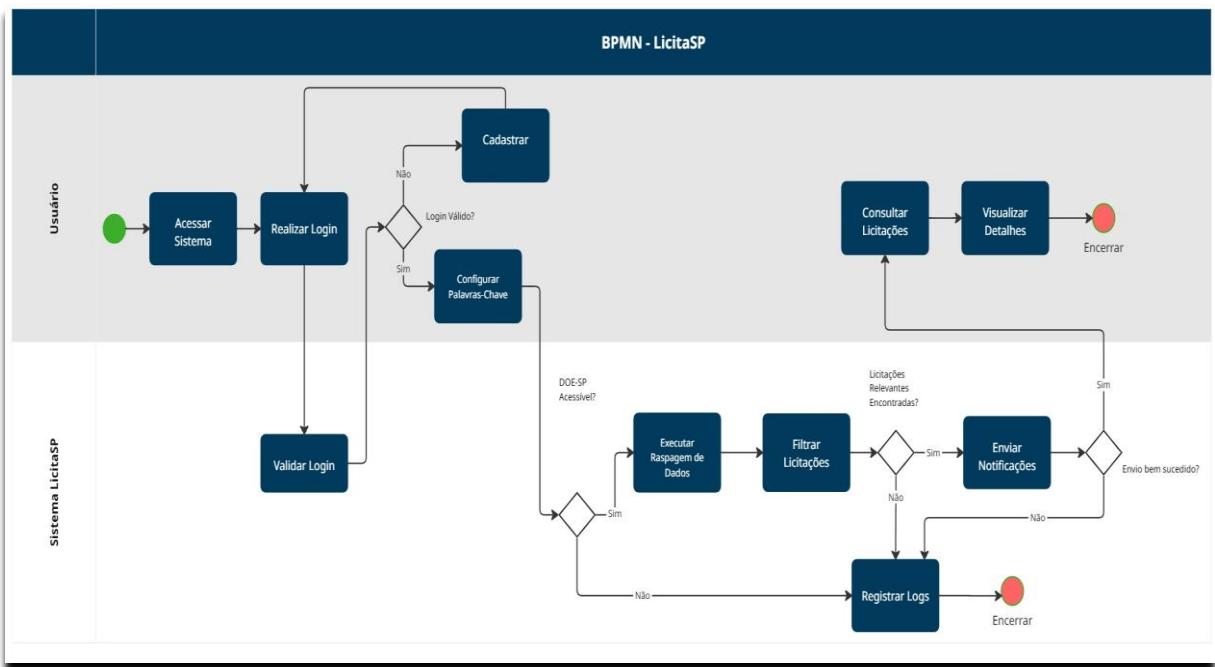
O termo BPMN significa Modelo e Notação de Processos de Negócios e corresponde a uma notação gráfica padronizada que permite representar visualmente as etapas de um processo empresarial. Segundo Molina e Andrade (2024), o BPMN foi desenvolvido para oferecer uma forma estruturada e comprehensível de modelar processos, promovendo uma visão unificada entre a área de negócio e a área técnica. Essa notação visa facilitar a comunicação entre diferentes perfis de *stakeholders* e reduzir ambiguidades no entendimento das operações organizacionais.

O principal objetivo do BPMN é fornecer uma notação inteligível e padronizada para todos os participantes do processo de negócio desde analistas de processos e desenvolvedores até gestores, favorecendo o alinhamento entre os diferentes níveis da organização. Conforme destacam Molina e Andrade (2024), o uso do BPMN possibilita mapear não apenas o fluxo ideal de execução de um processo, mas também as exceções e falhas que podem ocorrer, o que o torna uma ferramenta poderosa para o gerenciamento e melhoria contínua de processos.

A adoção do BPMN no projeto (Figura 2) tem como fundamento a necessidade de visualizar e compreender claramente as etapas dos processos de negócio,

permitindo que as expectativas de todos os envolvidos sejam alinhadas de maneira precisa. Essa abordagem possibilita uma modelagem mais eficiente, ao mesmo tempo em que fornece subsídios para a identificação de gargalos e oportunidades de otimização. Além disso, a utilização do BPMN favorece a padronização e a transparência, facilitando futuras automações e garantindo que o processo modelado esteja em conformidade com as boas práticas de engenharia de software e gestão organizacional (Molina e Andrade, 2024).

Figura 2 – BPMN



Fonte: os autores

3.2 Requisitos Funcionais

Requisitos funcionais são especificações que descrevem o que um sistema ou software deve realizar para cumprir sua função. Eles definem as funcionalidades e capacidades que o sistema precisa ter para atender às necessidades do usuário ou resolver um problema específico. Esses requisitos geralmente incluem ações que o sistema deve realizar em resposta a entradas, o comportamento esperado em diferentes cenários e as restrições que o sistema deve obedecer.

De acordo com Figueiredo (2011), os requisitos funcionais têm como objetivo descrever de forma clara as funcionalidades e serviços que o sistema deve oferecer, especificando como ele deve reagir a determinadas entradas e situações, além de definir restrições de comportamento. Esses requisitos são fundamentais para garantir

que o software atenda corretamente às necessidades e expectativas do cliente, assegurando uma compreensão precisa do que o sistema deve realizar (Quadro 3).

Quadro 3 – Requisitos Funcionais do sistema

ID: RF001	Nome do Requisito: Cadastro de Palavras-Chave
Descrição	O usuário deve poder cadastrar, listar, editar e excluir palavras-chave para buscar licitações.
Categoria: Funcional	Prioridades: Essencial
Informações	ID, Palavra, Data de cadastro, Status (Ativa/Inativa)
Regra de Negócio	Não possui
ID: RF002	Nome do Requisito: Cadastro de Usuários
Descrição	O administrador deve poder cadastrar, listar, editar e excluir usuários.
Categoria: Funcional	Prioridades: Essencial
Informações	ID, Nome, E-mail (único), Senha, Data de cadastro
Regra de Negócio	E-mail deve ser único.
ID: RF003	Nome do Requisito: Webscraping de Licitações
Descrição	O sistema deve buscar licitações no Diário Oficial do Estado de SP com base nas palavras-chave cadastradas para cada usuário
Categoria: Funcional	Prioridades: Essencial
Informações	ID, Título, Conteúdo, Data da publicação, Palavras-chave relacionadas, Link, Status (Lida/Não lida)
Regra de Negócio	Busca deve ser executada periodicamente, automaticamente ou manualmente pelo administrador.
ID: RF004	Nome do Requisito: Lista de Publicações
Descrição	O sistema deve listar as publicações encontradas e permitir filtros.
Categoria: Funcional	Prioridades: Essencial
Informações	ID, Título, Data da publicação, Status (Lida/Não lida)
Regra de Negócio	Filtragem por Data, Palavras-chave e Status.
ID: RF005	Nome do Requisito: Favoritos
Descrição	Usuários devem poder marcar publicações como favoritas e visualizar uma lista separada.
Categoria: Funcional	Prioridades: Importante
Informações	ID, Usuário, Publicação marcada como favorita
Regra de Negócio	Possibilidade de adicionar e remover favoritos.
ID: RF006	Nome do Requisito: Publicações Lidas
Descrição	Usuários devem poder marcar publicações como lidas e destacar essas publicações na lista.
Categoria: Funcional	Prioridades: Importante
Informações	ID, Usuário, Publicação marcada como lida
Regra de Negócio	Possibilidade de filtrar por publicações lidas/não lidas.
ID: RF007	Nome do Requisito: Interface de Administração
Descrição	O administrador deve ter acesso a uma interface para gerenciar palavras-chave, usuários e executar o scraping.
Categoria: Funcional	Prioridades: Essencial
Informações	Ações de gerenciamento para palavras-chave e usuários, botão para execução do scraping
Regra de Negócio	Não possui

Fonte: os autores

3.3 Requisitos Não Funcionais

Requisitos não funcionais são especificações que definem critérios para avaliar o funcionamento de um sistema, concentrando-se em como ele deve operar, em vez de o que ele deve fazer. Esses requisitos abrangem atributos de qualidade, como desempenho, segurança, usabilidade, confiabilidade e escalabilidade, que influenciam diretamente a experiência do usuário e a eficiência do sistema.

De acordo com o artigo Domatech (2024), os requisitos não funcionais estabelecem os critérios que medem o desempenho e a operação de um sistema. Eles não determinam o que o sistema deve fazer, mas especificam como ele deve se comportar em aspectos como desempenho, segurança e escalabilidade do software.

Além disso, a engenharia de requisitos define os requisitos não funcionais, ou requisitos de qualidade, como sendo os requisitos que especificam critérios que podem ser usados para descrever o funcionamento de um sistema, e não os comportamentos específicos.

Essas definições enfatizam que os requisitos não funcionais são fundamentais para garantir a qualidade e a eficiência de um sistema, estabelecendo padrões e restrições que orientam seu desenvolvimento e operação.

O Quadro 4 lista os RNF do projeto.

3.4 Casos de Uso

Casos de uso são utilizados para descrever os requisitos visíveis de um sistema. Eles são aplicados na fase de análise de requisitos de um projeto e contribuem para a elaboração de planos de teste e guias do usuário. Também são usados para criar e validar um design proposto, garantindo que ele atenda a todos os requisitos.

Casos de Uso também auxiliam na criação de cronogramas de projeto, ajudando a planejar o que será incluído em cada versão. Esse processo ajuda a identificar e lidar com riscos logo no início, resultando em sistemas mais robustos e de melhor qualidade. Segundo Schneider e Winters (2001), o uso de casos de uso também contribui para o planejamento e controle do desenvolvimento de sistemas.

O Diagrama de Casos de Uso do LictaSP e suas descrições são mostrados na Figura 3 e no Quadro 5, respectivamente.

