

LEGIS UP

Protótipo de Sistema de Automação para o Poder Legislativo Municipal

Adilson S. S. Moraes¹, Derick S. Conceição¹, Leon A. Q. Bentes¹,
Márcio A. B. Moda¹, Neuri M. F. Gato¹, Yuri R. Martins¹

¹Curso Superior em Análise e Desenvolvimento de Sistemas – Instituto Federal do Pará (IFPA)
CEP 68250-000 – Óbidos – PA – Brasil

{angel}@ifpa.edu.br

Abstract. *The present study aims to understand and develop a software prototype for automating and streamlining bureaucratic processes, including the management of legislative projects, agenda control, document management, transparency, and access to information, attendance and voting control, parliamentary committees, notifications, and alerts. This study aims to analyze and define the needs and characteristics of a software prototype project, providing a comprehensive overview of its functionalities and a technological solution aimed at automating legislative processes. Through this solution, it is believed that it will replace the physical archiving of documents, transitioning to a digital environment similar to the processes of the Chamber of Deputies and the Federal Senate. This proposal focuses on the development of an intuitive, comprehensive, and multi-platform software prototype, modernizing parliamentary activities and enhancing the efficiency and transparency of the process.*

Resumo. *O presente estudo busca entender e desenvolver um protótipo de software para automatização e agilização dos processos burocráticos dos parlamentos municipais, incluindo gerenciamento de projetos de lei, controle de pauta, gestão de documentos, transparência e acesso à informação, controle de presença e votação, comissões parlamentares, notificações e alertas. Este estudo tem como objetivo analisar e definir as necessidades e características de um projeto de protótipo de software, fornecendo uma visão abrangente das funcionalidades presentes e uma solução tecnológica visando a automatização nos processos legislativos. Através dessa solução acredita-se que, substituirá o arquivamento físico de documentos, proporcionando uma transição para um ambiente digital similar aos processos da Câmara dos Deputados e do Senado Federal, sendo essa proposta no desenvolvimento de um protótipo de software intuitivo, abrangente e multiplataforma, modernizando as atividades parlamentares e melhorando a eficiência e transparência do processo.*

1. Introdução

A palavra "parlamento" tem origem no grego antigo. Ela deriva do termo grego "παράλημπειν" (paralēmpein), que significa "falar ao lado de" ou "discutir". O prefixo "para-" indica "ao lado de", enquanto "lēmpein" se refere a "falar". Essa raiz etimológica reflete a natureza essencial de um parlamento como um local onde os representantes do

povo se reúnem para discutir e deliberar questões de interesse público O conceito de parlamento, como o entendemos hoje, tem suas origens na Grécia antiga, onde a democracia direta era praticada em cidades-estados como Atenas (A. R. Sathler e M. R. Sathler 2016). Os cidadãos se reuniam em assembleias para discutir e votar em questões políticas. Essa tradição de participação democrática influenciou o desenvolvimento das instituições parlamentares ao longo da história, inclusive na Europa medieval e moderna. Com o advento das revoluções industriais, a globalização tem causado desafios e impactos nos serviços públicos em todo o mundo, mais especificamente nas Câmaras Municipais (Eirão e Sow 2013).

O Poder Legislativo tem como função central a elaboração das leis, ao lado de exercer outras tarefas constitucionais como a apresentação pública de assuntos de interesse dos cidadãos, o debate sobre tais reivindicações de modo a agrega-las sob o interesse geral e a fiscalização política dos atos do executivo. Na lista de competências do Poder Legislativo Municipal, enumeradas pela Constituição, a principal é a de fazer, suspender, interpretar e revogar as leis de competência do Município. Outras funções é de fiscalizar e controlar os atos do Poder Executivo; funções administrativas internas de organização de seus serviços e uma função política adicional: a de representar o povo em suas queixas e reivindicações, operando como uma ouvidoria geral da sociedade.

Chama-se de Processo Legislativo todo conjunto de atos realizados pelos órgãos do Poder Legislativo, de acordo com regras previamente fixadas, para elaborar normas jurídicas (emendas à Constituição, leis complementares, leis ordinárias e outros tipos normativos dispostos no art. 59 da Constituição Federal).

“Processo legislativo é o conjunto de ações realizadas pelos órgãos do poder legislativo com o objetivo de proceder à elaboração das leis sejam elas constitucionais complementares e ordinárias bem como as resoluções e decretos legislativos”. (BRASIL, 2011).

Nesse contexto, atualmente muitos Poderes Legislativos ainda não usufruem de um sistema informatizado de processo legislativo, seja para acompanhamento de pautas, tramitações de projetos de leis, acesso às leis, que ofereça aos cidadãos a transparência dos atos legislativos, assim como a serviços que requerem celeridade, redução de custos, organização e clareza de informações.

A automatização dos processos legislativos por meio de um protótipo de software representaria um avanço significativo na modernização e na eficiência do Poder Legislativo. Uma solução tecnológica que atenderia às necessidades específicas do contexto legislativo, proporcionando uma gestão mais eficiente e transparente das atividades legislativas.

2. Objetivos

O presente estudo tem como objetivo analisar e definir as necessidades e características de um projeto de protótipo de software, fornecendo uma visão abrangente das funcionalidades presentes e uma solução tecnológica visando a automatização nos processos legislativos.

2.1. Objetivos Específicos

- Analisar os processos legislativos existentes, identificando pontos de melhoria e oportunidades de automatização.
- Definir os requisitos funcionais e não funcionais do software, levando em consideração as necessidades específicas do contexto legislativo.
- Proporcionar uma visão abrangente das funcionalidades necessárias para o protótipo de software, incluindo recursos de gestão documental, controle de processos, comunicação interna e externa, entre outros.
- Desenvolver uma solução tecnológica que atenda aos requisitos levantados, utilizando as melhores práticas de desenvolvimento de software e considerando a escalabilidade e a segurança da aplicação.
- Realizar testes e validações do protótipo de software, garantindo sua eficácia e adequação aos processos legislativos.

