
Documentação de Projeto

para o sistema

Gabinete Virtual

Versão 1.3.3

Projeto de sistema elaborado pelo aluno Bernardo Cavanellas Biondini e apresentado ao curso de **Engenharia de Software** da **PUC Minas** como parte do Trabalho de Conclusão de Curso (TCC) sob orientação de conteúdo dos professores Danilo de Quadros Maia, Leonardo Vilela Cardoso e Raphael Ramos Dias Costa, orientação acadêmica do professor Cleiton Silva Tavares e orientação de TCC II do professor (a ser definido no próximo semestre).

05/10/2024

Tabela de Conteúdo

Tabela de Conteúdo	2
Histórico de Revisões	2
1. Introdução	1
2. Modelos de Usuário e Requisitos	1
2.1 Descrição de Atores	2
2.2 Modelos de Usuários	3
Persona 1 - Chiara Biondini	3
Persona 2 - Raquel Assis	5
Persona 3 - Fernanda	6
Persona 4 - Equipe de Atendimento	7
Persona 5 - Equipe Geral (Funcionários)	9
2.3 Modelos de Casos de Uso e Histórias de Usuário	10
2.3.1 Diagrama de Casos de Uso	10
2.4 Diagrama de Sequência do Sistema e Contrato de Operações	13
3. Modelos de Projeto	33
3.1 Diagrama de Classes	34
3.2 Diagramas de Sequência	35
3.3 Diagramas de Comunicação	63
3.4 Arquitetura	98
3.5 Diagramas de Atividade	101
3.6 Diagrama de Componentes e Implantação.	103
4. Projeto de Interface com Usuário	106
4.1 Esboço das Interfaces	106
5. Glossário e Modelos de Dados	130

Histórico de Revisões

Nome	Data	Razões para Mudança	Versão
Iniciação do Documento	25/09/2025	N/A	1
Histórias de Usuário	04/10/2025	Adição das histórias de usuário	1.1.1

Diagrama de Casos de Uso	05/10/2025	Adição do Diagrama de Casos de Uso	1.1.2
Finalização Seção 2	11/10/2025	Ajustes e término da seção 2 (sem correções)	1.1.3
Entrega A4	12/10/2025	Adiciona diagramas de classe, de sequência e de comunicação	1.2
Entrega A5	02/11/2025	Seções 3.4, 3.5, 3.6 e 5. Realiza ajustes da correção A4	1.3
Correções A5	14/11/2025	Adição de descrição nas tabelas e imagens e contextualização no texto.	1.3.2
Atualização dos Wireframes	17/11/2025	Atualização dos Wireframes	1.3.3

1. Introdução

O sistema Gabinete Virtual é uma plataforma web desenvolvida para otimizar a gestão de demandas, emendas parlamentares e atividades administrativas de um gabinete político. Seu objetivo é integrar, em um único ambiente, informações que tradicionalmente ficam dispersas em planilhas, mensagens e documentos, oferecendo uma visão consolidada e estratégica das ações do mandato.

O sistema foi projetado com uma arquitetura cliente-servidor, composta por um frontend em React.js e um backend em Laravel/PHP, comunicando-se via API REST e utilizando um banco de dados relacional MySQL. Essa combinação garante escalabilidade, modularidade e facilidade de manutenção. Além disso, a aplicação contempla níveis de acesso específicos para diferentes perfis (deputada, chefe de gabinete, gestora de demandas, equipe de atendimento e cidadãos) promovendo transparência, eficiência operacional e controle das informações públicas.

A proposta central é permitir que o gabinete monitore suas ações de forma inteligente, automatizando tarefas repetitivas e proporcionando relatórios e dashboards que auxiliem na tomada de decisão política e administrativa.

Este documento agrega: 1) a elaboração e revisão de modelos de domínio e 2) modelos de projeto para o sistema Gabinete Virtual. A referência principal para a descrição geral do problema, domínio e requisitos do sistema é o documento de especificação que descreve a visão de domínio do sistema. Tal especificação acompanha este documento. Anexo a este documento também se encontra o Glossário.

2. Modelos de Usuário e Requisitos

Esta seção descreve os atores, perfis e necessidades que interagem com o sistema, além de apresentar os modelos de casos de uso, histórias de usuário e demais representações que orientam o desenvolvimento do projeto.

O objetivo é compreender quem são os usuários, o que esperam do sistema e como suas interações moldam o design e os requisitos funcionais do Gabinete Virtual.

2.1 Descrição de Atores

Nesta subseção é apresentado descrição de cada um dos atores que interagem com o sistema. Os atores definidos na Tabela 1 a seguir representam os diferentes perfis de usuários que interagem com o Gabinete Virtual. Cada ator desempenha funções específicas no contexto de atendimento à população, gestão administrativa e comunicação institucional.

Perfil	Descrição	Objetivos	Nível de acesso
Deputada	Usuária principal, tomadora de decisão política.	Consultar informações estratégicas sobre demandas, emendas, cidades e compromissos.	Leituras e relatórios consolidados
Chefe de Gabinete	Responsável pela coordenação geral da equipe e da agenda.	Organizar compromissos, acompanhar demandas e priorizar atividades.	Leitura, edição e controle sobre registros.
Gestora	Responsável por designar tarefas e acompanhar execuções.	Atribuir responsáveis, monitorar prazos e atualizar status das atividades.	Edição e controle de demandas e eventos.
Gestora de Demandas	Responsável por designar tarefas e acompanhar execuções.	Registrar novas demandas, atualizar informações e alimentar relatórios.	Acesso restrito às funções operacionais.
Atendimento	Responsável pelo	Acompanhar	Acesso restrito às

	contato externo.	demandas, verificar informações de lideranças.	funções operacionais
Funcionários	Executor das tarefas administrativas e operacionais		Acesso restrito às funções operacionais
Cidadão (Consulta Pública)	Usuário externo interessado em informações públicas.x'	Acessar dados do site oficial, como notícias, projetos e emendas.	Apenas leitura (público).

Tabela 1 - Descrição dos Atores do Sistema Gabinete Virtual

2.2 Modelos de Usuários

Esta subseção apresenta os modelos de usuários desenvolvidos para o Sistema de Gestão de Demandas de Gabinete, a partir da perspectiva do designer. Foram utilizados modelos de personas para guiar as decisões de design e usabilidade. As Tabelas 2 a 6 apresentam os modelos de usuários adotados no projeto, sendo cada uma explicada individualmente a seguir.