Quadro 4 – Requisitos Não Funcionais do sistema

ID: RNF001	Nome do Requisito: Tecnologias Utilizadas
Descrição	O sistema deve ser desenvolvido utilizando Javascript, Python, Flask, <i>Selenium</i> , PostgreSQL e um <i>front-end</i> simples.
Categoria: Não Funcional	Prioridades: Essencial
Informações	Python para <i>back-end</i> , Flask como <i>framework</i> , PostgreSQL para banco de dados, HTML/CSS/JS para <i>front-end</i>
Regra de Negócio	Não possui
ID: RNF002	Nome do Requisito: Armazenamento de Dados
Descrição	O banco de dados deve armazenar palavras-chave, usuários, publicações e a relação de favoritos/lidas.
Categoria: Não Funcional	Prioridades: Essencial
Informações	Entidades: Palavras-chave, Usuários, Publicações, Favoritos/Lidas
Regra de Negócio	Não possui
ID: RNF003	Nome do Requisito: Execução do <i>Scraping</i>
Descrição	O <i>scraping</i> deve ser executado automaticamente uma vez por dia ou manualmente pelo administrador.
Categoria: Não Funcional	Prioridades: Essencial
Informações	Execução via linha de comando ou interface (github actions)
Regra de Negócio	Não possui
ID: RNF004	Nome do Requisito: Interface de Usuário
Descrição	A interface deve ser simples, priorizando funcionalidades como listagem, filtros e ações básicas.
Categoria: Não Funcional	Prioridades: Essencial
Informações	Listagem de publicações, filtros, botões de ação
Regra de Negócio	Não possui

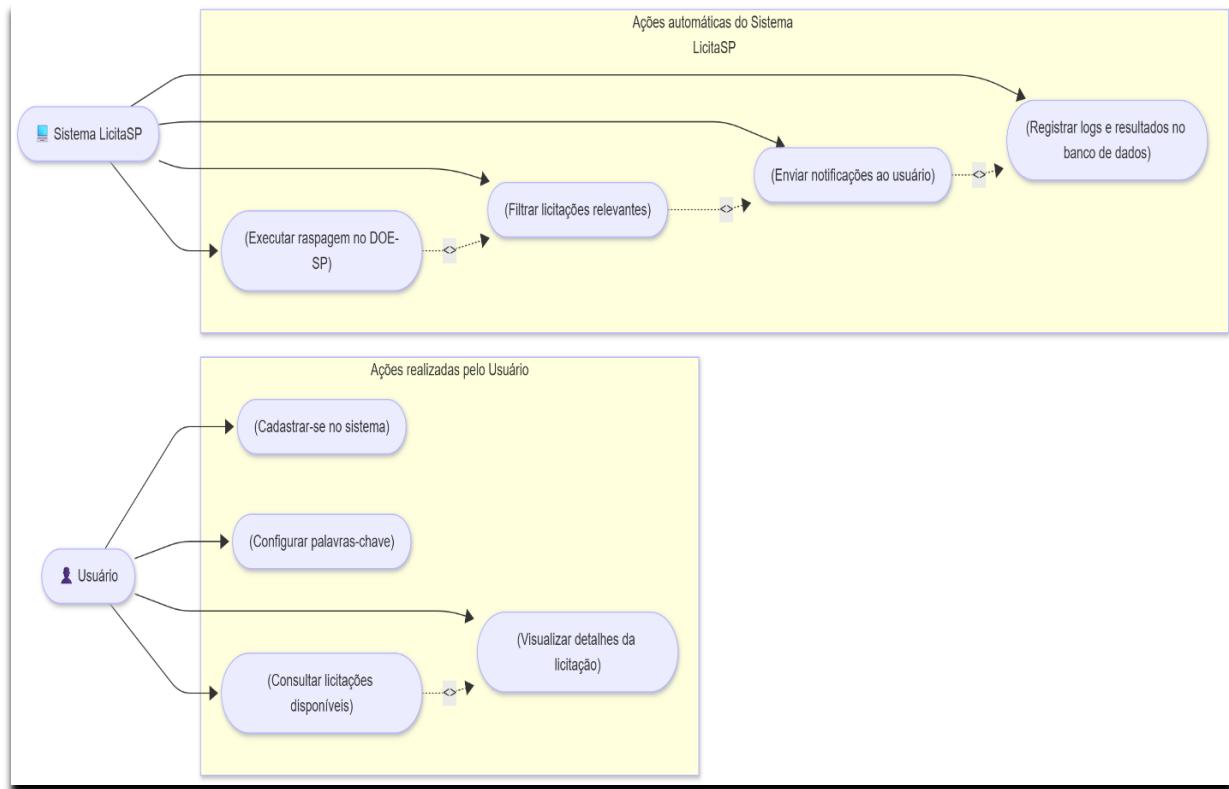
Fonte: os autores

3.5 Modelo e Diagrama Entidade-Relacionamento

O Modelo Entidade-Relacionamento (MER) é uma representação conceitual da organização do armazenamento do sistema, criada durante a etapa de modelagem. Ele tem como objetivo identificar as entidades, atributos e relacionamentos que compõem o domínio de informação, sem considerar ainda detalhes técnicos de implementação. Dessa forma, o MER descreve de maneira abstrata como os dados se relacionam entre si, oferecendo uma visão lógica e comprehensível para analistas e desenvolvedores.

Enquanto o MER define o que será representado e o Diagrama Entidade-Relacionamento (DER) mostra como isso se apresenta visualmente, tornando mais claro o entendimento da estrutura do banco de dados e garantindo coerência entre os dados modelados e a realidade do sistema.

Figura 3 – Diagrama de Casos de Uso



Fonte: os autores

Quadro 5 – Descrição dos Casos de Uso

Caso de Uso – Cadastrar-se no Sistema	
ID	UC 001
Descrição	Este caso de uso tem por objetivo permitir que o usuário crie uma conta no sistema Licitasp.
Ator Primário	Usuário do sistema
Pré-condição	Nenhuma
Cenário Principal	<ol style="list-style-type: none"> 1. O caso de uso inicia quando o usuário seleciona a opção “Cadastrar-se”. 2. O sistema carrega o formulário de cadastro. 3. O usuário informa nome, e-mail e senha. 4. O sistema valida os dados informados. 5. O sistema verifica se o e-mail já existe. 6. O sistema salva o novo registro no banco de dados. 7. O sistema confirma o cadastro e exibe mensagem de sucesso. 8. O sistema envia mensagem de boas-vindas ao usuário.
Pós-condição	Usuário cadastrado e apto a acessar o sistema.
Cenário Alternativo	<p>5a – O e-mail informado já existe</p> <p>5a.1 – O sistema exibe mensagem de erro e solicita outro e-mail.</p>
Caso de Uso – Configurar Palavras-Chave	

ID	UC 002
Descrição	Este caso de uso tem por objetivo permitir que o usuário configure palavras-chave utilizadas para buscar licitações.
Autor Primário	Usuário do sistema
Pré-condição	Usuário deve estar autenticado no sistema.
Cenário Principal	<ol style="list-style-type: none"> 1. O usuário acessa a opção “Cadastrar Palavras”. 2. O sistema exibe o campo para inserção de palavras-chave. 3. O usuário adiciona uma ou mais palavras. 4. O sistema grava as palavras no perfil do usuário. 5. O sistema confirma o salvamento e exibe mensagem de sucesso.
Pós-condição	Palavras-chave registradas para uso na busca automática.
Cenário Alternativo	<p>3a – O usuário não informa nenhuma palavra.</p> <p>3a.1 – O sistema solicita pelo menos uma palavra-chave.</p>
Caso de Uso – Consultar Licitações Disponíveis	
ID	UC 003
Descrição	Este caso de uso tem por objetivo permitir que o usuário visualize as licitações coletadas e filtradas pelo sistema.
Autor Primário	Usuário do sistema
Pré-condição	Usuário deve estar autenticado no sistema.
Cenário Principal	<ol style="list-style-type: none"> 1. O usuário acessa a opção “Consultar Licitações”. 2. O sistema exibe a lista de licitações disponíveis. 3. O usuário aplica filtros de data, órgão ou palavra-chave. 4. O sistema apresenta os resultados correspondentes.
Pós-condição	Usuário visualiza as licitações disponíveis no momento.
Cenário Alternativo	<p>4a – Nenhuma licitação encontrada.</p> <p>4a.1 – O sistema exibe mensagem “Nenhuma licitação disponível no momento”.</p>
Caso de Uso – Visualizar Detalhes da Licitação	
ID	UC 004
Descrição	Este caso de uso tem por objetivo permitir que o usuário visualize detalhes completos de uma licitação específica.
Autor Primário	Usuário do sistema
Pré-condição	Usuário deve ter acessado a lista de licitações.
Cenário Principal	<ol style="list-style-type: none"> 1. O usuário seleciona uma licitação da lista. 2. O sistema carrega os detalhes da licitação. 3. O sistema exibe órgão, objeto, data e link do edital.
Pós-condição	Detalhes da licitação exibidos ao usuário.
Cenário Alternativo	<p>2a – Licitação não encontrada.</p> <p>2a.1 – O sistema exibe mensagem de erro.</p>
Caso de Uso – Executar Raspagem no DOE-SP	
ID	UC 005

Descrição	Este caso de uso tem por objetivo realizar automaticamente a coleta de publicações do Diário Oficial do Estado de São Paulo.
Autor Primário	Sistema Licitasp
Pré-condição	Sistema configurado e com acesso à internet.
Cenário Principal	<ol style="list-style-type: none"> 1. O sistema inicia a rotina de raspagem programada. 2. O sistema acessa o portal do DOE-SP. 3. O sistema coleta as publicações disponíveis. 4. O sistema armazena temporariamente os dados coletados.
Pós-condição	Publicações do DOE-SP coletadas e prontas para filtragem.
Cenário Alternativo	<p>2a – DOE-SP indisponível.</p> <p>2a.1 – O sistema registra o erro e reagenda a execução.</p>
Caso de Uso – Filtrar Licitações Relevantes	
ID	UC 006
Descrição	Este caso de uso tem por objetivo aplicar filtros sobre as publicações coletadas com base nas palavras-chave dos usuários.
Autor Primário	Sistema Licitasp
Pré-condição	Existência de publicações coletadas e palavras-chave cadastradas.
Cenário Principal	<ol style="list-style-type: none"> 1. O sistema carrega as publicações coletadas. 2. O sistema compara o conteúdo com as palavras-chave dos usuários. 3. O sistema separa as licitações relevantes. 4. O sistema salva os resultados filtrados.
Pós-condição	Licitações filtradas e associadas aos usuários correspondentes.
Cenário Alternativo	<p>3a – Nenhuma licitação corresponde às palavras-chave.</p> <p>3a.1 – O sistema registra o evento sem enviar notificações.</p>
Caso de Uso – Enviar Notificações ao Usuário	
ID	UC 007
Descrição	Este caso de uso tem por objetivo enviar automaticamente notificações aos usuários sobre novas licitações encontradas.
Autor Primário	Sistema Licitasp
Pré-condição	Existência de licitações filtradas para um ou mais usuários.
Cenário Principal	<ol style="list-style-type: none"> 1. O sistema identifica usuários com licitações novas. 2. O sistema gera o conteúdo das notificações. 3. O sistema envia e-mails e alertas via painel. 4. O sistema registra o envio no log.
Pós-condição	Usuários notificados sobre novas licitações.
Cenário Alternativo	3a – Falha no envio.