3. Fundamentação Teórica

De acordo com a Constituição Federal (*Constituição da República Federativa do Brasil de 1988 - Capítulo IV dos Municípios. Art. 29 s.d.; Constituição da República Federativa do Brasil de 1988 - Capítulo IV dos Municípios. Art. 30 s.d.; Constituição da República Federativa do Brasil de 1988 - Capítulo IV dos Municípios. Art. 31 s.d.*), o poder legislativo municipal é o órgão público encarregado de legislar a gestão das cidades, ou seja, elaborar, discutir, aprovar e modificar as leis que regem os municípios. É uma das partes integrantes da estrutura administrativa de uma cidade e desempenha um papel crucial na representação dos interesses da população local. As principais atribuições desse poder incluem: elaborar e aprovar leis municipais: os vereadores, que compõem o poder legislativo municipal, têm a função de criar leis que regulamentem assuntos de interesse local, como questões tributárias, urbanísticas, de saúde, educação, transporte, entre outros; fiscalizar o executivo: cabe à câmara municipal fiscalizar as ações do poder executivo, verificando a aplicação dos recursos públicos, a execução de políticas municipais e o cumprimento das leis vigentes; representar os cidadãos: os vereadores são eleitos pelo voto popular para representar os interesses dos cidadãos perante o governo municipal, ou seja, são os representantes do povo na esfera legislativa local; aprovar o orçamento municipal: é de competência da câmara municipal aprovar o orçamento anual do município, estabelecendo as despesas e receitas previstas para o exercício financeiro; promover audiências públicas e debates: a câmara municipal pode promover audiências públicas e debates para discutir temas de relevância para a comunidade, dando voz aos cidadãos e estimulando a participação popular na vida política local.

Em conformidade com a Lei Nº 8.159 de 8 de janeiro de 1991 (Brasil 1991), que dispõe sobre a política nacional de arquivos públicos e privados no Brasil. Essa lei é fundamental para a preservação e o acesso à informação, estabelecendo diretrizes para a gestão, organização, preservação e acesso aos documentos de interesse público. Alguns dos principais pontos desta lei são: Definição do arquivo, acesso à informação, responsabilidade dos órgãos públicos, preservação do patrimônio documental e padronização de procedimentos.

Em virtude da Lei Nº 13.709/18 de 14 de agosto de 2018 (Brasil 2018), lei geral de proteção de dados pessoais (LGPD) do Brasil, que estabelece regras e diretrizes

para o tratamento de dados pessoais por organizações públicas e privadas, com o objetivo de proteger a privacidade e os direitos dos titulares dos dados. Seus principais pontos de regulamentação estão: definição de dados pessoais; princípios para estabelecer o tratamento de dados como finalidade, adequação, necessidade, livre acesso, qualidade dos dados, transparência, segurança, prevenção, não discriminação e responsabilização e prestação de contas; consentimento; direito dos titulares; responsabilidade e segurança; transferência internacional de dados e sanções. Uma informação importante sobre esta lei é que ela busca alinhar o Brasil com padrões internacionais de proteção de dados, como o Regulamento Geral de Proteção de Dados (GDPR) da União Europeia, visando garantir a privacidade e a segurança das informações pessoais dos cidadãos brasileiros.

Em conformidade com a Lei Orgânica do Município de Óbidos, no estado do Pará (**lei-obidos**), serão considerados, especialmente aqueles que versam sobre a criação e o funcionamento da Câmara Municipal. A Lei Orgânica é o documento que estabelece as normas fundamentais para a organização e o funcionamento do município, incluindo as competências dos poderes legislativo, executivo e judiciário, bem como os direitos e deveres dos cidadãos.

Dentre os trechos relevantes da Lei Orgânica de Óbidos, serão abordados aqueles que tratam das atribuições da Câmara Municipal, suas competências, a composição, eleição e funcionamento dos vereadores, às sessões legislativas, o processo legislativo municipal, entre outros aspectos relacionados ao papel e às atividades da instituição legislativa no âmbito municipal.

Para embasamento teórico de modelagem de sistemas, este estudo baseou-se na obra de XEXÉO (**de2002modelagem**) para a construção deste protótipo, no que diz a respeito na coleta de requisitos, pois segundo o autor a coleta de requisitos é uma etapa fundamental no processo de desenvolvimento de sistemas, desempenhando um papel crucial na definição do escopo e na garantia de que o produto final atenda às necessidades dos usuários e às expectativas do cliente. Neste contexto, o trabalho de XEXÉO, em seu estudo de 2007, fornece insights valiosos sobre as melhores práticas e abordagens para a coleta de requisitos visando a modelagem de sistemas eficazes.

No que se refere aos diagramas de casos de uso, diagramas de atividades, diagrama de classes, modelagem de banco de dados, identificação de tecnologias e o protótipo, o autor XEXÉO em sua supra obra "Modelagem de Sistemas" de 2007, estabelece que essas etapas são meramente técnicas de documentação e modelagem de um sistema.

XEXÉO destaca a importância de um processo estruturado e iterativo para a coleta de requisitos, no qual os analistas de sistemas interagem de forma colaborativa com os stakeholders para identificar, documentar e validar as necessidades do sistema. Essa abordagem baseia-se na premissa de que a compreensão profunda do contexto e dos objetivos do projeto é essencial para o sucesso do empreendimento.