A Tabela 2 apresenta a persona Deputada, considerada o nível máximo de acesso dentro do Gabinete Virtual. Essa persona é responsável pela tomada de decisões estratégicas, consulta de indicadores, acompanhamento de emendas, projetos de lei e demandas gerais. A Deputada utiliza o sistema principalmente para obter visões consolidadas, acessar relatórios completos e validar informações cadastradas pela equipe. Sua necessidade central é ter uma interface simples, rápida e capaz de fornecer dados confiáveis em tempo real, permitindo decisões políticas baseadas em evidências.

Persona 1 - Chiara Biondini

Cargo	Deputada Estadual
-------	-------------------

Persona 1 - Chiara Biondini	
Idade	23 anos
Formação	Administração
Perfil Tecnológico	Básico – Utiliza o sistema principalmente para consultas em dispositivos móveis. Prefere interfaces intuitivas e informações resumidas.
Responsabilidades	Tomada de decisões estratégicas, consulta de informações sobre municípios, demandas e emendas. Representação política e prestação de contas às comunidades.
Motivações	Ter informações rápidas e confiáveis antes de visitas e reuniões políticas; Demonstrar transparência e eficiência na gestão do mandato; Tomar decisões estratégicas baseadas em dados consolidados.
Frustrações	Dificuldade em acessar relatórios resumidos e navegar em sistemas complexos; Falta de informações consolidadas durante visitas aos municípios; Dependência de terceiros para obter informações estratégicas.
Necessidades de Design	Interface simplificada e intuitiva; Visualização de gráficos e relatórios prontos; Acesso rápido via dispositivos móveis; Dashboard com informações-chave; Relatórios executivos sintéticos.
Citação Representativa	“Quero abrir o sistema e entender rapidamente o que está acontecendo nas cidades que visito.”

Tabela 2 - Descrição Persona 1 do Gabinete Virtual

A Tabela 3 descreve a persona Chefe de Gabinete, que atua como figura intermediária entre a Deputada e os demais setores. Esse usuário precisa de acesso ampliado para gerenciar a agenda, supervisionar demandas, acompanhar emendas e auxiliar na priorização de ações. A persona usa o sistema para organizar processos, revisar cadastros e orientar fluxos internos, garantindo que os encaminhamentos estejam alinhados à estratégia do mandato. Seu foco está na coordenação operacional.

Persona 2 - Raquel Assis	
Cargo	Chefe de Gabinete
Idade	38 anos
Formação	Administração Pública
Perfil Tecnológico	Intermediário – Usa sistemas administrativos, planilhas e ferramentas online de agenda. Familiarizada com gestão de equipes remotas.
Responsabilidades	Coordenação da equipe do gabinete; Organização da agenda da deputada; Supervisão do andamento das demandas; Gestão de conflitos internos; Comunicação entre equipe e deputada.
Motivações	Ter uma visão consolidada das atividades do gabinete; Acompanhar o progresso das demandas em tempo real; Garantir que a equipe trabalhe de forma sincronizada; Evitar conflitos de agenda e perda de prazos.
Frustrações	Informações dispersas em planilhas e mensagens; Falta de notificações automáticas; Retrabalho constante; Dificuldade em priorizar tarefas urgentes; Ausência de visão consolidada do gabinete.

Persona 2 - Raquel Assis	
Necessidades de Design	Painel de controle com visão geral; Alertas visuais de prazos e prioridades; Interface limpa e organizada; Filtros por status e responsável; Sistema de notificações eficiente; Calendário integrado com detecção de conflitos.
Citação Representativa	“Preciso ter uma visão rápida de tudo o que está acontecendo no gabinete, sem perder prazos.”

Tabela 3 - Descrição Persona 2 do Gabinete Virtual

Na Tabela 4, é apresentada a persona Gestora de Demandas, responsável pelo gerenciamento direto das solicitações recebidas pelo gabinete. Esse usuário utiliza o sistema para abrir, classificar, atribuir responsáveis, monitorar prazos e atualizar status das demandas. Além disso, acessa relatórios específicos para análise de volume, prazos médios e gargalos. Sua interface exige eficiência, clareza e controle rigoroso dos registros. Essa persona é fundamental para o processamento interno do gabinete.

Persona 3 - Fernanda	
Cargo	Gestora de Demandas
Idade	32 anos
Formação	Gestão Pública
Perfil Tecnológico	Avançado – Usa softwares de produtividade, dashboards e ferramentas de comunicação interna. Confortável com sistemas complexos.
Responsabilidades	Designar responsáveis para cada demanda; Acompanhar o andamento de tarefas; Gerar relatórios de status; Garantir a execução das demandas dentro dos prazos; Analisar métricas de desempenho da equipe.

Persona 3 - Fernanda	
Motivações	Otimizar o trabalho da equipe; Garantir a execução eficiente das demandas; Ter métricas claras de desempenho; Identificar gargalos e oportunidades de melhoria.
Frustrações	Falta de histórico centralizado; Dificuldade para gerar relatórios precisos; Ausência de indicadores de desempenho da equipe; Perda de contexto sobre demandas antigas; Dificuldade em rastrear responsabilidades.
Necessidades de Design	Filtros eficientes e busca avançada; Campos padronizados; Botões de atualização rápida; Feedback visual claro; Histórico de alterações detalhado; Relatórios customizáveis; Indicadores visuais de performance.
Citação Representativa	“Quero um sistema que me mostre o que está pendente e quem é o responsável, de forma visual e objetiva.”

Tabela 4 - Descrição Persona 3 do Gabinete Virtual

A Tabela 5 documenta a persona Atendimento, responsável pelo registro inicial de demandas e pelo relacionamento direto com cidadãos, instituições e lideranças. Esse usuário cadastra informações, anexa documentos, atualiza registros e repassa demandas para análise. Sua interação demanda telas simples, diretas e otimizadas para rapidez, já que frequentemente trabalha sob fluxo contínuo de solicitações. O papel é essencial para a etapa de entrada de dados no sistema.

Persona 4 - Equipe de Atendimento	
Cargo	Atendentes do Gabinete
Idade	26–35 anos

Persona 4 - Equipe de Atendimento			
Formação	Ciências Políticas, Administração, Comunicação Social		
Perfil Tecnológico	Intermediário – Acostumados com planilhas e sistemas administrativos simples. Necessitam de orientação clara sobre processos.		
Responsabilidades	Registrar e atualizar demandas; Cadastrar instituições e lideranças; Registrar pessoas atendidas; Definir prazos iniciais; Anexar documentos (ofícios); Enviar informações aos gestores.		
Motivações	Executar as tarefas com clareza; Evitar retrabalho ou dúvidas sobre o fluxo correto; Contribuir para o bom funcionamento do gabinete; Ter reconhecimento pelo trabalho bem feito.		
Frustrações	Falta de padronização no preenchimento; Erros por falta de feedback do sistema; Dificuldade em localizar informações cadastradas anteriormente; Incerteza sobre campos obrigatórios; Perda de tempo em tarefas repetitivas.		
Necessidades de Design	Campos com preenchimento guiado; Validações em tempo real; Mensagens de confirmação claras; Histórico de alterações acessível; Tooltips explicativos; Formulários intuitivos; Autocomplete e sugestões.		
Citação Representativa	“Quero saber exatamente o que devo preencher e ver que a informação foi salva corretamente.”		