	3a.1 – O sistema registra o erro e tenta novamente mais tarde.
Caso de Uso – Registrar Logs e Resultados	
ID	UC 008
Descrição	Este caso de uso tem por objetivo registrar logs de execução, erros e resultados no banco de dados.
Autor Primário	Sistema Licitasp
Pré-condição	Sistema em execução e eventos ocorrendo.
Cenário Principal	<ol style="list-style-type: none"> 1. O sistema registra cada rotina executada. 2. O sistema armazena logs de sucesso e falha. 3. O sistema mantém histórico das execuções.
Pós-condição	Logs e resultados armazenados no banco de dados.
Cenário Alternativo	<p>2a – Falha na gravação.</p> <p>2a.1 – O sistema salva os logs localmente e tenta novamente posteriormente.</p>

Fonte: os autores

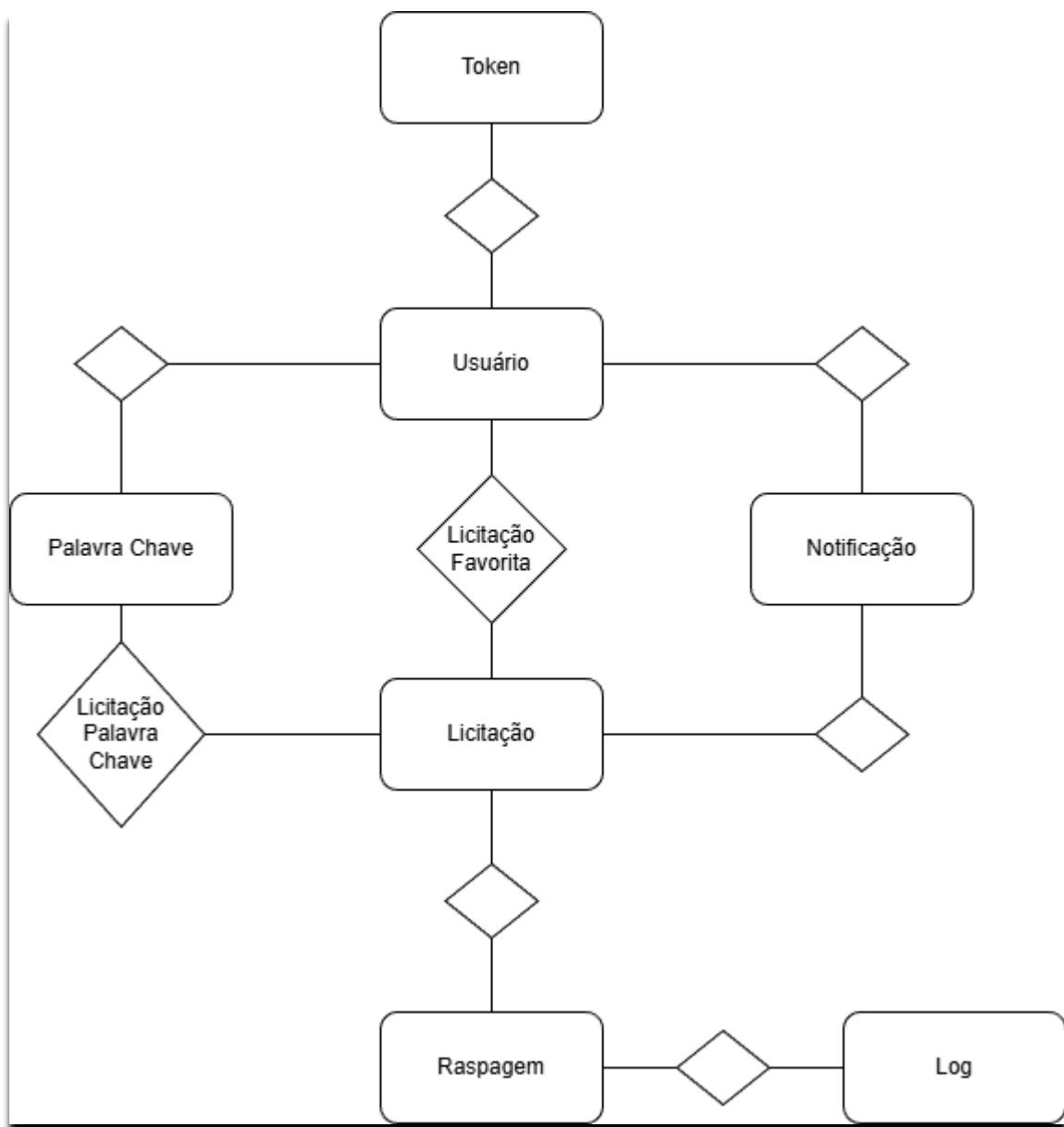
Conforme destacam Mesquita e Finger (1998), o processo de modelagem de dados envolve a transformação gradual do modelo conceitual para o modelo lógico e, posteriormente, para o modelo físico. Essa transição tem como objetivo garantir que a estrutura conceitual dos dados seja mantida ao longo do processo de implementação, permitindo um mapeamento fiel das entidades e relacionamentos no banco de dados final. Nesse sentido, o DER é uma ferramenta fundamental de validação e comunicação entre o projetista de dados e os demais envolvidos no desenvolvimento do sistema. As definições do MER e do DER do projeto são apresentadas nas Figuras 4 e 5, respectivamente.

3.6 Diagrama de Classes (DC)

Segundo Júnior (2020), o Diagrama de Classes é uma representação gráfica fundamental na modelagem de sistemas orientados a objetos, utilizada para definir a estrutura e a relação entre as classes de um sistema. Ele descreve os atributos, operações e os relacionamentos entre as classes, além de suas características e comportamentos esperados.

O DC oferece uma visão estática do sistema, permitindo identificar e organizar seus principais componentes de forma a facilitar o desenvolvimento e a manutenção. Cada classe representa uma entidade com propriedades (atributos) e comportamentos (métodos), enquanto os relacionamentos, como associação, dependência, agregação e composição, mostram como essas classes interagem entre si dentro do sistema.

Figura 4 – Diagrama MER



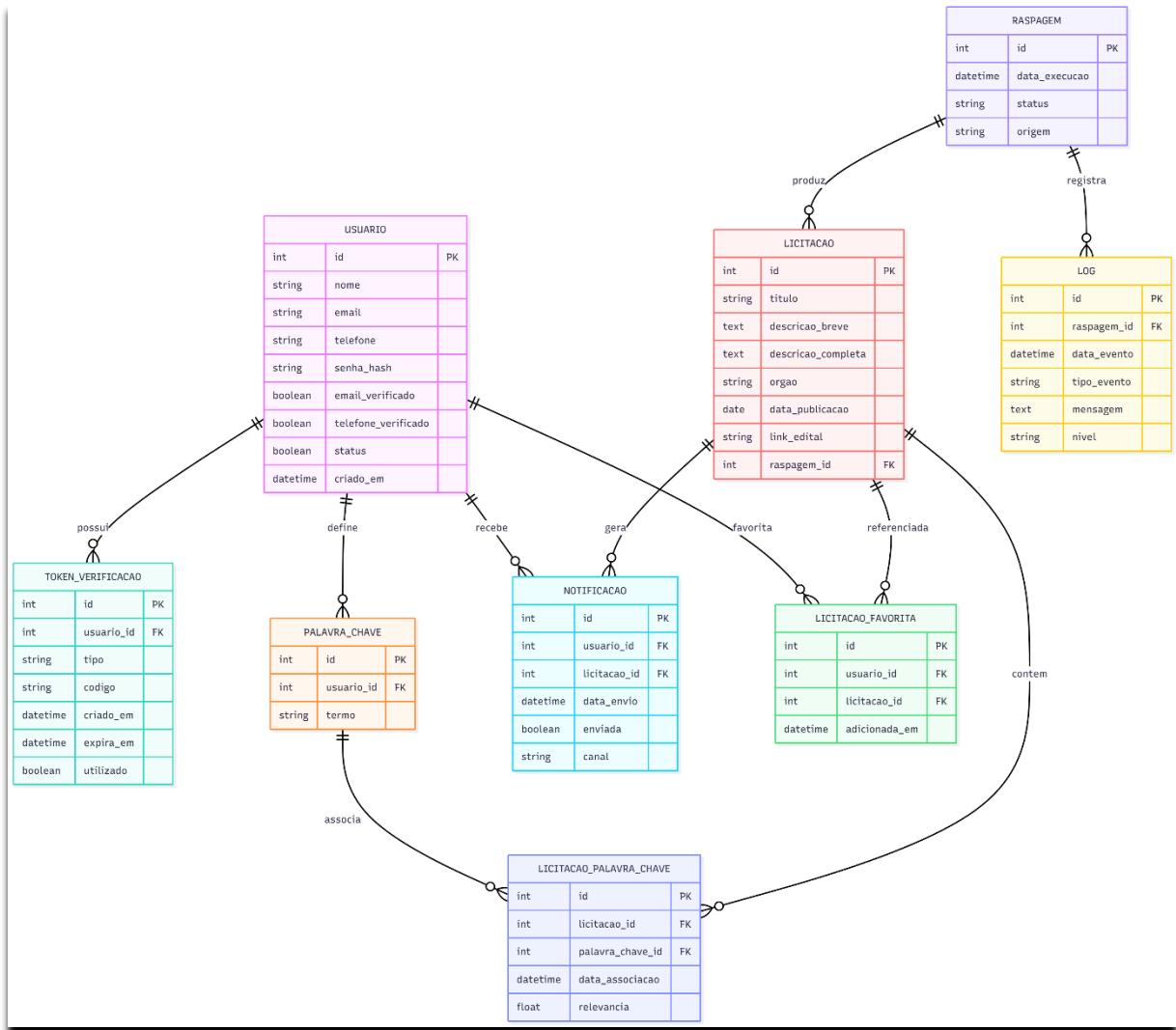
Fonte: os autores

O DC é uma ferramenta essencial para o planejamento e a documentação de software, promovendo uma compreensão mais clara do sistema entre os desenvolvedores e outros *stakeholders* envolvidos no projeto (Júnior, 2020).

O Diagrama de Classe foi utilizado nesse projeto com o intuito de otimizar a aplicação bem como o banco de dados, e calcular métricas do projeto, possibilitando, assim, avaliar o custo aproximado do desenvolvimento como um todo.

O DC do projeto é mostrado na Figura 6.