Um aspecto fundamental abordado por XEXÉO (2007) é a necessidade de utilizar uma variedade de técnicas e ferramentas durante a coleta de requisitos, de modo a capturar informações de maneira abrangente e precisa. Isso pode incluir entrevistas com os usuários finais, observação direta das operações em campo, análise de documentos existentes e realização de workshops de brainstorming.

Além disso, XEXÉO (2007) destaca a importância de garantir a participação ativa

e contínua dos stakeholders ao longo do processo de coleta de requisitos. Isso não apenas ajuda a garantir que todas as perspectivas sejam consideradas, mas também aumenta o engajamento e o comprometimento das partes interessadas com o projeto.

Outro aspecto relevante discutido por XEXÉO (2007) é a necessidade de documentar claramente os requisitos identificados, utilizando uma linguagem clara e concisa e adotando padrões de documentação reconhecidos pela indústria. Isso ajuda a evitar mal-entendidos e ambiguidades durante o desenvolvimento do sistema, garantindo que todos os envolvidos tenham uma compreensão comum dos requisitos.

Em resumo, o estudo de XEXÉO (2007) sobre coleta de requisitos para modelagem de sistemas destaca a importância de um processo estruturado, colaborativo e iterativo, no qual a compreensão profunda das necessidades do usuário é fundamental. Ao adotar as práticas e abordagens recomendadas por XEXÉO, as organizações podem aumentar a probabilidade de sucesso de seus projetos de desenvolvimento de sistemas, entregando soluções que atendam verdadeiramente às necessidades e expectativas de seus usuários.

A Unified Modeling Language (UML) e a modelagem de dados são duas ferramentas essenciais no desenvolvimento de sistemas de software. Neste sentido, utilizaremos a abordagem de XEXÉO, conforme descrito em sua obra de 2007, para modelar a construção em base dessas técnicas na engenharia de software.

Para XEXÉO (2007), a UML é uma linguagem padrão para modelar sistemas de software. Ela oferece uma variedade de diagramas que permitem aos desenvolvedores visualizar, especificar, construir e documentar os aspectos de um sistema. Os principais diagramas da UML incluem diagramas de casos de uso, diagramas de classes, diagramas de sequência e diagramas de atividades. Diagramas de Casos de Uso: Permitem identificar os diferentes casos de uso de um sistema, descrevendo as interações entre os usuários e o sistema. E entre estão: Diagramas de Classes e Diagramas de Atividades.

De acordo com o BORGES (de2002modelagem), a modelagem de dados é o processo de criar um modelo conceitual que descreve a estrutura e o comportamento dos dados em um sistema de informação. Ela envolve a identificação das entidades de dados, seus atributos e os relacionamentos entre elas. Os principais modelos de dados incluem o modelo conceitual, o modelo lógico e o modelo físico. Modelo Conceitual: Descreve as entidades de negócio e os relacionamentos entre elas, sem se preocupar com detalhes de implementação. O Modelo Lógico é o modelo conceitual em estruturas de dados específicas de um sistema de gerenciamento de banco de dados, como tabelas, chaves primárias e estrangeiras. Enquanto o Modelo Físico reflete a implementação real do modelo lógico em um banco de dados específico, incluindo detalhes de armazenamento, índices e restrições.

4. Metodologia

4.1. Coleta de Requisitos e a Modelagem de Software

Na primeira etapa deste estudo foi na coleta de requisitos, sendo esta etapa indispensável neste projeto de desenvolvimento e modelagem de sistemas de acordo com o XEXÉO (2007). A coleta de requisitos partiu-se do pressuposto no relato de um funcionário da

Câmara Municipal de Óbidos e com os materiais disponíveis acerca do funcionamento de um parlamento municipal.

Portanto, é importante considerar que a modelagem de sistemas foi estabelecida como um critério fundamental para a definição dos requisitos funcionais no escopo do presente estudo.

Requisitos Funcionais

- **Gerenciamento de Projetos de Lei:** O sistema deve permitir o cadastro, acompanhamento e atualização de projetos de lei, incluindo informações sobre sua tramitação, autores e status atual.
- **Gestão de Documentos:** O sistema deve oferecer recursos para o armazenamento, organização e recuperação de documentos relacionados às atividades parlamentares, garantindo fácil acesso e controle de versões.
- **Transparência e Acesso à Informação:** Deve ser possível disponibilizar informações sobre as atividades parlamentares de forma transparente e acessível ao público, incluindo a publicação de pautas, atas de sessões, projetos de lei em tramitação, entre outros.
- **Controle de Presença e Votação:** Deve haver funcionalidades para registrar a presença dos parlamentares nas sessões legislativas, bem como para registrar e contabilizar votos durante as votações.
- **Notificações e Alertas:** Deve ser possível configurar notificações e alertas automáticos para manter os usuários atualizados sobre eventos importantes, como novos projetos de lei, alterações na pauta ou reuniões de comissões.

Após a etapa crucial de coleta de requisitos em um projeto de desenvolvimento de software, é fundamental seguir para o próximo passo para garantir que os requisitos levantados anteriormente sejam devidamente compreendidos, validados e traduzidos em uma solução eficaz. Portanto, mencionamos a técnica da engenharia de software a linguagem UML, a abaixo seguirá os diagramas de casos de uso e o diagrama de atividades. Um diagrama que permite identificar os diferentes casos de uso de um sistema, descrevendo as interações entre os usuários e o sistema. Enquanto o outro diagrama representa o fluxo de controle entre as atividades do sistema, mostrando as ações e decisões que ocorrem em um processo.