Tabela 5 - Descrição Persona 4 do Gabinete Virtual

Por fim, a Tabela 6 apresenta a persona Funcionários, composta por membros da equipe que apoiam atividades administrativas gerais. Esses usuários possuem permissões mais restritas, atuando em cadastros específicos, atualização de informações e suporte a rotinas internas. Não acessam relatórios estratégicos nem funcionalidades sensíveis. O sistema deve oferecer segurança e delimitação clara de permissões, garantindo que esse grupo interaja apenas com os módulos correspondentes ao seu nível de responsabilidade.

Persona 5 - Equipe Geral (Funcionários)	
Cargo	Funcionários do Gabinete
Idade	28–50 anos
Formação	Diversas áreas
Perfil Tecnológico	Básico a Intermediário – Usuários ocasionais do sistema, principalmente para consultas e cadastros básicos.
Responsabilidades	Cadastrar emendas parlamentares; Atualizar status de projetos; Consultar informações sobre cidades, lideranças, instituições e igrejas; Manter dados atualizados; Cadastrar e atualizar conteúdo do site.
Motivações	Contribuir com a organização geral do gabinete; Ter acesso fácil a informações necessárias para o trabalho diário; Executar tarefas de forma eficiente; Manter dados precisos e atualizados.
Frustrações	Interfaces complexas; Falta de treinamento adequado; Dificuldade em encontrar as funcionalidades necessárias; Menus confusos; Falta de documentação clara.

Persona 5 - Equipe Geral (Funcionários)	
Necessidades de Design	Navegação intuitiva; Menus claros e organizados; Busca eficiente; Acesso direto às funcionalidades mais utilizadas; Ajuda contextual; Manual do usuário acessível; Organização lógica das informações.
Citação Representativa	"Preciso de um sistema simples que me permita encontrar e atualizar informações rapidamente."

Tabela 6 - Descrição Persona 5 do Gabinete Virtual

2.3 Modelos de Casos de Uso e Histórias de Usuário

Os requisitos funcionais foram organizados em casos de uso, representando as interações de cada ator com o sistema. Os casos de uso abrangem: Gestão de cadastros (instituições, lideranças, cidades, igrejas), Gestão de conteúdo institucional, Gestão de projetos de lei, Gestão de emendas, Gestão de demandas, Gestão da agenda, Relatórios segmentados

2.3.1 Diagrama de Casos de Uso

A Figura 1 apresenta o diagrama de casos de uso do sistema Gabinete Virtual, reunindo em uma única visão todos os atores e funcionalidades do sistema. Os diferentes perfis (Deputada, Chefe de Gabinete, Gestora de Demandas, Atendimento e Funcionários) encontram-se à esquerda, enquanto, no interior do sistema, estão organizados os casos de uso que representam os principais módulos da plataforma.

No centro do diagrama está o UC01 – Autenticar no Sistema + Validar Permissão, modelado como um caso de uso transversal. Essa escolha se deve ao fato de que todas as funcionalidades dependem desse fluxo padrão, que envolve autenticação, verificação de perfil e validação das permissões específicas. Por isso, todos os demais casos de uso possuem relação *include* com o UC01, conforme registrado na nota explicativa.

A partir desse caso central, o diagrama estrutura seis áreas funcionais principais. O UC02 – Gerenciar Cadastros agrupa operações de inclusão e consulta de Instituições, Lideranças, Cidades e Igrejas, reunidas em um único caso devido à similaridade operacional dessas rotinas administrativas. O UC03 – Gerenciar Conteúdo do Site contempla ações relacionadas ao CMS do gabinete, como publicação de notícias, atualização de páginas institucionais e registro de projetos exibidos ao público.

Os módulos legislativos são representados no UC04 – Gerenciar Projetos de Lei e no UC05 – Gerenciar Emendas, ambos estruturados em fluxos de cadastro, atualização de status e consulta, refletindo práticas internas de trabalho. O UC06 – Gerenciar Demandas consolida um conjunto maior de ações, entre elas abertura de demandas, atribuição de responsáveis, registro de atendimentos, anexação de ofícios, definição de prazos e notificações, agrupadas em um único caso por fazerem parte de um mesmo ciclo de atendimento.

O UC07 – Gerenciar Agenda reúne funcionalidades como criação de eventos, visualização de calendários, detecção de conflitos e alertas. Por fim, os casos UC08a, UC08b, UC08c e UC08d representam relatórios distintos, cada um com regras próprias de acesso: PLs e Emendas para Deputada e Chefe de Gabinete; Demandas para Deputada, Gestora de Demandas e Atendimento; e o Relatório Consolidado Geral exclusivo da Deputada. As notas ao lado de cada caso de uso reforçam seu escopo e as permissões vinculadas.

Assim, a Figura 1 sintetiza todas as capacidades do sistema, evidenciando como cada ator interage com os diferentes módulos e como o núcleo de autenticação e permissões permeia toda a operação, garantindo clareza, segurança e coerência com o funcionamento real de um gabinete parlamentar.