Figura 5 – Diagrama DER

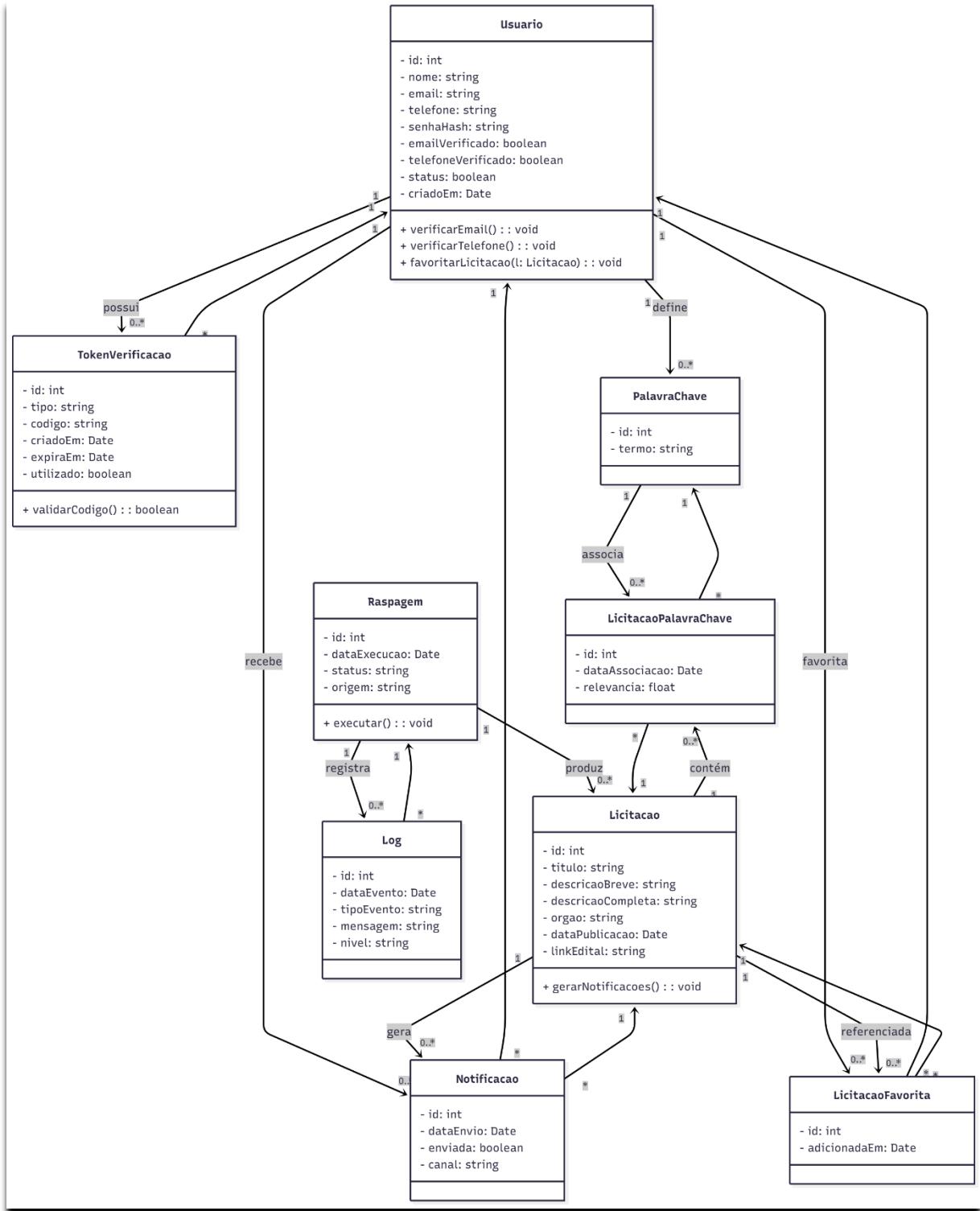


Fonte: os autores

3.7 Diagrama de Atividades (DA)

O Diagrama de Atividades ilustra como as ações se inter-relacionam e se encadeiam, utilizando elementos como nós de ação, nós de decisão, bifurcações, junções e nós de início e fim. Essas representações ajudam a visualizar a lógica de execução das atividades, considerando o fluxo de controle e as condições para a transição entre atividades. O DA é valioso para identificar e documentar as interações entre processos, auxiliando na análise e na validação de fluxos de atividades.

Figura 6 – Diagrama de Classe



Fonte: os autores

Segundo Mello (2011) o DA é uma ferramenta da *Unified Modeling Language* (UML) usada para representar o fluxo de atividades em um sistema ou processo. Ele descreve a sequência de ações ou tarefas e os caminhos que podem ser seguidos, oferecendo uma visão dinâmica dos processos de negócio e da lógica dos algoritmos do sistema.

A escolha do Diagrama de Atividades para o projeto (Figura 7) se justifica por diversas razões, relacionadas à clareza, eficiência e organização do fluxo de operações. Em um sistema que lida com processos sequenciais e interdependentes, como a busca de licitações e notificação do usuário, o DA se torna uma ferramenta essencial para visualizar e entender cada etapa envolvida.

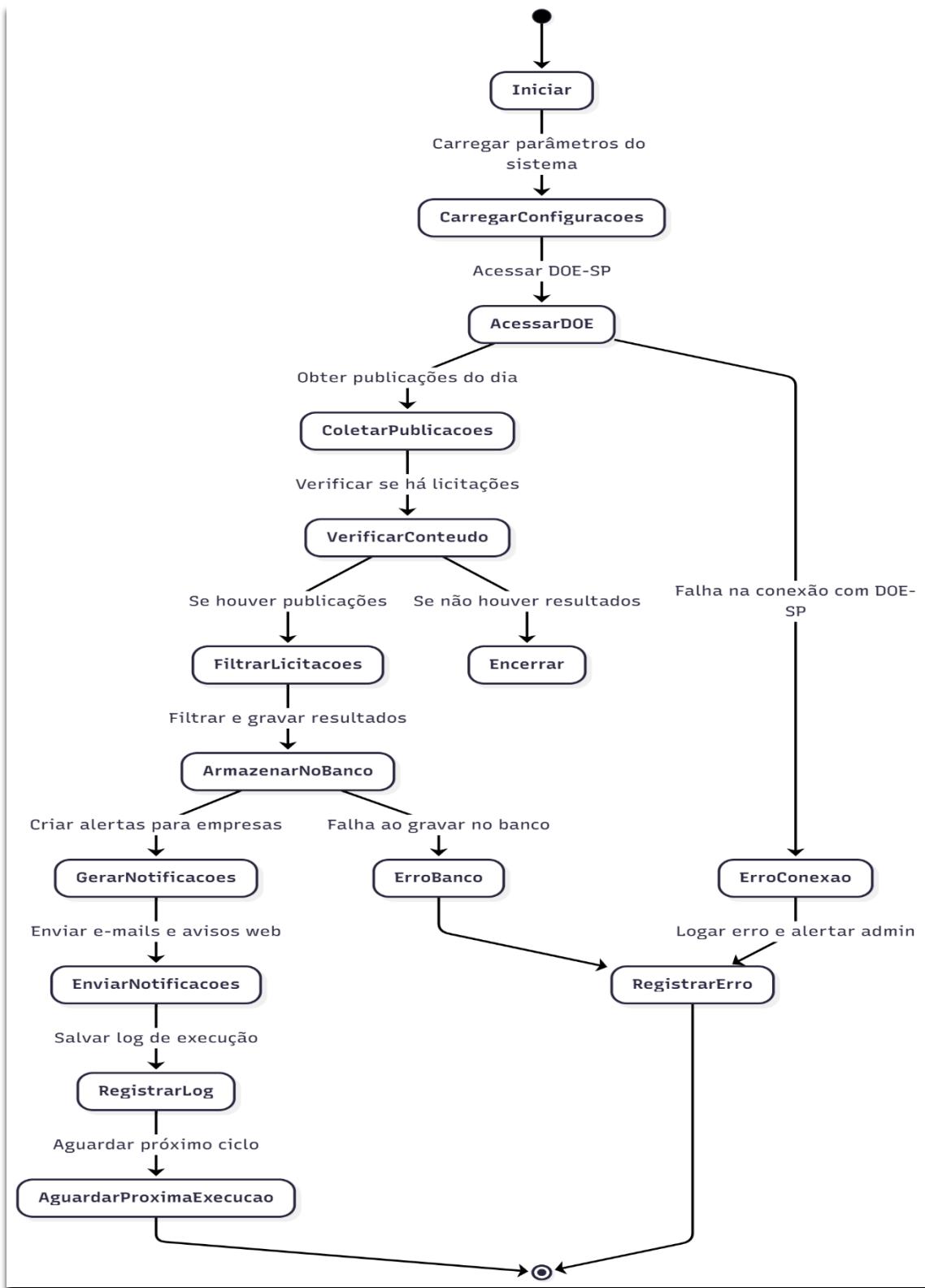
3.8 Diagrama de Estados (DE)

Um Diagrama de Estados, também conhecido como Diagrama de Máquina de Estados, é uma representação gráfica que descreve o comportamento dinâmico de um sistema, destacando os diferentes estados que um objeto ou componente pode assumir e as transições entre esses estados em resposta a eventos. Essa ferramenta é amplamente utilizada na Engenharia de Software para modelar sistemas reativos e orientados a eventos.

A importância de um DE reside em sua capacidade de modelar comportamentos complexos permitindo a visualização clara de como um sistema responde a diversos eventos, facilitando a compreensão de comportamentos intricados; identificar inconsistências pois, ao mapear todos os possíveis estados e transições, é possível detectar lacunas ou conflitos no design do sistema, contribuindo para a melhoria da qualidade do software; e comunicação eficaz servindo como uma ferramenta de comunicação entre membros da equipe de desenvolvimento, analistas e partes interessadas, garantindo que todos compartilhem uma compreensão comum do funcionamento do sistema. Em resumo, os Diagramas de Estados são essenciais para a modelagem precisa do comportamento de sistemas, auxiliando no desenvolvimento de soluções mais robustas e eficientes.

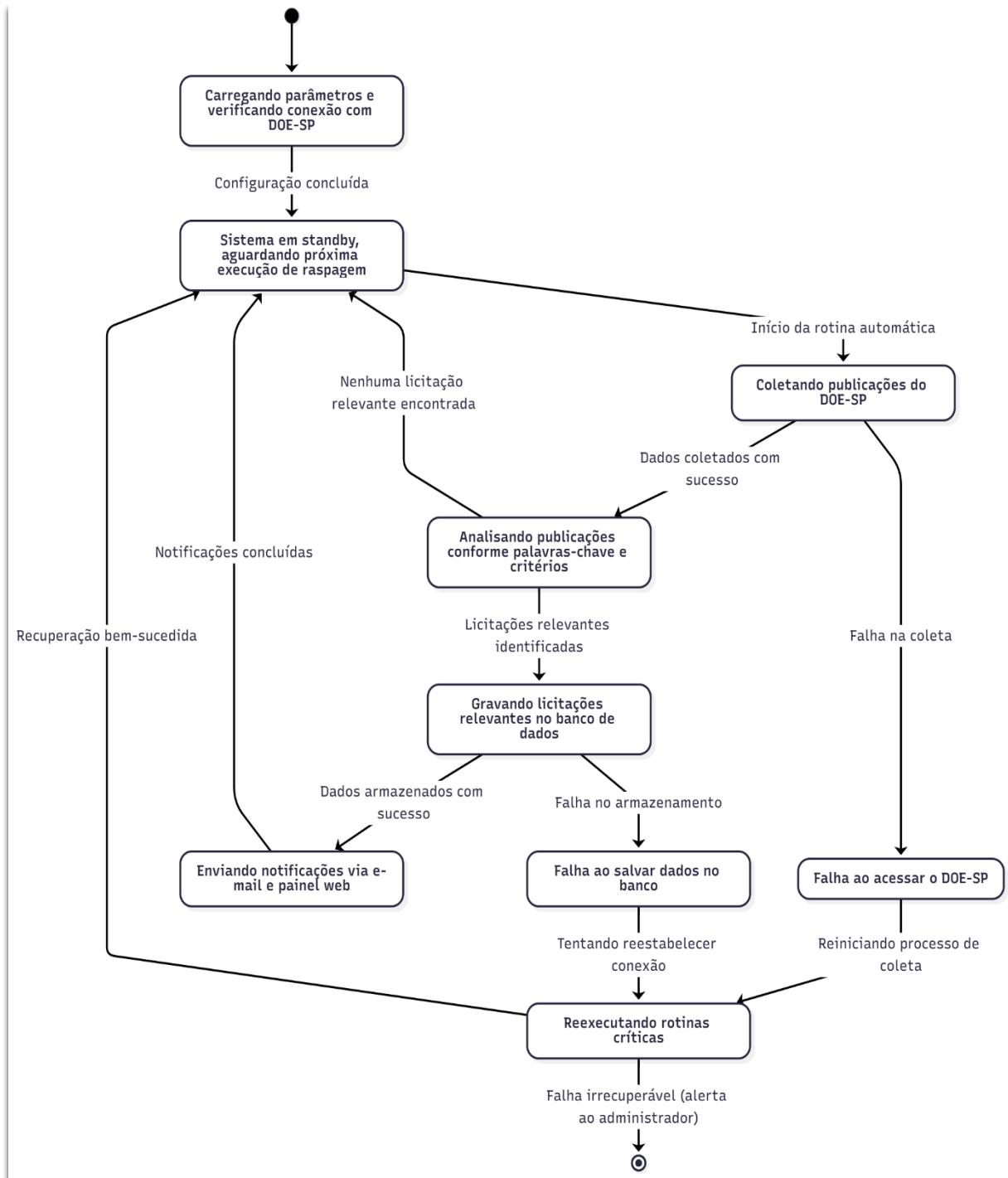
A Figura 8 apresenta o DE do LictaSP.