4.2. Diagrama de Casos de Uso

O diagrama de casos de uso é uma técnica de modelagem amplamente utilizada para capturar os requisitos funcionais de um sistema. Ele fornece uma visão geral das funcionalidades que o sistema deve fornecer e das interações entre os usuários e o sistema. O principal objetivo do diagrama de casos de uso é identificar os diferentes cenários de uso do sistema e documentar os requisitos de forma clara e concisa.

Um diagrama de casos de uso é composto por três elementos principais: atores, casos de uso e relações entre eles. Seguir a Figura 1 referente ao diagrama de casos de uso do protótipo:

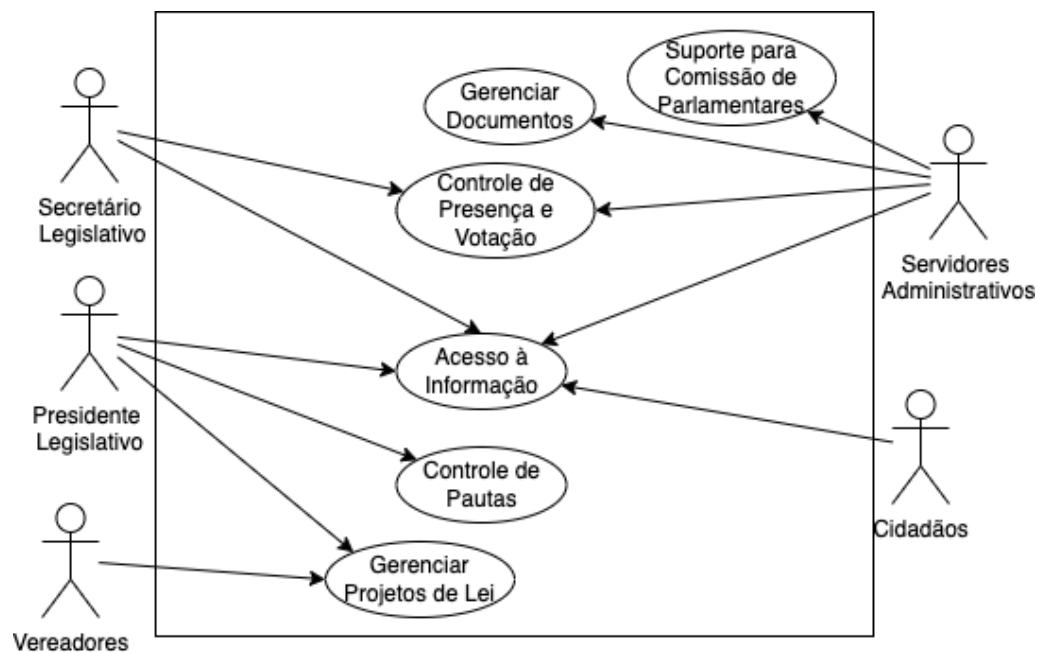


Figura 1. Diagrama de casos de uso

Fonte: Elaborada pelos autores

4.3. Diagrama de Atividades

O diagrama de atividades é uma técnica da UML e é utilizada para modelar o comportamento dinâmico de um sistema. Ele descreve o fluxo de controle de atividades, mostrando a sequência de ações que ocorrem em um processo ou sistema.

Este tipo de diagrama é composto por nós e arestas, representando atividades e transições entre elas, respectivamente. As atividades podem ser simples, como executar uma ação, ou compostas, consistindo em subatividades ou loops. Já as transições indicam o fluxo de controle entre as atividades.

Conforme na Figura 2:

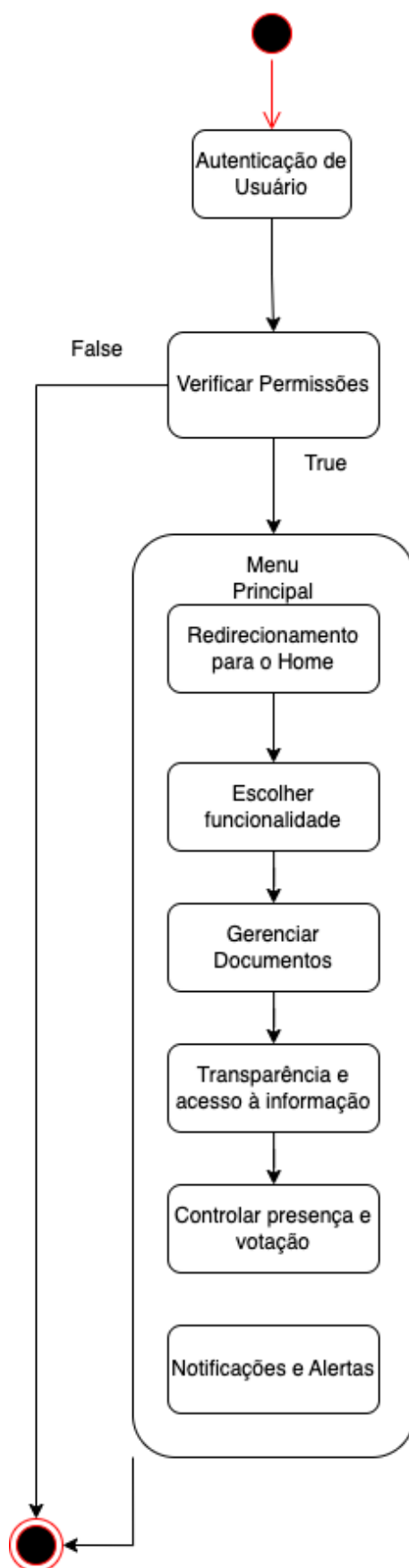


Figura 2. Diagrama de Atividades

Fonte: Elaborada pelos autores

4.4. Diagrama de Classes

Os diagramas de classes são uma representação visual das classes, interfaces e seus relacionamentos dentro de um sistema. Eles são uma ferramenta fundamental na modelagem de sistemas orientados a objetos, permitindo aos desenvolvedores e analistas de sistemas entender a estrutura e a organização do software que estão projetando ou mantendo.