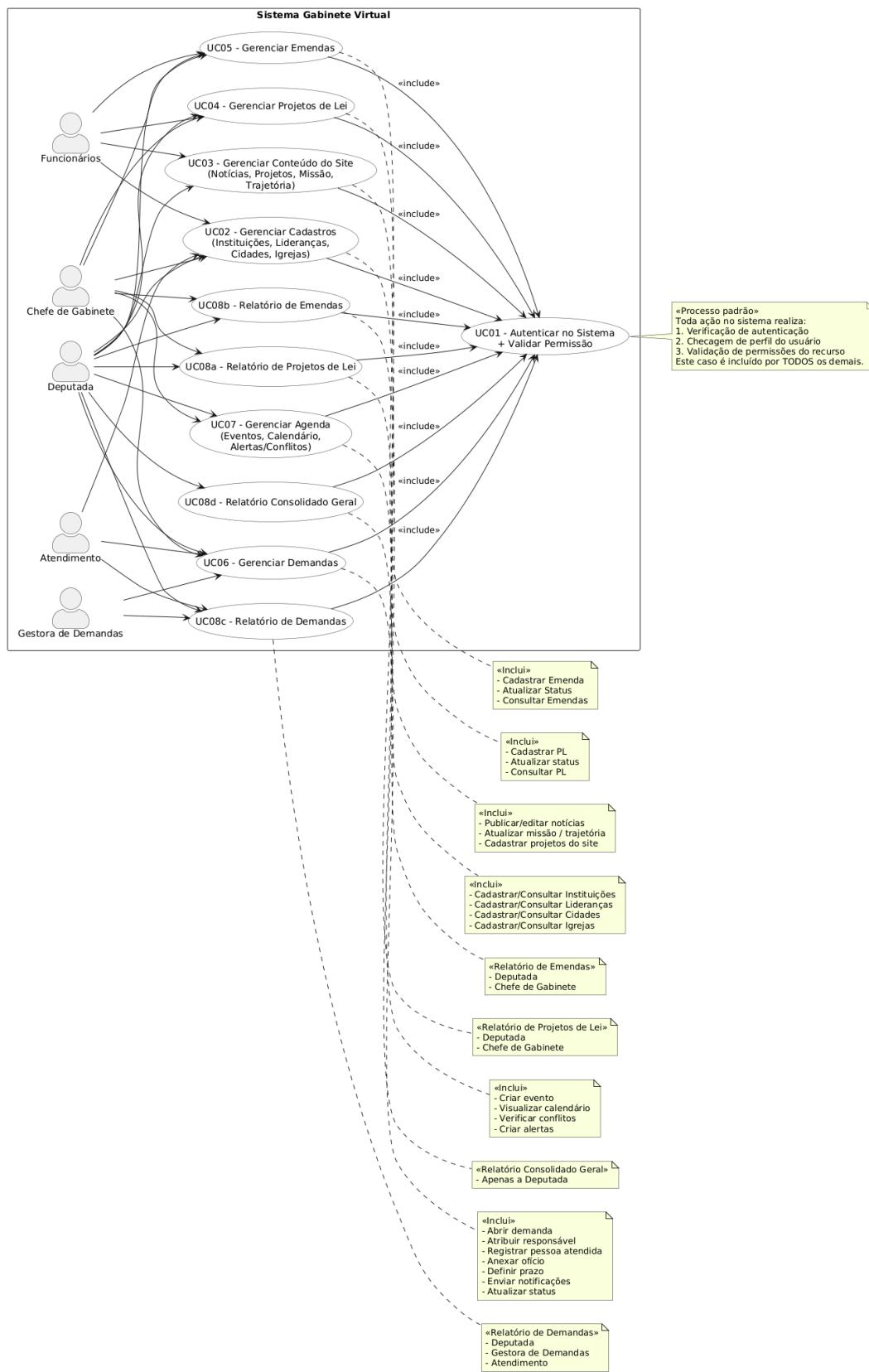


Figura 1 - Diagrama de Casos de Uso do Gabinete Virtual

2.4 Diagrama de Sequência do Sistema e Contrato de Operações

Nesta seção, são apresentados os Diagramas de Sequência do Sistema (DSS) para as principais operações que modificam o estado do sistema, acompanhados de seus respectivos contratos de operação.

Como mostra a Figura 2, o usuário informa suas credenciais, e o sistema aciona o serviço de autenticação para validar e verificar se o usuário está ativo. Se as credenciais forem válidas, um token é gerado e retornado ao cliente, permitindo que ele accesse as funcionalidades conforme o seu perfil. Caso contrário, uma mensagem de erro é emitida. O diagrama contempla os cenários de autenticação bem-sucedida e falha por credenciais incorretas ou usuário inativo. A tabela 7 apresenta o contrato de operação para o Diagrama.

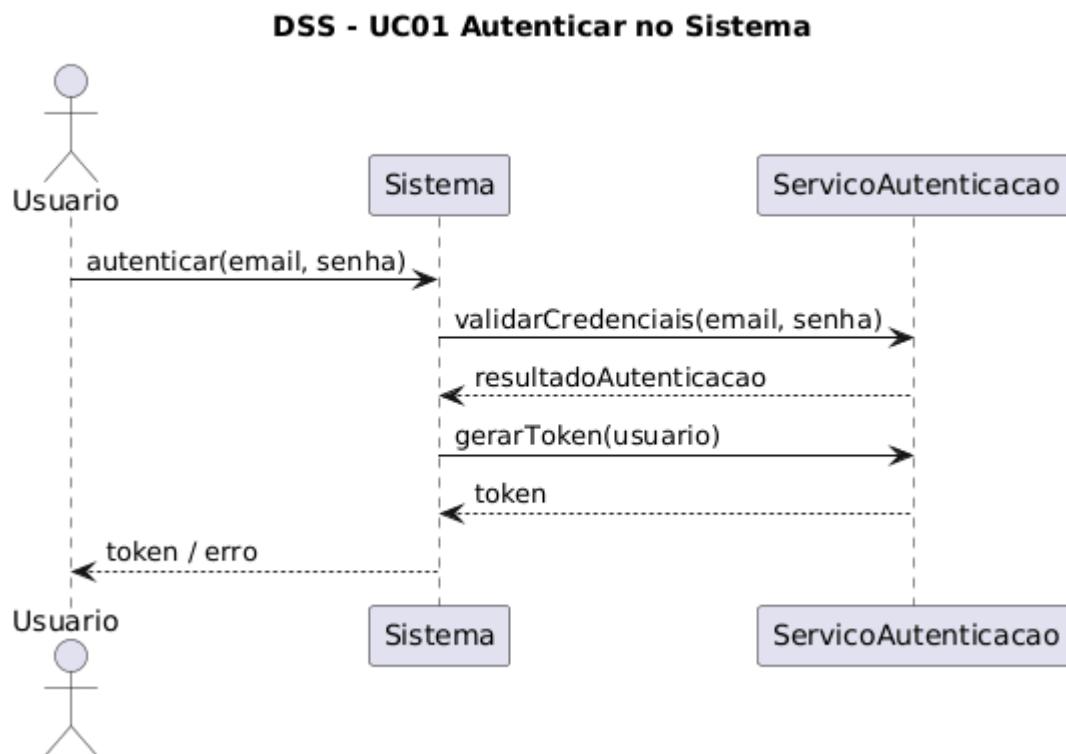


Figura 2 - DSS: Autenticar no Sistema

Operação	autenticar(email, senha)
Referências Cruzadas	UC01 – Autenticar no Sistema Classes: Usuario, PerfilAcesso
Pré-condições	Usuário cadastrado e ativo Credenciais informadas

Pós-condições	Token gerado Último login atualizado
---------------	---

Tabela 7 - Contrato de Operação: Autenticar no Sistema

A Figura 3 demonstra o fluxo onde o usuário solicita o cadastro de uma instituição. O sistema primeiro valida autenticação e permissão específica para essa ação. Em seguida, os dados são enviados ao repositório, que persiste a instituição e devolve o objeto criado. O caso contempla os cenários de sucesso, além de falhas por campos inválidos ou ausência de permissão. O Contrato de Operação referente está disponível na Tabela 8.