Figura 7 – Diagrama de Atividades



Fonte: os autores

Figura 8 – Diagrama de Estados



Fonte: os autores

3.9 Diagrama de Sequência (DS)

Segundo Bacurau, Leal e Ramos (2022), o Diagrama de Sequência é um dos elementos da UML usados para a modelagem de sistemas orientados a objetos, oferecendo uma visão detalhada das interações entre objetos em um sistema ao longo do tempo. O DS representa a troca de mensagens entre diferentes partes do sistema e as ordens em que essas trocas acontecem, focando na cronologia das interações para ilustrar como cada objeto responde a estímulos e chamadas. Dessa forma, ele é especialmente útil para a compreensão dos processos dinâmicos e da lógica de execução de um sistema, facilitando a análise das operações e do fluxo de informações entre componentes de software.

No projeto LictaSP o DC (Figura 9) foi utilizado para documentar e visualizar a interação entre os diferentes elementos envolvidos no sistema, como o *scraper* de dados, armazenamento no banco de dados e notificação do cliente. O objetivo é ilustrar de maneira precisa como essas partes se comunicam para realizar tarefas críticas, garantindo que todas as etapas do processo sejam corretamente seguidas e que os dados fluam adequadamente entre os módulos.

3.10 Matriz de Rastreabilidade

A Matriz de Rastreabilidade é uma ferramenta usada para garantir o acompanhamento e a verificação dos requisitos em todas as etapas do desenvolvimento, promovendo maior controle de qualidade e alinhamento com as expectativas do projeto (Trindade e Lucena, 2016).

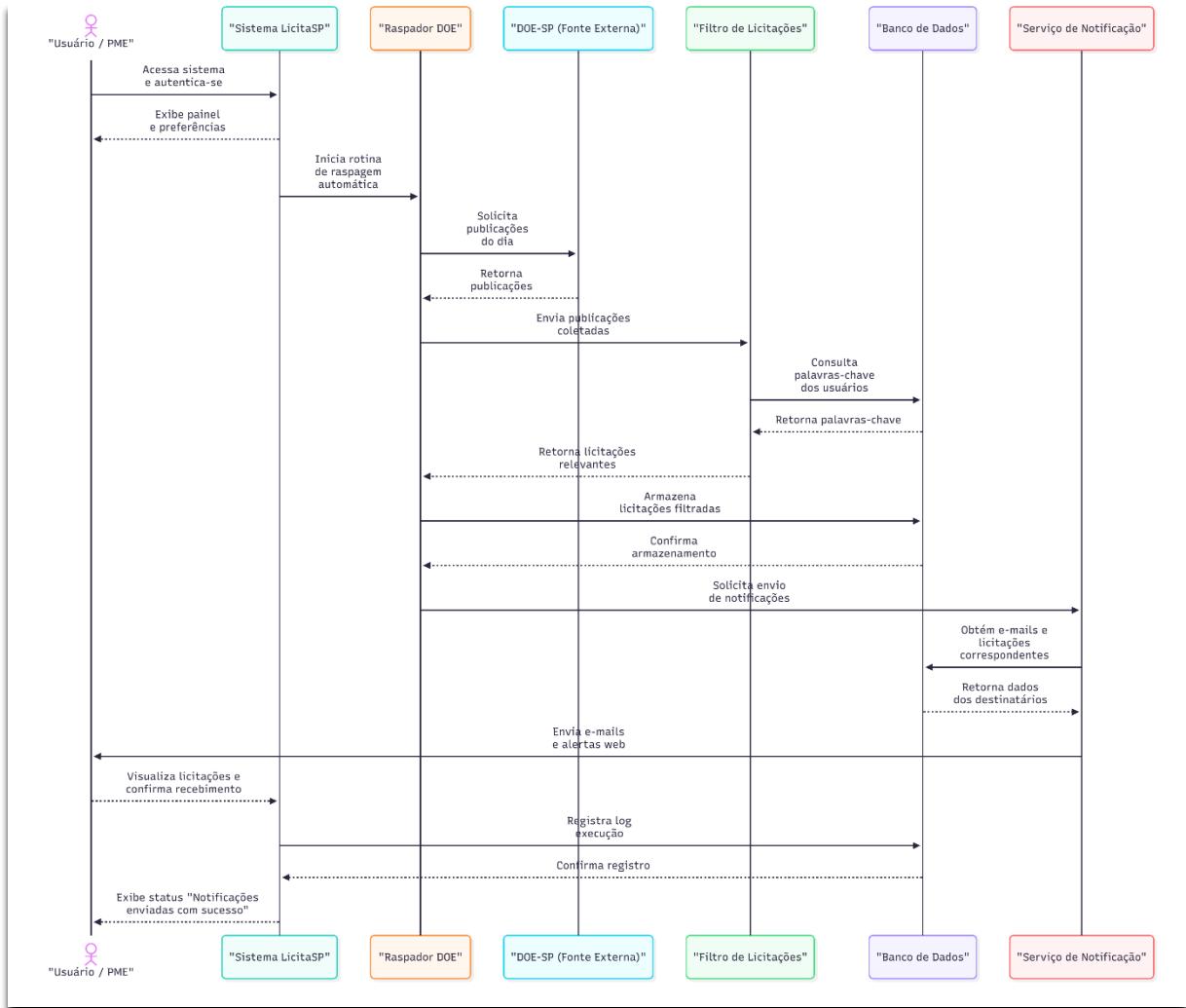
Esse artefato é essencial porque ajuda a identificar lacunas no desenvolvimento, facilita o gerenciamento de mudanças e melhora a comunicação entre a equipe. Ela assegura que cada requisito inicial seja coberto por uma fase de desenvolvimento ou teste, promovendo maior controle de qualidade e alinhamento com as expectativas dos stakeholders.

4 Ferramentas e Métodos

O desenvolvimento do sistema LictaSP foi guiado por um conjunto estruturado de ferramentas e métodos de análise, planejamento e modelagem de software. O projeto teve início com a identificação do problema e das necessidades dos usuários, por meio de uma entrevista exploratória com um cliente em potencial, proprietário de

uma pequena empresa que relatou um pouco das dores diárias que PMEs tem em comum, e assim surgiu o *insight* para iniciar a análise de viabilidade do projeto.

Figura 9 – Diagrama de Sequência



Fonte: os autores

Quadro 6 – Matriz de Rastreabilidade

ID Req.	Descrição	Tipo	Casos de Uso Relacionados	Módulo / Componente	Critério de Aceitação / Teste
REQ-001	O sistema deve permitir o cadastro de novos usuários.	Funcional	UC001 – Cadastrar-se no sistema	Módulo de Autenticação / Usuário	Ao preencher e confirmar o formulário, o sistema deve salvar os dados e exibir mensagem de sucesso.

ID Req.	Descrição	Tipo	Casos de Uso Relacionados	Módulo / Componente	Critério de Aceitação / Teste
REQ-002	O sistema deve permitir a configuração de palavras-chave para filtragem.	Funcional	UC002 – Configurar palavras-chave	Módulo de Perfil / Filtro	Usuário consegue salvar palavras e o sistema utiliza essas configurações na filtragem.
REQ-003	O sistema deve permitir consultar as licitações coletadas.	Funcional	UC003 – Consultar licitações disponíveis	Módulo de Consulta	Deve exibir licitações correspondentes aos filtros aplicados.
REQ-004	O sistema deve exibir detalhes completos da licitação selecionada.	Funcional	UC004 – Visualizar detalhes da licitação	Módulo de Exibição / Licitação	Exibir título, órgão, data e link do edital.
REQ-005	O sistema deve realizar a raspagem automática do DOE-SP diariamente.	Funcional	UC005 – Executar raspagem no DOE-SP	Módulo de Raspagem	O sistema deve coletar novas publicações automaticamente e registrar execução.
REQ-006	O sistema deve aplicar filtros nas licitações coletadas com base nas palavras-chave dos usuários.	Funcional	UC006 – Filtrar licitações relevantes	Módulo de Filtro / Processamento	Somente licitações que contenham as palavras definidas devem ser armazenadas.

ID Req.	Descrição	Tipo	Casos de Uso Relacionados	Módulo / Componente	Critério de Aceitação / Teste
REQ-007	O sistema deve enviar notificações automáticas aos usuários sobre novas licitações.	Funcional	UC007 – Enviar notificações ao usuário	Módulo de Notificação	Usuário deve receber e-mail ou alerta no painel em até 5 minutos após coleta.
REQ-008	O sistema deve registrar logs de execução e erros no banco de dados.	Funcional	UC008 – Registrar logs e resultados	Módulo de Log / Banco de Dados	Cada rotina executada deve gerar registro no log.
REQ-009	O sistema deve estar disponível 24h por dia, exceto em períodos de manutenção.	Não Funcional	UC005, UC006, UC007, UC008	Infraestrutura / Servidor	O uptime deve ser $\geq 99\%$.
REQ-010	O sistema deve armazenar dados de forma segura (criptografia de senhas e conexões seguras).	Não Funcional	UC001, UC002	Segurança / Banco de Dados	Dados sensíveis devem ser criptografados e acessos autenticados.
REQ-011	O sistema deve apresentar interface responsiva e de fácil uso.	Não Funcional	UC001 a UC004	Interface Web (Flask / Bootstrap)	Interface adaptável a desktop e mobile sem perda de funcionalidade.

Fonte: os autores

Com base nas informações obtidas na entrevista, foi possível realizar a análise SWOT, identificando os pontos fortes, fracos, oportunidades e ameaças do cenário atual de monitoramento de licitações. Em seguida, foi elaborada a Matriz 5W2H, com o objetivo de planejar as ações estratégicas para o desenvolvimento do sistema, definindo claramente o que seria feito, por quem, quando, por quê, como e quanto custaria.