Esses diagramas são compostos por caixas retangulares que representam as classes, onde são especificados os atributos e métodos de cada classe. As linhas conectam essas caixas para mostrar os relacionamentos entre as classes, como associação, agregação, composição e herança.

Os diagramas de classes fornecem uma visão abstrata e simplificada do sistema, permitindo aos desenvolvedores compreender a estrutura do software e identificar possíveis problemas de design. Eles também facilitam a comunicação entre os membros da equipe de desenvolvimento, pois fornecem uma representação visual clara do sistema e de suas interações conforme a Figura 3 a seguir:

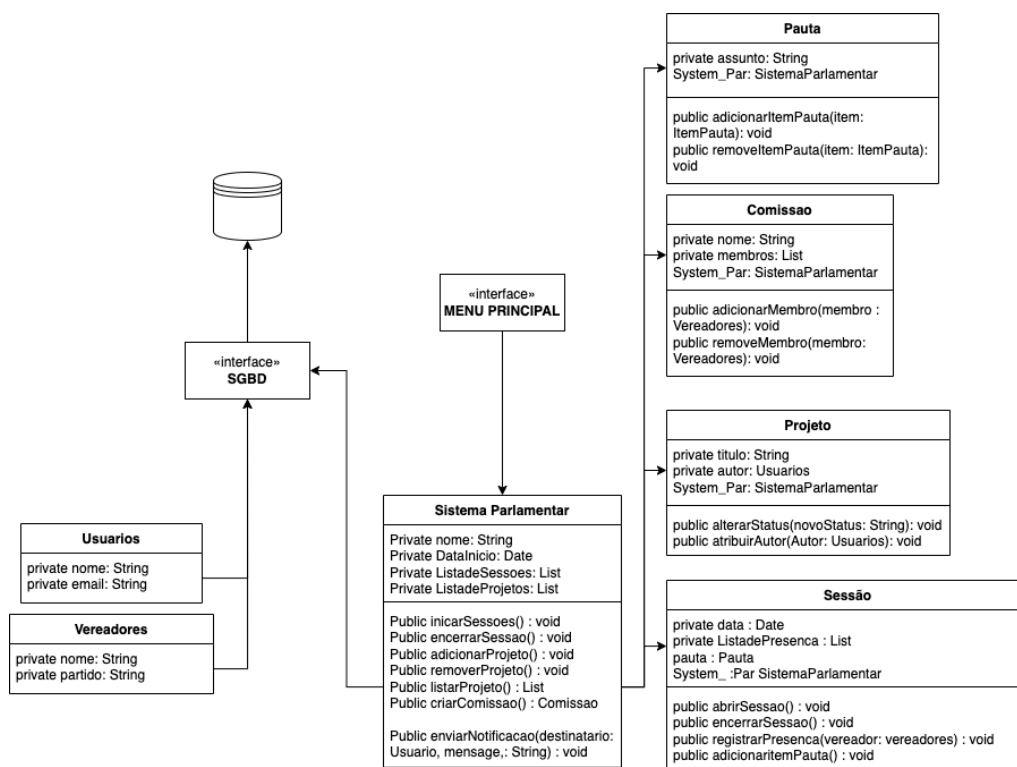


Figura 3. Diagrama de Classes

Fonte: Elaborada pelos autores

Além disso, os diagramas de classes são uma parte essencial da documentação do sistema, pois descrevem a estrutura e o comportamento do software de forma concisa e compreensível. Eles são amplamente utilizados em todas as fases do ciclo de vida do desenvolvimento de software, desde a análise e projeto até a implementação e manutenção.

4.5. Modelagem de Banco de Dados

Paralelamente à elaboração dos diagramas UML e coleta de requisitos, é importante desenvolver outra parte da modelagem do sistema: o modelo de dados do sistema. Isso envolve a identificação das entidades de dados, seus atributos e os relacionamentos entre elas. A modelagem de dados fornece uma base sólida para o design do banco de dados e garante a integridade e consistência dos dados no sistema.

O modelo conceitual de dados é uma representação abstrata e independente de qualquer tecnologia específica que descreve as entidades e os relacionamentos de um sistema de informação. Ele é projetado para capturar as principais informações e regras de negócio de uma organização, proporcionando uma visão clara e organizada dos dados que serão armazenados e manipulados pelo sistema. Este modelo descreve as entidades de negócio, que são os objetos ou conceitos relevantes para a organização, e os relacionamentos entre essas entidades, que mostram como elas estão conectadas umas às outras. Além disso, o modelo conceitual pode incluir atributos que descrevem as características ou propriedades das entidades.

O principal objetivo do modelo conceitual de dados é fornecer uma base sólida para o projeto de banco de dados e sistemas de informação, permitindo que os analistas e desenvolvedores compreendam os requisitos de negócio e as necessidades de informação da organização. Ele serve como um ponto de partida para o design detalhado do banco de dados, ajudando a garantir que o sistema atenda adequadamente às necessidades do usuário final, em alusão à Figura 4.