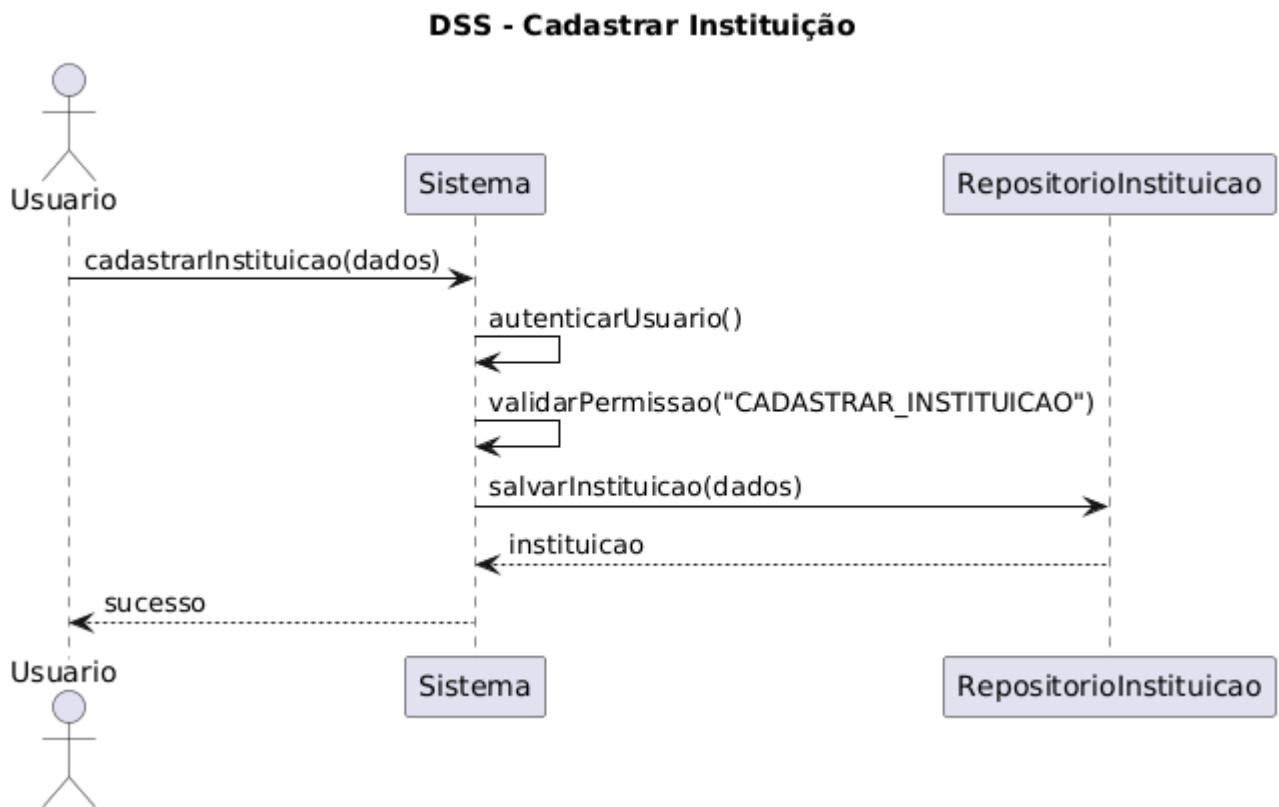


Figura 3 - DSS: Cadastrar Instituição

Operação	cadastrarInstituicao(dados)
Referências Cruzadas	UC03 – Cadastrar Instituição

Pré-condições	Usuário autenticado Permissão CADASTRAR_INSTITUICAO
Pós-condições	Instituição salva no banco

Tabela 8 - Contrato de Operação: Cadastrar Instituição

Como mostra a Figura 4 e o contrato de operação da Tabela 9, o ator submete os dados de uma nova liderança. Após validar identidade e permissão, o sistema verifica também se a instituição associada existe. Com todas as condições atendidas, a liderança é registrada e o sistema retorna os dados persistidos. O DSS evidencia o fluxo principal e possíveis falhas de validação relacionadas ao vínculo com a instituição.