Essas duas análises forneceram uma visão sólida para o levantamento de requisitos funcionais e não funcionais, que serviram como base para a modelagem dos diagramas UML (Casos de Uso, Atividades, Classes, Sequência e Estados), além do diagrama BPMN, que representou graficamente o fluxo de interação entre o usuário e o sistema, destacando as etapas automáticas de raspagem de dados e notificação.

Após a etapa de modelagem, iniciou-se o desenvolvimento do protótipo web, seguindo princípios de modularidade, escalabilidade e segurança.

4.1 Métodos Adotados

Os métodos utilizados para conduzir o projeto são detalhados a seguir.

Entrevista com cliente em potencial que serviu como base para compreender o problema e validar hipóteses sobre as necessidades das pequenas e médias empresas (PME).

Análise SWOT foi utilizada para mapear forças (automação e acessibilidade), fraquezas (dependência do DOE-SP), oportunidades (ampliar participação de PME) e ameaças (mudanças em editais oficiais).

Matriz 5W2H foi a ferramenta de planejamento que ajudou a estruturar o plano de ação e priorizar etapas de desenvolvimento.

O Levantamento de Requisitos determinou as funcionalidades principais e complementares do sistema.

A modelagem BPMN representou o fluxo operacional do sistema com duas raias horizontais (Usuário e Sistema Licitasp), evidenciando as decisões automáticas do sistema.

E os Diagramas UML forneceram suporte técnico para a modelagem estrutural e comportamental do sistema.

4.2 Ferramentas Utilizadas

As ferramentas escolhidas para o projeto foram selecionadas com base em sua eficiência, escalabilidade e ampla adoção na comunidade de desenvolvimento web. Todas possuem licença *open source* e documentação oficial, o que garante liberdade de uso e aprendizado contínuo.

Flask foi o *framework* usado para a implementação do *front-end* do protótipo.

Python foi a linguagem de programação usada no *back-end* para processar raspagens e notificações.

Os dados foram mantidos no Sistema Gerenciador de Bancos de Dados Relacionais (SGBDR) PostgreSQL, de código aberto.

Os diagramas UML foram criados com o Mermaid Chart.

O diagrama BPMN foi criado na plataforma Miro, ferramenta on-line para edição e visualização de diagramas.

Os diagramas da análise do banco de dados foram criados com Drawio.

O Visual Studio Code foi a IDE utilizada no desenvolvimento do código fonte, que foi mantido na plataforma Github a ferramenta para versionamento de código, e possibilitar a portabilidade da aplicação a diferentes plataformas de operação, foi utilizado o Docker, ferramenta *Devops* para criação de ambientes reproduzíveis e escaláveis.

4.3 Métodos de Execução da Interface

A interface do LictaSP foi desenvolvida com base em três princípios fundamentais: usabilidade seguindo diretrizes de UX/UI modernas (*layout* limpo e navegação intuitiva); acessibilidade da aplicação web com visualização em dispositivos móveis; e escalabilidade com a implementação de componentes modulares.

5 Desenvolvimento

O desenvolvimento do protótipo funcional do LictaSP concentrou-se na criação da interface *web*, representando a interação do usuário com o sistema e a estrutura geral das funcionalidades planejadas.

Essa etapa teve o objetivo de validar o conceito, a usabilidade e o fluxo de navegação. O protótipo foi desenvolvido utilizando Python, Flask e Bootstrap, para garantir modularidade, responsividade e escalabilidade.

As telas foram elaboradas, permitindo visualizar a estrutura e o comportamento do sistema de forma interativa. O foco foi demonstrar a jornada do usuário, desde o cadastro até a visualização de licitações filtradas e notificações recebidas.

Durante essa fase, foram desenvolvidas as telas apresentadas nas Figuras 10 a 15 que são autoexplicativas.

Figura 10 – Dashboard

The screenshot shows the Licitasp dashboard. On the left, there's a sidebar with the Licitasp logo, the name 'Rafael' and 'Desenvolvedor', and a 'MENU' section containing 'Home' and 'Configurações'. The main area has a header with the Licitasp logo. Below it is a 'Publicações' section with tabs for 'Todas', 'Avisos de Licitações', 'Avisos de ratificação', and 'Comunicados'. It displays three items: 1) 'DIÁRIO OFICIAL Avisos de Licitações' from 20/01/2025, 2) 'AVISOS DE RATIFICAÇÃO Edital N° 6.320' from 20/01/2025, and 3) 'PROJETOS DE LEI PL 9543/2018' from 20/01/2025. To the right is a search bar and a calendar for November 2025, with the 8th highlighted.

Fonte: os autores

Figura 11 – Tela de login

The screenshot shows the Licitasp login page. On the left is the Licitasp logo. On the right, there are input fields for 'Usuário' (User) and 'Senha' (Password), both with placeholder text and icons. Below them is a large blue 'Entrar' (Enter) button. Underneath the button, there's a link 'Não tem conta? Registrar' (Don't have an account? Register) and a link 'Esqueceu a senha?' (Forgot password?).

Fonte: os autores

Figura 12 – Tela de registro

The screenshot shows the registration interface for the Licitasp system. At the top left is the Licitasp logo, which consists of a blue circle containing the letters 'LS'. Below the logo is the word 'LICITASP' in a bold, dark blue sans-serif font. Underneath the logo is a black rectangular button with the white text 'Crie sua conta'. To the right of the logo is a registration form. The form fields are labeled 'Nome completo', 'E-mail', 'Usuário', 'Senha', and 'Confirmar senha'. Each label is followed by a text input field. Below the input fields is a green rounded rectangle button with the white text 'Criar Conta'. At the bottom right of the form area is a small link 'Já tem conta? Entrar'.

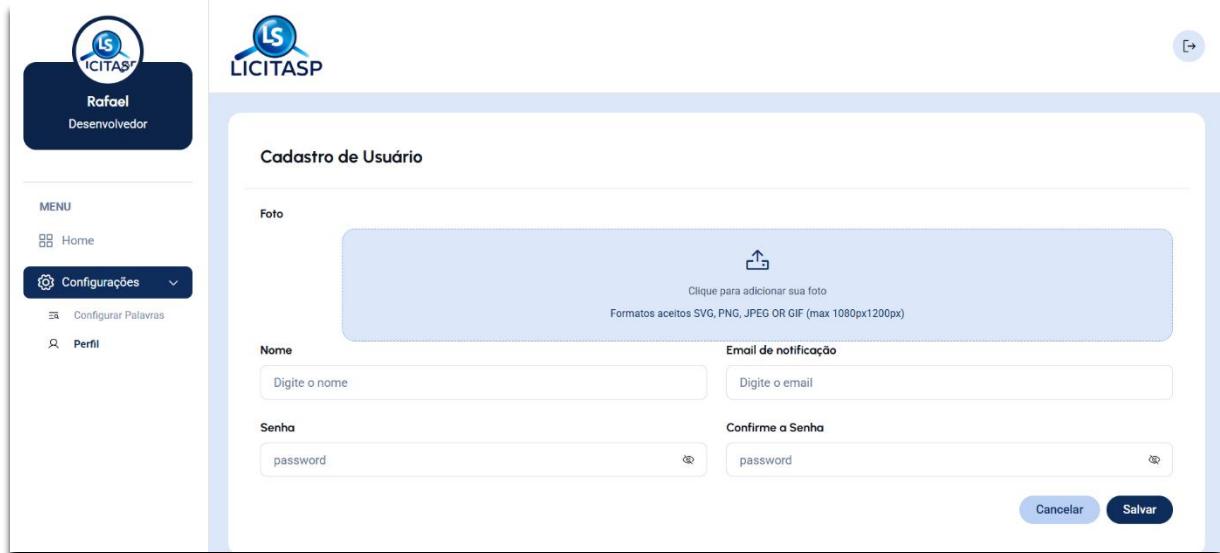
Fonte: os autores

Figura 13 – Tela de recuperação de senha

The screenshot shows the password recovery interface for the Licitasp system. At the top left is the Licitasp logo. To the right of the logo is a form with a single text input field labeled 'Email' with the placeholder 'Digite seu email'. Below the input field is a dark blue rounded rectangle button with the white text 'Recuperar Senha'. At the bottom right of the form area is a small link 'Lembrou sua senha? Entrar'.

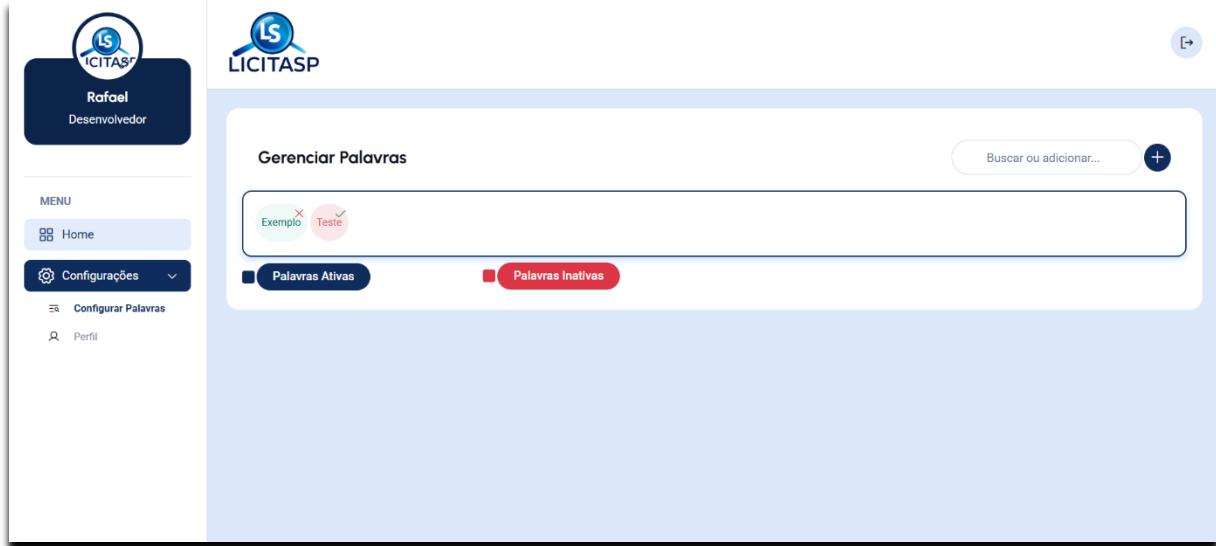
Fonte: os autores

Figura 14 – Tela de gerenciamento do perfil



Fonte: os autores

Figura 15 – Tela de configuração de palavras-chave



Fonte: os autores

Embora o protótipo não possua ainda integração com um servidor de operação em produção, a arquitetura de *back-end* foi planejada com base em boas práticas de desenvolvimento Python e foi dimensionada para ser escalável.

O Diagrama de Classes modelado no projeto representa essa estrutura: as classes foram projetadas com baixo acoplamento e alta coesão, permitindo que,

futuramente, o sistema possa crescer de forma modular, adicionando novas fontes de dados e tipos de notificações.

Dessa forma, o protótipo cumpre o papel de prova de conceito visual e funcional, servindo como base para o desenvolvimento completo em futuras fases do projeto.