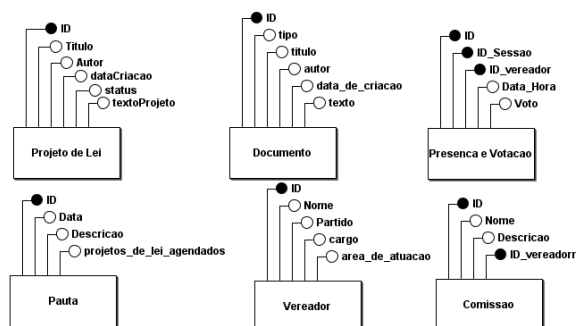


Figura 4. Modelo Conceitual
Elaborada pelos Autores

Enquanto o modelo físico de dados é uma representação concreta e detalhada das estruturas de armazenamento e organização dos dados em um sistema de informação. Ao contrário do modelo conceitual, que é independente de qualquer tecnologia específica, o modelo físico descreve como os dados serão realmente armazenados e acessados em um banco de dados, levando em consideração aspectos como o tipo de banco de dados, o sistema de gerenciamento de banco de dados (SGBD) e os requisitos de desempenho. Nesse modelo, as entidades de negócio são traduzidas em tabelas, os relacionamentos são representados por chaves estrangeiras e os atributos são mapeados para os campos das tabelas. Além disso, são definidos índices, restrições de integridade, partições e outras estruturas específicas do banco de dados para otimizar o desempenho e garantir a consistência e integridade dos dados.

O modelo físico de dados é uma etapa crucial no processo de desenvolvimento de sistemas de informação, pois fornece a base concreta para a implementação do banco de dados e do sistema como um todo. Ele é elaborado com base no modelo conceitual de dados, levando em consideração as necessidades do usuário final, os requisitos de negócio e as características do ambiente de implantação. A elaboração do modelo físico envolve a definição detalhada das tabelas, campos, índices e outras estruturas de armazenamento, bem como a otimização do esquema para garantir o desempenho e a escalabilidade do sistema. É importante ressaltar que o modelo físico pode variar dependendo do tipo de banco de dados e das tecnologias utilizadas, sendo adaptado para atender aos requisitos específicos do projeto.

```

CREATE TABLE ProjetoDeLei (
    projeto_id INT AUTO_INCREMENT PRIMARY KEY,
    titulo VARCHAR(255),
    autor VARCHAR(100),
    data_criacao DATE,
    status ENUM('Em andamento', 'Aprovado', 'Rejeitado'),
    texto LONGTEXT
);

-- Tabela Pauta
CREATE TABLE Pauta (
    pauta_id INT AUTO_INCREMENT PRIMARY KEY,
    data DATE,
    descricao VARCHAR(255)
);

-- Tabela Projeto de Lei na Pauta (relacionamento muitos para muitos)
CREATE TABLE ProjetoDeLei_Pauta (
    projeto_id INT,
    pauta_id INT,
    PRIMARY KEY (projeto_id, pauta_id),
    FOREIGN KEY (projeto_id) REFERENCES ProjetoDeLei(projeto_id),
    FOREIGN KEY (pauta_id) REFERENCES Pauta(pauta_id)
);

-- Tabela Documento
CREATE TABLE Documento (
    documento_id INT AUTO_INCREMENT PRIMARY KEY,
    tipo VARCHAR(100),
    titulo VARCHAR(255),
    autor VARCHAR(100),
    data_criacao DATE,
    texto LONGTEXT
);

-- Tabela Vereador
CREATE TABLE Vereador (
    vereador_id INT AUTO_INCREMENT PRIMARY KEY,
    nome VARCHAR(100),
    partido VARCHAR(50),
    cargo VARCHAR(50),
    area_atuacao VARCHAR(255)
);

-- Tabela Presença e Votação
CREATE TABLE PresencaVotacao (
    presenca_votacao_id INT AUTO_INCREMENT PRIMARY KEY,
    sessao_legislativa_id INT,
    vereador_id INT,
    data_hora DATETIME,
    voto ENUM('A favor', 'Contra', 'Abstenção'),
    FOREIGN KEY (sessao_legislativa_id) REFERENCES SessaoLegislativa(sessao_id),
    FOREIGN KEY (vereador_id) REFERENCES Vereador(vereador_id)
);

-- Tabela Comissao Parlamentar
CREATE TABLE ComissaoParlamentar (
    comissao_id INT AUTO_INCREMENT PRIMARY KEY,
    nome VARCHAR(255),
    descricao VARCHAR(500)
);

-- Tabela Vereador_ComissaoParlamentar (relacionamento muitos para muitos)
CREATE TABLE Vereador_ComissaoParlamentar (
    vereador_id INT,
    comissao_id INT,
    PRIMARY KEY (vereador_id, comissao_id),
    FOREIGN KEY (vereador_id) REFERENCES Vereador(vereador_id),
    FOREIGN KEY (comissao_id) REFERENCES ComissaoParlamentar(comissao_id)
);

-- Tabela Notificacao
CREATE TABLE Notificacao (
    notificacao_id INT AUTO_INCREMENT PRIMARY KEY,
    destinatario VARCHAR(100),
    assunto VARCHAR(255),
    mensagem TEXT,
    data_hora DATETIME
);

```

Figura 5. Modelo Físico (MySQL)
Elaborada pelos Autores

4.6. Protótipo do Legis Up

As tecnologias escolhidas pelos autores, Java, MySQL e Ubuntu Server, desempenham papéis fundamentais na arquitetura e no funcionamento do sistema proposto. O Java, uma linguagem de programação versátil e amplamente adotada, oferece a flexibilidade necessária para desenvolver aplicativos robustos e escaláveis. Sua capacidade de ser exe-

cutado em diversas plataformas, juntamente com sua vasta biblioteca de recursos, o torna uma escolha ideal para o desenvolvimento de sistemas complexos.

O MySQL, por sua vez, é um sistema de gerenciamento de banco de dados relacional confiável e de alto desempenho. Sua estrutura escalável e capacidade de lidar com grandes volumes de dados tornam-no uma escolha popular entre os desenvolvedores para armazenar e gerenciar informações críticas do sistema. Além disso, sua compatibilidade com o Ubuntu Server facilita a integração e implantação do banco de dados no ambiente do servidor.