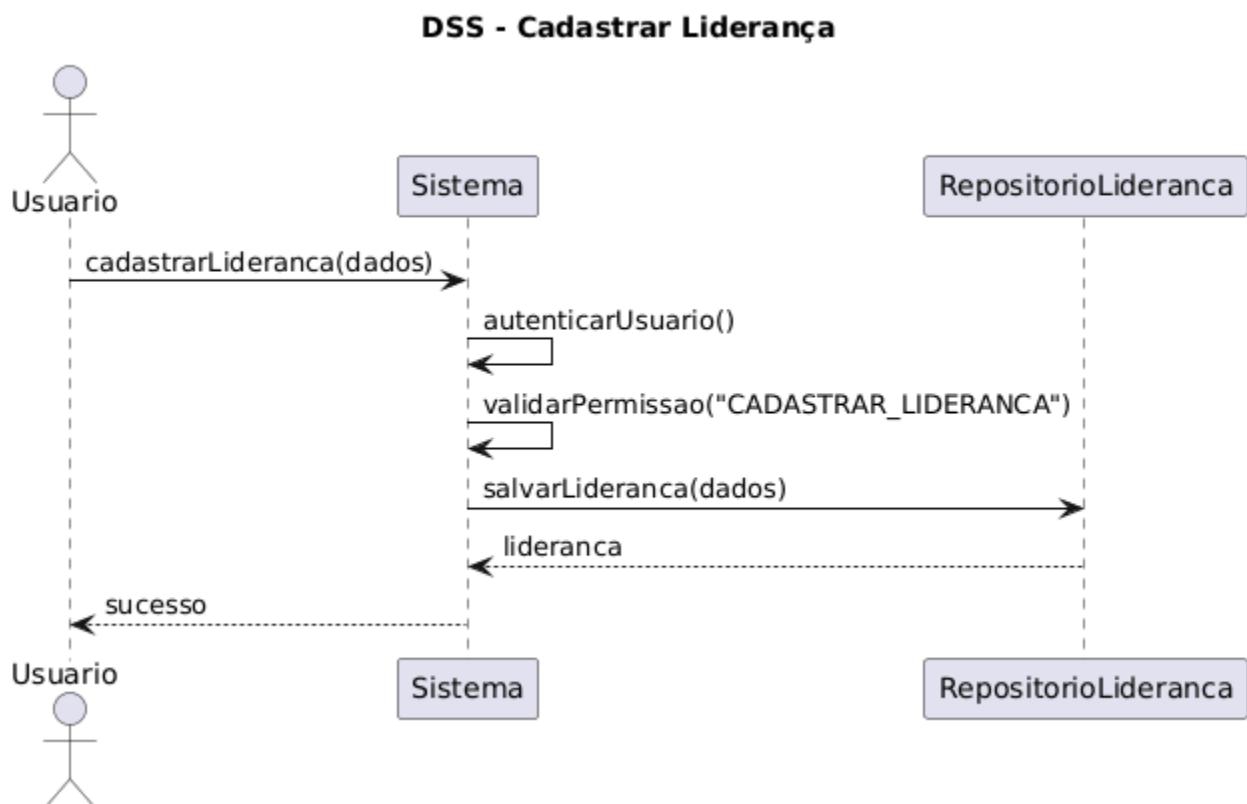


Figura 4 - DSS: Cadastrar Liderança

Operação	cadastrarLideranca(dados)
Referências Cruzadas	UC03 – Cadastrar Liderança
Pré-condições	Usuário autenticado Permissão CADASTRAR_LIDERANCA

Pós-condições	Liderança cadastrada
---------------	----------------------

Tabela 9 - Contrato de Operação: Cadastrar Liderança

A Figura 5 demonstra o cadastro de uma cidade e a Tabela 10 o contrato de operação referente. O sistema autentica o usuário, verifica permissões e persiste os dados no banco. O diagrama apresenta o fluxo de criação e o cenário onde a operação falha devido a dados incompletos ou ausência de permissão adequada.

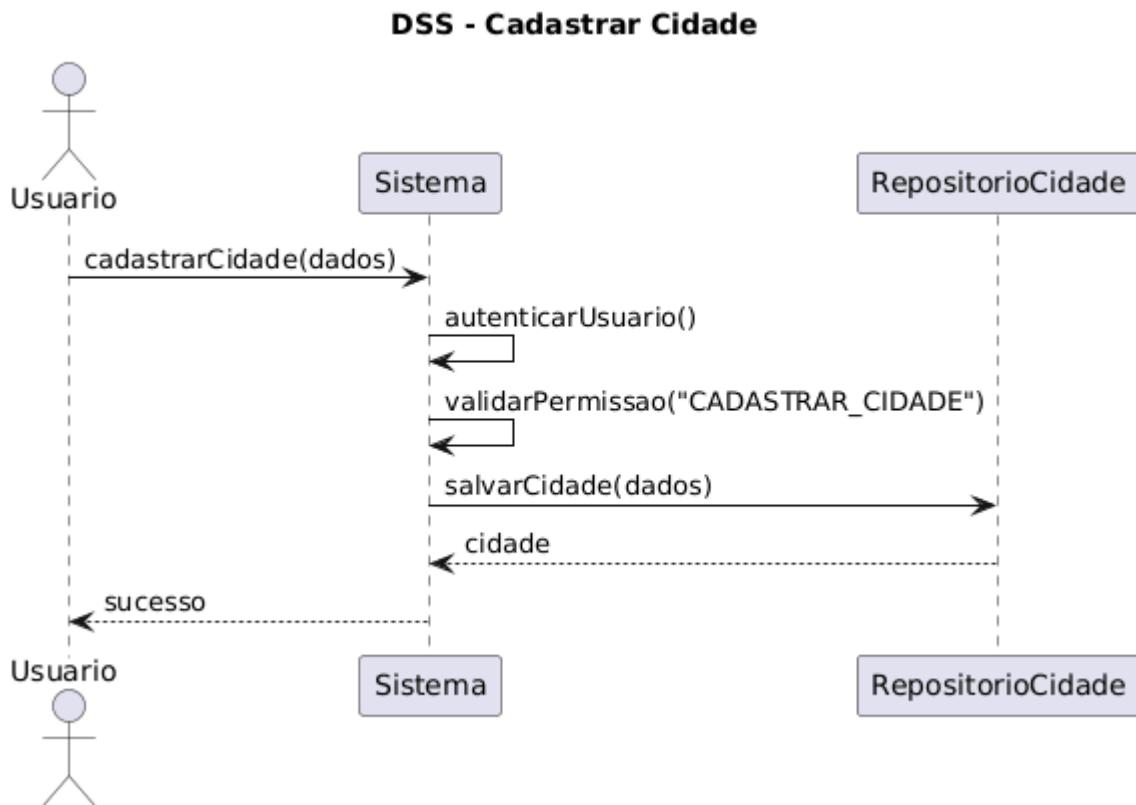


Figura 5 - DSS: Cadastrar Cidade

Operação	cadastrarCidade(dados)
Referências Cruzadas	UC03 – Cadastrar Cidade
Pré-condições	Usuário autenticado Permissão CADASTRAR_CIDADE
Pós-condições	Cidade registrada

Tabela 10 - Contrato de Operação: Cadastrar Cidade

Na Figura 6, o usuário cobra uma igreja vinculada a um município. O fluxo inclui validação de permissão, confirmação da existência da cidade e persistência da entidade. O diagrama destaca tanto

o fluxo bem-sucedido quanto potenciais rejeições por inconsistências no vínculo ou permissões insuficientes. A Tabela 11 apresenta o Contrato de Operação do DSS presente na imagem.