6 Resultados e Discussão

A Análise de Portabilidade tem como objetivo avaliar o grau de facilidade com que o sistema pode ser transferido entre diferentes ambientes de execução, plataformas e dispositivos.

Como o Licitasp foi desenvolvido em formato de protótipo *web*, sua arquitetura foi planejada para ser altamente portátil, minimizando dependências específicas de ambiente.

No Quadro 7 são descritos os principais aspectos da análise.

Quadro 7 – Análise de Portabilidade

Critério	Descrição	Resultado da Análise
Ambiente de Desenvolvimento	O protótipo foi construído com Python, executável em qualquer sistema que suporte Python 3.12 ou Docker	Portável entre Windows, Linux e marcos
Banco de Dados	O <i>back-end</i> planejado utiliza PostgreSQL, amplamente suportado por provedores de nuvem e contêineres Docker.	Portável entre servidores locais e em nuvem <i>serverless</i> (AWS, Render, Railway).
Interface Web	O protótipo foi desenvolvido com design responsivo, compatível com navegadores modernos e dispositivos móveis	Portável entre <i>desktops</i> , <i>tablets</i> e <i>smartphones</i>
Dependências de Software	Todas as bibliotecas são <i>open source</i> , sem dependência de plataformas proprietárias	Fácil reimplementação e manutenção
Conteinerização futura	O sistema poderá ser empacotado em Docker para padronização de ambiente e escalabilidade	Elevada portabilidade operacional

Fonte: os autores

6.2 Cálculo de Métricas

A estimativa de custo do software por meio da Análise de Pontos de Função (APF) é uma técnica amplamente utilizada para mensurar o tamanho funcional de um sistema, independentemente da linguagem de programação empregada. Segundo Santos e Canal (2020), essa metodologia possibilita calcular o esforço de desenvolvimento a partir de parâmetros objetivos, considerando fatores como entradas, saídas, consultas, arquivos e interfaces, além do Fator de Ajuste Técnico (TAF do termo em inglês *Technical Adjustment Factor*) que estabelece o nível de

influência técnica do sistema. No projeto, a aplicação da APF permitiu estimar de forma estruturada o prazo e o custo de desenvolvimento, resultando em uma avaliação aproximada do investimento necessário para a implementação da solução (Devmedia, 2016).

O primeiro passo foi o levantamento das funções do sistema, considerando as principais categorias definidas pelo método APF: entradas, saídas, consultas, arquivos e interfaces. Cada uma dessas funções foi detalhada de acordo com o número de ocorrências e classificada quanto à complexidade (simples, média ou complexa), atribuindo-se pesos correspondentes conforme o padrão de referência.

Com base nos dados coletados no modelo de dados e nos requisitos funcionais, o resultado foi a soma ponderada de cada tipo de função, resultando em um total de 263 Pontos de Função Brutos (FP'b).

Em seguida, foi calculado o Fator de Ajuste (FA), composto por 14 características técnicas, como comunicação de dados, performance, volume de transações, facilidade de implantação e reutilização de código. Cada fator foi avaliado com nota de 0 a 5, e o total obtido foi 70 pontos.

Aplicando a fórmula padrão $FA = 0,65 + (0,01 \times NI)$, obteve-se o fator 1,35.

O total ajustado de Pontos de Função (FP'r) foi então obtido multiplicando-se o FP'b pelo FA: 355 Pontos de Função Ajustados (FP'r).

Com essa métrica, foi necessário estimar a quantidade média de linhas de código (LOC) por Ponto de Função, de acordo com a linguagem escolhida.

O sistema foi implementado utilizando Python, Bootstrap e PostgreSQL enquadrando-se na categoria Geradores de Código, com uma média de 15 LOC por FP.

Assim, o total estimado de linhas de código (KLOC) foi: $355 \times 15 = 5326$ LOC.

O tipo de sistema foi classificado como Sistema Web, considerando sua natureza de aplicação distribuída e interface responsiva. A produtividade média adotada para esse tipo de sistema foi de 3.300 LOC/mês.

Dividindo o total de KLOC pela taxa de produtividade, obteve-se o prazo estimado de desenvolvimento de 1,61 mês, o que corresponde a aproximadamente 35 dias úteis de trabalho.

Para o cálculo do custo total, adotaram-se os parâmetros operacionais definidos pela norma internacional ISO para métricas de esforço, considerando 132 horas de trabalho por mês (22 dias × 6 horas por dia).

O valor hora foi estabelecido em R\$ 18,00, com base em uma estimativa compatível com o custo médio de um desenvolvedor júnior.

A fórmula aplicada para cálculo do investimento total foi, (Custo = Resultado × 132 × Valor da Hora), Substituindo os valores: $1,61 \times 132 \times \text{R\$ } 18,00 = \text{R\$ } 3.834,54$.

Esse valor representa o investimento estimado para a implementação completa do protótipo funcional do sistema Licitasp, incluindo análise, desenvolvimento, testes e validação.

Além das métricas técnicas, o cálculo considerou também: o número de tabelas e colunas previstos no modelo de dados, que influenciam diretamente o número de arquivos e interfaces; a linguagem de programação e *frameworks* utilizados (Python, Flask, Bootstrap e PostgreSQL), que impactam a produtividade média (LOC/FP); o tipo de sistema (web), que define a produtividade base para o cálculo de esforço, e o valor da hora de trabalho adotado como referência para o custo total.

Dessa forma, o valor de R\$ 3.834,54 reflete uma estimativa documentada de esforço e investimento necessários para o desenvolvimento do Licitasp, considerando sua arquitetura tecnológica, o escopo funcional e as métricas reconhecidas internacionalmente de análise de software. O resumo dos cálculos está mostrado no Quadro 8.

Quadro 8 – Métricas

Função	Nº de ocorrência	Complexidade	Peso	Resultado	Nível de Influência do Sistema (0 a 5)	
entradashas	2	Simples	3	6	Comunicação de dados	5
		Médio	4	0	Performance	5
	7	complexo	6	42	Volume de transações	5
						Eficiência do usuário final
saídashas	2	Simples	4	8	Processamento complexo	5
	7	Médio	5	35	Facilidade de implantação	5
	1	complexo	7	7	Múltiplos locais	5
						Processamento distribuído
consultashas	2	Simples	3	6	Utilização de equipamento	5
	1	Médio	4	4	Entrada de dados on-line	5
	7	complexo	6	42	Atualização on-line	5
						Reutilização de código
	9	Simples	7	63	Facilidade operacional	5

Função	Nº de ocorrência	Complexidade	Peso	Re-sultado	Nível de Influência do Sistema (0 a 5)				
arquivos		Médio	10	0	Facilidade de mudanças	5			
		complexo	15	0					
					Total de NI	70			
interfaces	10	Simples	5	50					
		Médio	7	0					
		complexo	10	0					
				Total de FP'b	263	FA = multiplicar o NI pela taxa real =0,65+(0,01*70)			
						1,35			
					FP'r = multiplicar o FP'b pelo FA	355			
					Informe o nº de LOC da Linguagem	15			
Estimativas do número médio de LOC por FP					KLOC = Multiplicar o FP'r pelo tipo de linguagem				
					Total de KLOC	5326			
					Informe o tipo de sistema	3.300			
					PRAZO (dividir o KLOC pelo tipo de sistema)				
					(RESULTADO DA DIVISÃO)QTDD DE MESES	1,61386			
Tipo de Sistema		Produ - Kloc/Loc /mês			QTDD DE DIAS (% x 22)				
Sistema Comercial		2.500			QTDD DE HORAS (% x 6)				
Comércio Eletrônico		3.600			QTDD DE MINUTOS (% x 60)				
Sistema Web		3.300							
CUSTO - INVESTIMENTO									
PRAZO					Informe o valor da hora de trabalho	R\$ 18,00			
DIAS/MÊS		22			ISO (NORMA INTERNACIONAL) HORAS/MÊS =	132			
HORAS/DIA		6			FÓRMULA = 132 * RESULTADO DA DIVISÃO * VALOR DA HORA				
MINUTOS/HORA		60							
CONFIRMAÇÃO (RESULTADO * 132 * Vr DA HORA)					VALOR TOTAL DO PROJETO =	R\$ 3.834,54			

Fonte: os autores

6.3 Proposta Comercial

O modelo de negócio proposto é baseado em *Software as a Service* (SaaS), no qual o usuário paga uma assinatura mensal para receber alertas e relatórios personalizados de licitações. O Quadro 9 apresenta a proposta comercial.

Quadro 9 – Proposta Comercial

Tipo	Descrição
Produto	Sistema web LictaSP – monitoramento e notificação de licitações públicas do DOE-SP.
Público-alvo	Pequenas e médias empresas interessadas em participar de licitações públicas.
Proposta de valor	Automatizar o acompanhamento de oportunidades públicas, reduzindo o tempo de busca e ampliando a participação das PMEs.
Modelo de Receita	Assinatura mensal (plano básico e plano profissional).
Canais de distribuição	Plataforma web, e-mail marketing, redes sociais e parcerias com associações comerciais.
Custo operacional	Hospedagem em nuvem, manutenção técnica, atualização de base de dados e suporte ao cliente.

Fonte: os autores

O planejamento da comercialização, na projeção inicial foi definido como: 1. Plano básico: R\$ 39,90/mês com até 10 palavras-chave, e 1 e-mail de notificação; 2. Plano profissional: R\$ 79,90/mês com palavras-chave ilimitadas, múltiplos destinatários e relatórios em PDF.

O modelo comercial também prevê versões futuras personalizadas para órgãos públicos e consultorias, com integração de múltiplos diários oficiais e alertas em tempo real.

Considerações finais

A partir de uma entrevista inicial com um cliente em potencial, foram levantadas informações reais sobre as dificuldades enfrentadas por empresas no acompanhamento de editais.

Com base nessas informações o projeto da solução e seu desenvolvimento concentraram-se na prototipagem do sistema, resultando em um aplicativo web com interface funcional que guia o usuário pelo fluxo operacional melhorando sua experiência de uso.

O *back-end*, será implementado em versão futura, tendo sido projetado e dimensionado para suportar crescimento e integração futuros, utilizando uma estrutura de classes escalável e modular.

Como desdobramentos futuros, estão previstos: a implementação completa do *back-end* em Python com PostgreSQL; o desenvolvimento de um módulo de

inteligência artificial para leitura automatizada de editais; e a integração do sistema com outros Diários Oficiais estaduais e municipais.

O projeto, portanto, representa um passo inicial sólido rumo à democratização do acesso às oportunidades públicas, oferecendo às PME uma ferramenta prática, acessível e escalável.

Referências

BACURAU, Rodrigo M.; LEAL, Brauliro G.; RAMOS, Ricardo A. **Uma abordagem para a construção de diagramas da UML concomitante à prototipação de interface** [em linha]. set. 2022.

DEVMEDIA. **Contagem de pontos de função**. 2016. Disponível em: <https://www.devmedia.com.br/contagem-de-pontos-de-funcao/34390>. Acesso em: 13 nov. 2025.