O Ubuntu Server, uma distribuição Linux amplamente reconhecida, é a escolha ideal para hospedar e executar aplicativos e serviços em ambientes de servidor. Sua segurança, estabilidade e facilidade de uso o tornam uma opção confiável para suportar as demandas de um sistema crítico como o proposto neste trabalho. Além disso, sua ampla comunidade de usuários e recursos de suporte profissional garantem um ambiente de desenvolvimento e operação estável e confiável.

Para incorporar essas tecnologias ao sistema proposto, os autores planejam seguir um processo de desenvolvimento passo a passo. Inicialmente, eles configurarão um ambiente de desenvolvimento usando o Ubuntu Server, instalando e configurando o Java Development Kit (JDK) e o MySQL Server. Em seguida, eles desenvolverão o código-fonte do aplicativo usando a linguagem Java, seguindo as melhores práticas de desenvolvimento de software e design orientado a objetos.

Após o desenvolvimento do aplicativo, os autores planejam integrá-lo ao banco de dados MySQL, definindo o esquema de banco de dados e estabelecendo conexões JDBC para acessar e manipular os dados. Os autores realizarão testes extensivos para garantir que o sistema funcione conforme o esperado e atenda aos requisitos de desempenho, segurança e confiabilidade. Finalmente, eles implantarão o sistema em um ambiente de produção, configurando o servidor Ubuntu para hospedar o aplicativo e o banco de dados MySQL, garantindo sua disponibilidade e escalabilidade.

Portanto, as tecnologias Java, MySQL e Ubuntu Server desempenham papéis essenciais no desenvolvimento e na operação do sistema proposto, oferecendo as ferramentas e recursos necessários para criar um aplicativo eficiente, seguro e escalável. Com um planejamento cuidadoso e uma abordagem metodológica, os autores estão confiantes de que podem desenvolver e implantar com sucesso o sistema, atendendo às necessidades e expectativas dos usuários finais.

4.7. Protótipo do Legis UP

Seguindo a linha de raciocínio de XEXÉO (2007), a prototipagem é um processo essencial no desenvolvimento de software, pois permite aos desenvolvedores e stakeholders visualizarem e testarem uma versão inicial do sistema antes de sua implementação completa. Este tópico abrange os diferentes tipos de prototipagem, como protótipos de baixa e alta fidelidade, bem como suas vantagens e desvantagens.

Os protótipos de baixa fidelidade são esboços simples e rápidos que representam as funcionalidades principais do sistema. Eles geralmente são criados usando papel e lápis ou ferramentas de prototipagem rápida, como o Balsamiq. Embora esses protótipos não sejam muito detalhados, eles são úteis para capturar ideias iniciais, explorar fluxos de tra-

balho e validar conceitos com os stakeholders. No entanto, eles podem não fornecer uma representação precisa do produto final e podem ser limitados em termos de interatividade.

Por outro lado, os protótipos de alta fidelidade são versões mais elaboradas e detalhadas do sistema, que se aproximam mais do produto final em termos de aparência e funcionalidade. Eles são criados usando ferramentas de design e desenvolvimento, como o Figma, e podem incluir elementos interativos, como botões clicáveis e campos de entrada. Esses protótipos são ideais para testar a usabilidade do sistema, obter feedback detalhado dos usuários e validar o design visual. No entanto, eles exigem mais tempo e recursos para serem criados.

Independentemente do tipo de protótipo utilizado, a prototipagem oferece uma série de benefícios. Em primeiro lugar, ela permite uma comunicação mais eficaz entre os membros da equipe e os stakeholders, pois fornece uma representação tangível do sistema em desenvolvimento. Além disso, a prototipagem ajuda a identificar e corrigir problemas de usabilidade e design precocemente no processo de desenvolvimento, economizando tempo e recursos no longo prazo. Por fim, os protótipos podem ser usados para obter feedback dos usuários finais e iterar rapidamente sobre o design do sistema, garantindo que ele atenda às suas necessidades e expectativas.



Figura 6. Página Inicial

Fontes: Elaborada pelos Autores



[HOME](#)
[PROTOCOLO](#)
[LEGISLAÇÃO](#)
[SISTEMA DE VOTAÇÃO](#)
[TRAMITES LEGAIS](#)



Cartas

Documentos Legislativos

Projetos de Leis

Ofícios

Convites

Pesquisar

Destinatário	Autoria	Data	Compartilhar	Download

Figura 7. Página de Protocolos

Fontes: Elaborada pelos Autores



[HOME](#)
[PROTOCOLO](#)
[LEGISLAÇÃO](#)
[SISTEMA DE VOTAÇÃO](#)
[TRAMITES LEGAIS](#)