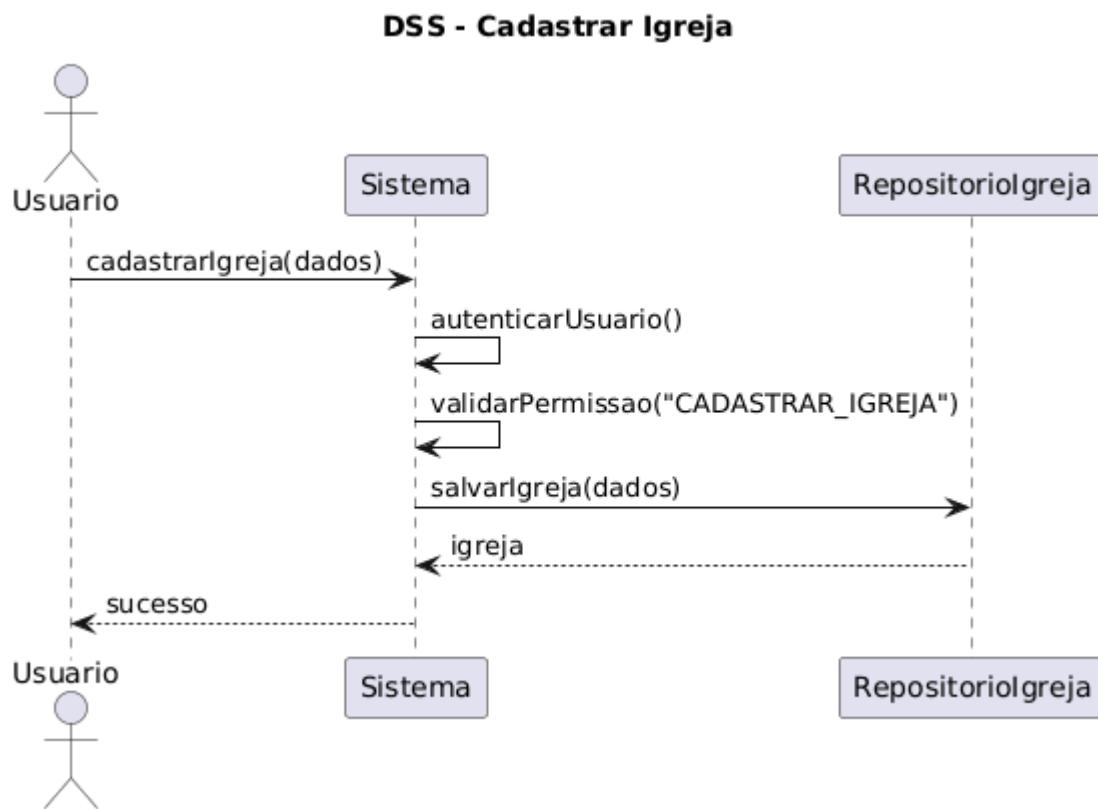


Figura 6 - DSS: Cadastrar Igreja

Operação	cadastrarIgreja(dados)
Referências Cruzadas	UC03 – Cadastrar Igreja
Pré-condições	Usuário autenticado Permissão CADASTRAR_IGREJA
Pós-condições	Igreja cadastrada

Tabela 11 - Contrato de Operação: Cadastrar Igreja

A Figura 7 descreve como o ator registra uma notícia para publicação. O sistema autentica o usuário, verifica permissões e então persiste a notícia com autor associado. O DSS contempla o cenário de sucesso e condições de erro, como ausência de título ou conteúdo e a Tabela 12 contempla o contrato de operação referente a esse cenário.

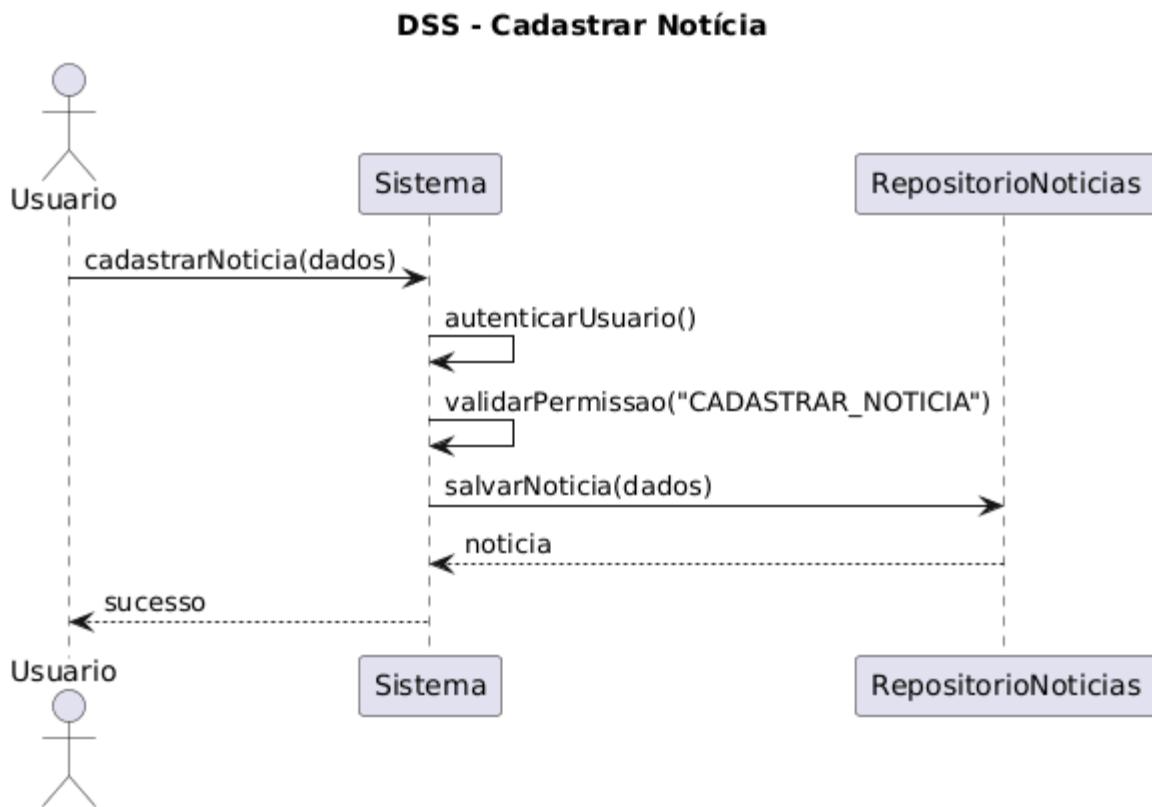


Figura 7 - DSS: Cadastrar Notícia

Operação	cadastrarNoticia(dados)
Referências Cruzadas	UC11 – Cadastrar Notícias
Pré-condições	Usuário autenticado Permissão CADASTRAR_NOTICIA
Pós-condições	Notícia criada

Tabela 13 - Contrato de Operação: Cadastrar Notícia

Como mostra a Figura 8, o usuário altera o texto da seção “Missão” do *site*. Após validação de permissão, o sistema atualiza o conteúdo no CMS. O diagrama contempla tanto o fluxo bem-sucedido quanto o caso em que o texto enviado está vazio ou a permissão não é concedida. Além disso, abaixo é possível observar a Tabela 14 que define o Contrato de Operação referente ao diagrama.