DOMATECH. **Requisitos funcionais e não funcionais: entenda a importância**. 2024. Disponível em: <https://www.domatech.com.br/blog/tecnologia/requisitos-funcionais-e-nao-funcionais>. Acesso em: 26 set. 2025.

FIGUEIREDO, Eduardo. **Requisitos funcionais e requisitos não funcionais**. Belo Horizonte: DCC/ICEEx – UFMG, [s.d.]. Disponível em: https://homepages.dcc.ufmg.br/~figueiredo/disciplinas/aulas/req-funcional-rnf_v01.pdf. Acesso em: 9 nov. 2025.

ISMAIL, André Martins; CINTRA, Yara Consuelo. **Utilização do Business Model Canvas como apoio estratégico em uma empresa de confecção digital**. *Revista de Empreendedorismo e Gestão de Micro e Pequenas Empresas*, v. 6, n. 1, p. 40–59, 2021.

JÚNIOR, Edwar Saliba. **Diagrama de classes**. Uberaba (MG): Instituto Federal de Educação, Ciência e Tecnologia do Triângulo Mineiro, jan. 2020. Disponível em: https://esj.eti.br/IFTM/Disciplinas/Grau03/APOO/APOO_Unidade_05_DiagramaDe-Classes.pdf. Acesso em: 9 nov. 2025.

LISBÔA, Maria da Graça Portela; GODOY, Leoni Pentiado. **Aplicação do método 5W2H no processo produtivo do produto: a joia.** *Iberoamerican Journal of Industrial Engineering*, v. 4, n. 7, p. 32–47, 2012.

MESQUITA, Eduardo José Soler; FINGER, Marcelo. **Projeto de dados em bancos de dados distribuídos.** 1998. Tese (Doutorado em Ciência da Computação) – Instituto de Matemática e Estatística, Universidade de São Paulo, 27 mar. 1998.

MOLINA, Sidny de Almeida; ANDRADE, Alexandre Acácio de. **Modelagem de falhas e exceções em processos de negócios em BPMN: um mapeamento sistemático da literatura.** In: *Workshop em Modelagem e Simulação de Sistemas Intensivos em Software (MSSiS)*. SBC, 2024. p. 11–20.

MELLO, Rafael Maiani. **Técnica para Inspeção de Diagramas de Atividades.** 2011. Tese de Doutorado. Universidade Federal do Rio de Janeiro.

SANTOS, Ubirajara Petri; CANAL, Ana Paula. **Estimativas de software utilizando análise de pontos de função.** [s.l.: s.n.], [s.d.]. Disponível em: https://tfgonline.lapinf.ufn.edu.br/media/midias/TFG_II_BIRA_2.07.pdf. Acesso em: 9 nov. 2025.

SCHNEIDER, Geri; WINTERS, Jason P. **Applying use cases: a practical guide.** Pearson Education, 2001.

SPETH, Christophe. **A análise SWOT: uma ferramenta chave para o desenvolvimento de estratégias empresariais.** 50Minutos.es, 2023.

TRINDADE, Gabriela Oliveira; LUCENA, Márcia. **Rastreabilidade de requisitos em metodologias ágeis: um estudo exploratório.** In: *Simpósio Brasileiro de Sistemas de Informação (SBSI)*. SBC, 2016. p. 478–485.

VAROLO, Flávio Augusto de Queiroz; FIIRST, Clóvis. **Utilização do termo de abertura de projetos (TAP) como ferramenta para elaboração de propostas e planos de trabalhos no município de Guaíra–PR.** *Revista Competitividade e Sustentabilidade – ComSus*, v. 7, n. 2, 2020.

Apêndice 1 – Termo de Abertura do Projeto (TAP) para o projeto Licitasp

1. Identificação do Projeto

Nome do Projeto: Licitasp – Sistema Web de Automação para Extração e Notificação de Licitações do DOE-SP

Patrocinador: LicitTech Solutions Ltda. (empresa fictícia criada para fins acadêmicos)

Gerente de Projeto: Rafael Albano da Silva

Data de Abertura: 03/08/2025

2. Justificativa / Contextualização do Projeto

Pequenas e médias empresas (PMEs) enfrentam dificuldade para acompanhar diariamente as publicações de editais e licitações no Imprensa Oficial do Estado de São Paulo (DOE-SP). Isso reduz sua participação em processos de contratação pública. O sistema Licitasp propõe automatizar a raspagem dos editais, filtragem por palavras-chave e notificação imediata aos usuários, aumentando oportunidades e democratizando o acesso.

3. Objetivos de Alto Nível

- Extrair automaticamente as edições do DOE-SP em formato digital.
- Filtrar publicações relevantes para PMEs com base em parâmetros configuráveis (palavras-chave, regiões, setor de atividade).
- Armazenar os resultados em banco de dados para consulta e histórico.
- Notificar empresas interessadas via interface web ou e-mail de forma personalizada.

4. Entregas Principais

- Protótipo funcional do sistema web: interface, *back-end*, banco de dados.
- Módulo de raspagem automatizada do DOE-SP.
- Módulo Web para configurações da busca e lista de licitações.
- Sistema de notificação (web/e-mail).

5. Escopo Preliminar

- Incluído: coleta de publicações do DOE-SP, filtragem por palavras-chave, notificação para usuários cadastrados, interface administrativa.
- Excluído: integração com outras diárias oficiais estaduais fora de São Paulo, app móvel nativo (será considerado em fase futura).

6. Marcos Principais

- M1 – Aprovação do TAP
- M2 – Entrega do protótipo de raspagem
- M3 – Entrega das buscas e notificações
- M4 – Entrega da filtragem configurável
- M5 – Entrega final e homologação

7. Premissas e Restrições

- Premissas: o DOE-SP disponibiliza seus arquivos *online*; infraestrutura de hospedagem disponível.
- Restrições: orçamento limitado; cronograma de 3 meses; equipe restrita.

8. Riscos Iniciais

- Alterações na estrutura de publicação do DOE-SP que comprometem a raspagem.
- Baixa adesão de PMEs durante fase piloto.
- Problemas de desempenho ou escalabilidade.

9. Principais *Stakeholders*

- Patrocinador - Licitatech Solutions Ltda. (empresa fictícia criada para fins acadêmicos)
- Gerente de Projeto – Rafael Albano da Silva
- Usuários (PMEs) – Licitatech Solutions Ltda.

10. Autoridade do Gerente de Projeto

O gerente de projeto está autorizado a alocar os recursos humanos e técnicos aprovados no orçamento, solicitar alterações com impacto em prazo ou custo, e conduzir reuniões de decisão com os *stakeholders* descritos.

11. Aprovação

Assinatura _____

Apêndice 2 – Perguntas e Respostas

1. Hoje vocês consultam apenas o Diário Oficial de SP ou precisam acompanhar outros canais oficiais também?

“Por enquanto só SP, mas temos interesse em acompanhar outros estados e, quem sabe, o Diário Oficial da União.”

Relação com SWOT/5W2H:

Oportunidade: Expandir a cobertura.

What: Suportar múltiplas fontes.

Why: Evitar dependência de uma única fonte.

2. O sistema é voltado apenas para uso interno ou vocês imaginam abrir o acesso para outras empresas do setor?

“Iremos utilizar internamente, mas vimos que outras empresas também sofrem com o mesmo problema. Pode virar um produto SaaS.”

Relação com SWOT/5W2H:

Oportunidade: Ampliar mercado.

What: Expandir público-alvo.

Why: Aumentar base de clientes.

3. Se forem vender para outras empresas como pensam em monetizar o sistema, por exemplo, com planos?

“Sim, queremos algo intuitivo e escalável para pequenas empresas com alertas simples, notificações e relatórios.”

Relação com SWOT/5W2H:

Força: Flexibilidade comercial.

What: Oferecer planos diferenciados.

Why: Atender vários perfis de clientes.

4. Os usuários precisam acessar o sistema pelo celular ou é algo secundário?

“Muitos gestores conferem alertas fora do escritório. Ter uma versão simples no celular seria ótimo. Mas por enquanto iríamos somente enviar as notificações e ter o site web.”

Relação com SWOT/5W2H:

Oportunidade: Expansão para mobile.

What: Criar app via WebView futuramente.

Why: Atender usuários mobile e offline.

5. Quem será responsável por ajustar o código quando o Diário Oficial mudar o formato?

“Quero que o sistema seja modular e na medida do possível ajustável para reaproveitar o máximo de código possível se isso acontecer.”

Relação com SWOT/5W2H:

Fraqueza: Dependência técnica.

How: Automação de monitoramento e logs.

6. O sistema deve permitir reaproveitar funções ou será algo fixo para o Diário Oficial de SP?

“Queremos que seja modular, se amanhã quisermos adicionar outro estado, não precisar reescrever tudo.”

Relação com SWOT/5W2H:

Força: Arquitetura escalável.

How: Uso de injeção de dependências e modularização.

7. O que acontece se o sistema falhar por alguns dias, há algum plano de contingência?

“Não seria o fim do mundo, mas preferimos que o sistema avise se parar de funcionar. Podemos remediar se não for muito demorado a manutenção.”

Relação com SWOT/5W2H:

Fraqueza: Falta de monitoramento.

How: Criar alertas automáticos em caso de falha.

8. Há dados confidenciais ou sensíveis nos registros do Diário Oficial?

“Não, são informações públicas. Mas o histórico de pesquisa interna pode ser sensível, então queremos controle de acesso por usuários.”

Relação com SWOT/5W2H:

Ameaça: Exposição de dados internos.

How: Controle de permissões e autenticação.

9. O que é essencial para o sistema e o que pode ficar para depois?

“Essencial seria captura, filtro e notificação inicialmente. Classificação automática, dashboards e alertas direcionados podem vir depois.”

Relação com SWOT/5W2H:

Força: Foco no MVP.

When: Após lançamento web.

10. Como os usuários devem ser notificados das publicações *e-mail, WhatsApp*?

“E-mail resolve por enquanto. Vamos sentir como vai ser, e se necessário vamos pagar para utilizar o *WhatsApp*.”

Relação com SWOT/5W2H:

Oportunidade: Expandir comunicação no futuro.

How: Notificações simples no MVP.

11. Vocês precisam guardar todas as versões das publicações ou apenas o registro mais recente?

“Inicialmente apenas as licitações diárias, e o envio do relatório por *e-mail*. Mas se der para armazenar para consultas futuras seria bom.”

Relação com SWOT/5W2H:

Fraqueza: Armazenamento excessivo.

How: Armazenar apenas metadados.

12. Vocês pensam em integrar o sistema com ferramentas de compliance ou gestão regulatória?

“Não tem necessidade, é mais busca de informações públicas.”

Relação com SWOT/5W2H:

Oportunidade: Integrações estratégicas.

When: Após validação inicial.

13. O que tornaria o sistema de vocês melhor que outras soluções de monitoramento existentes?

“Não conhecemos soluções para esse problema, talvez existam, mas não temos conhecimento.”

Relação com SWOT/5W2H:

Força: Conhecimento setorial.

Why: Valor agregado ao cliente final.