Leis Municipais

Lei Organica

Leis Antigas

Modificadas recentemente

Pesquisar

Nº da Lei	Título	Autor	Status	Download

Figura 8. Página das Leis

Fontes: Elaborada pelos Autores



Figura 9. Página Legislativa

Fontes: Elaborada pelos Autores

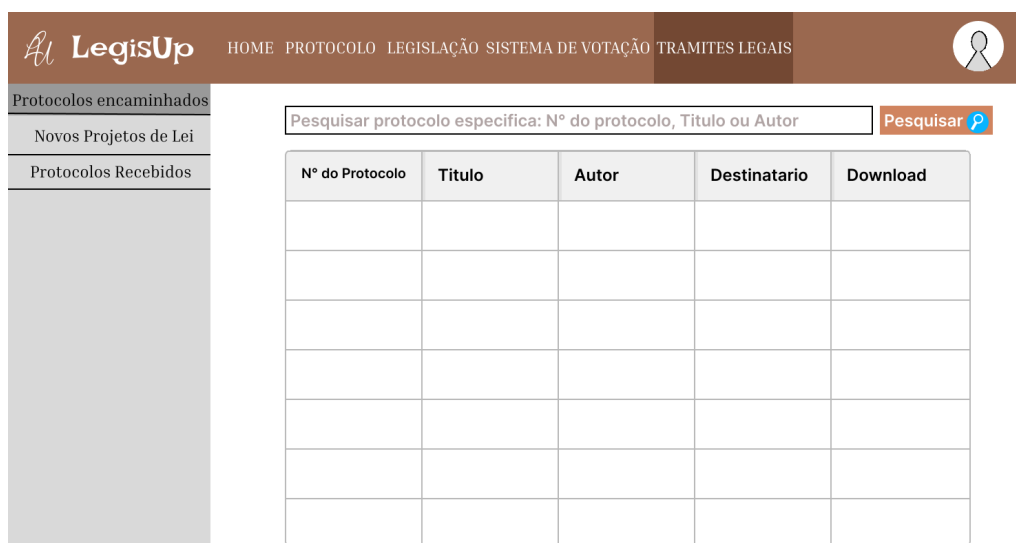


Figura 10. Página dos Trâmites Legais

Fontes: Elaborada pelos Autores

No entanto, a prototipagem também apresenta algumas desvantagens. Protótipos de baixa fidelidade podem não fornecer uma representação precisa do produto final, levando a expectativas equivocadas por parte dos stakeholders. Além disso, protótipos de alta fidelidade podem ser demorados e custosos para serem desenvolvidos, especialmente se forem necessárias várias iterações de design. Portanto, é importante equilibrar os benefícios da prototipagem com as restrições de tempo e recursos do projeto.

5. Resultados Esperados

Esperamos que o presente protótipo de software funcional, que automatize os processos legislativos de forma eficiente e transparente. Pois dessa forma a redução do tempo e dos

recursos necessários para a realização das atividades legislativas.

Contamos que através dessa intervenção tecnológica apresente alguma melhoria na comunicação interna e externa, facilitando o acesso à informação e promovendo a participação cidadã.

Assim como ocorre na Câmara dos Deputados, através deste protótipo poderá ser possível ter o maior controle e rastreabilidade dos documentos e processos legislativos.

Os autores deste estudo visam na Contribuição para a modernização e a digitalização do processo legislativo, acompanhando as tendências tecnológicas atuais.

6. Considerações Finais

Consideramos que através deste estudo foi crucial destacar a interação entre as tecnologias selecionadas - Java, MySQL e Ubuntu Server - e outros elementos importantes, como a Lei Orgânica Municipal, a LGPD (Lei Geral de Proteção de Dados) e a Modelagem de Sistemas.

Buscamos compreender que a Lei Orgânica Municipal desempenha um papel fundamental na definição do contexto legal em que o protótipo de software será desenvolvido e utilizado. É essencial garantir que o sistema esteja em conformidade com os requisitos e regulamentações estabelecidos pela Lei Orgânica, garantindo assim sua legitimidade e adesão às diretrizes governamentais locais.

Vale mencionar que a LGPD, por sua vez, representa uma preocupação crescente em relação à proteção e privacidade dos dados dos cidadãos. Ao desenvolver o protótipo de software, é fundamental considerar as disposições da LGPD e implementar medidas adequadas de segurança e privacidade de dados para garantir a conformidade legal e a proteção das informações dos usuários.

Por fim, A Modelagem de Sistemas também desempenha um papel crucial no desenvolvimento do protótipo de software, fornecendo uma estrutura conceitual para entender os requisitos do sistema, identificar funcionalidades essenciais e definir a arquitetura de software adequada. Através de uma abordagem sistemática de modelagem de sistemas, os desenvolvedores podem garantir que o protótipo atenda às necessidades dos usuários finais e funcione de maneira eficiente e eficaz.

Referências

- Brasil (1991). *Lei nº8159, de 8 de janeiro de 1991. Dispõe sobre a política nacional de arquivos públicos e privados no Brasil*. URL: https://www.planalto.gov.br/ccivil_03/leis/18159.htm.
- (2018). *Lei nº13.709/2018, de 14 de Agosto de 2018. Dispõe sobre o tratamento de dados pessoais físicos e digitais no Brasil*. URL: https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/113709.htm.
- Constituição da República Federativa do Brasil de 1988 - Capítulo IV dos Municípios. Art. 29 (s.d.)*. URL: <https://portal.stf.jus.br/constituicao-supremo/artigo.asp?abrirBase=CF&abrirArtigo=29>.
- Constituição da República Federativa do Brasil de 1988 - Capítulo IV dos Municípios. Art. 30 (s.d.)*. URL: <https://constituicao.stf.jus.br/dispositivo/cf-88-parte-1-titulo-3-capitulo-4-artigo-30>.

- Constituição da República Federativa do Brasil de 1988 - Capítulo IV dos Municípios.*
Art. 31 (s.d.). URL: <https://constituicao.stf.jus.br/dispositivo/cf-88-parte-1-titulo-3-capitulo-4-artigo-31>.
- Eirão, Thiago Gomes e Marilene Mendes Sow (2013). “Sistema de Legislação Informatizada (Legin): uma experiência em gestão da informação jurídica na Câmara dos Deputados”. Em: *E-legis, Brasília* 10, pp. 89–100.
- Sathler, André Rehbein e Malena Rehbein Sathler (2016). “Representação e parlamento: uma abordagem sócio-genética”. Em: *Revista Brasileira de Ciências Sociais* 31, e319105.