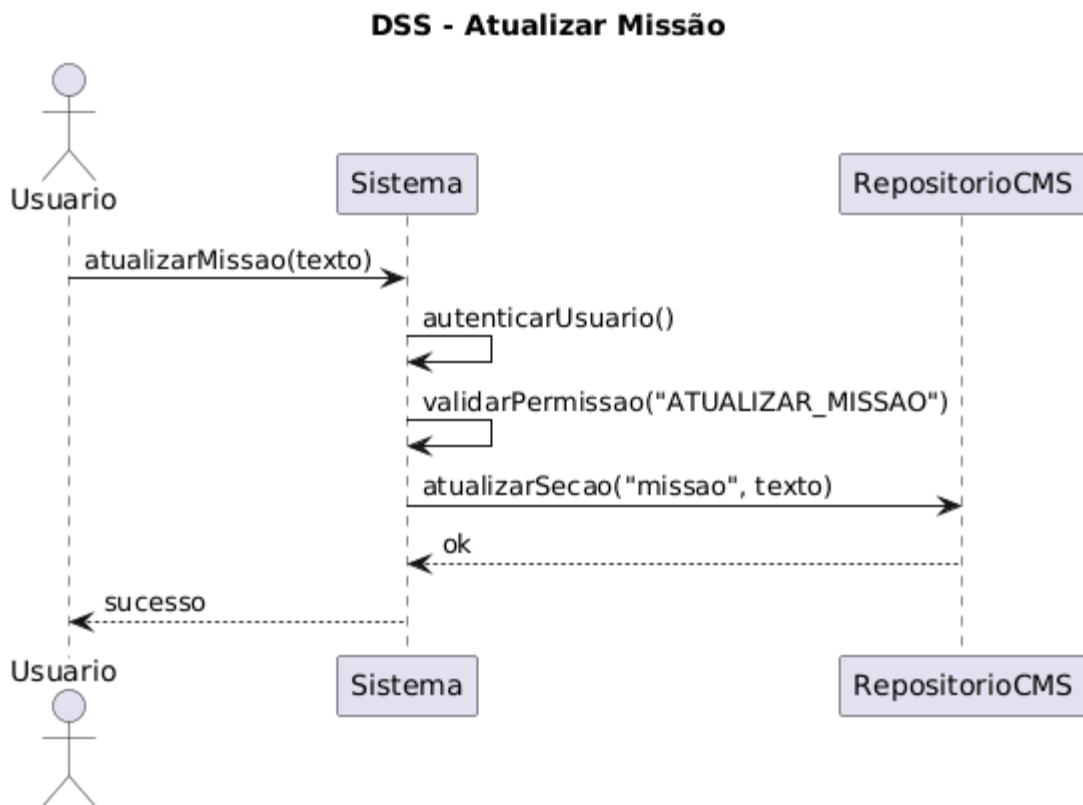


Figura 8 - DSS7: Atualizar Missão

Operação	atualizarMissao(texto)
Referências Cruzadas	UC13 – Atualizar Missão
Pré-condições	Usuário autenticado Permissão ATUALIZAR_MISSAO
Pós-condições	Texto da missão atualizado

Tabela 14 - Contrato de Operação: Atualizar Missão

A Figura 9 demonstra o fluxo de atualização da seção “Trajetória”. Assim como na missão, a ação requer permissão específica. O conteúdo é sobreescrito e torna-se imediatamente disponível para o site. O diagrama evidencia validações e possíveis erros de conteúdo inválido.

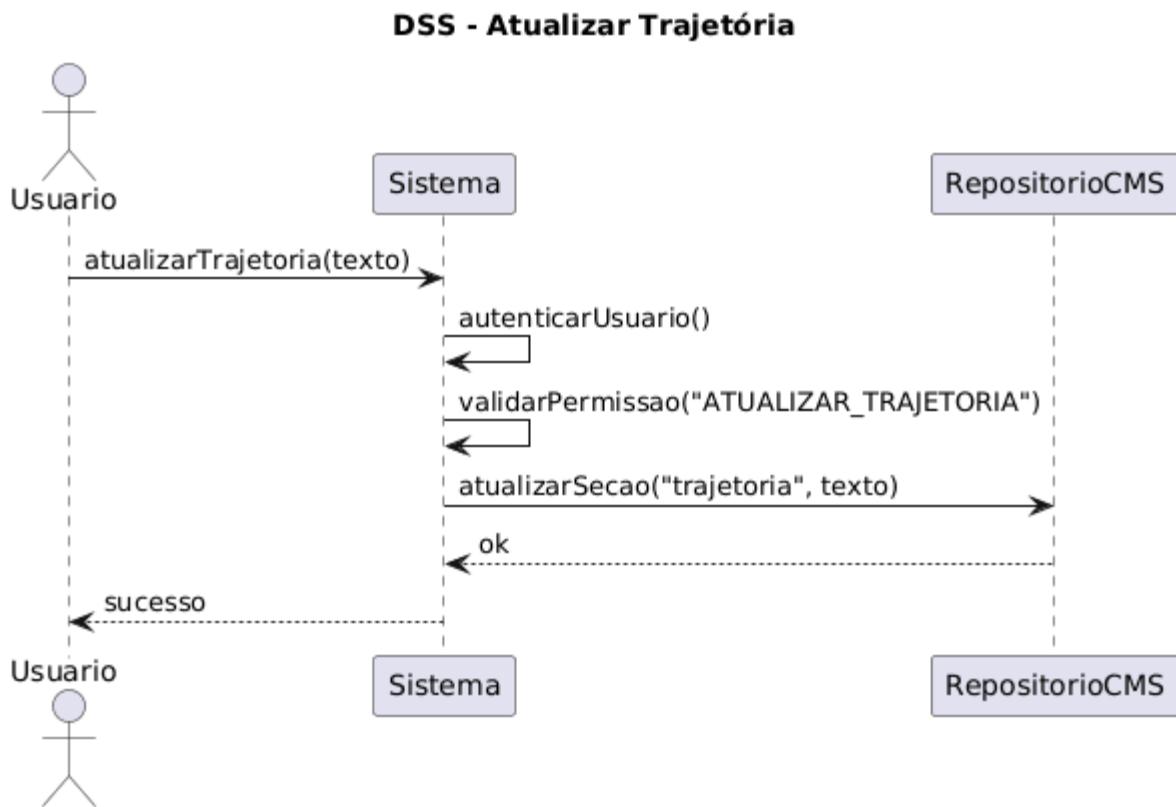


Figura 9 - DSS: Atualizar Trajetória

Operação	atualizarTrajetoria(texto)
Referências Cruzadas	UC14 – Atualizar Trajetória
Pré-condições	Usuário autenticado Permissão ATUALIZAR_TRAJETORIA
Pós-condições	Trajetória atualizada

Tabela 15 - Contrato de Operação: Atualizar Trajetória

Na Figura 10, que antecede a Tabela 16 referente ao Contrato de Operação de Cadastro de Projeto, mostra o fluxo para essa operação. O usuário solicita o cadastro de um projeto institucional para exibição no *site*. O sistema valida permissões, persiste os dados e retorna o projeto cadastrado. O DSS mostra o fluxo principal e possíveis falhas por dados incompletos.

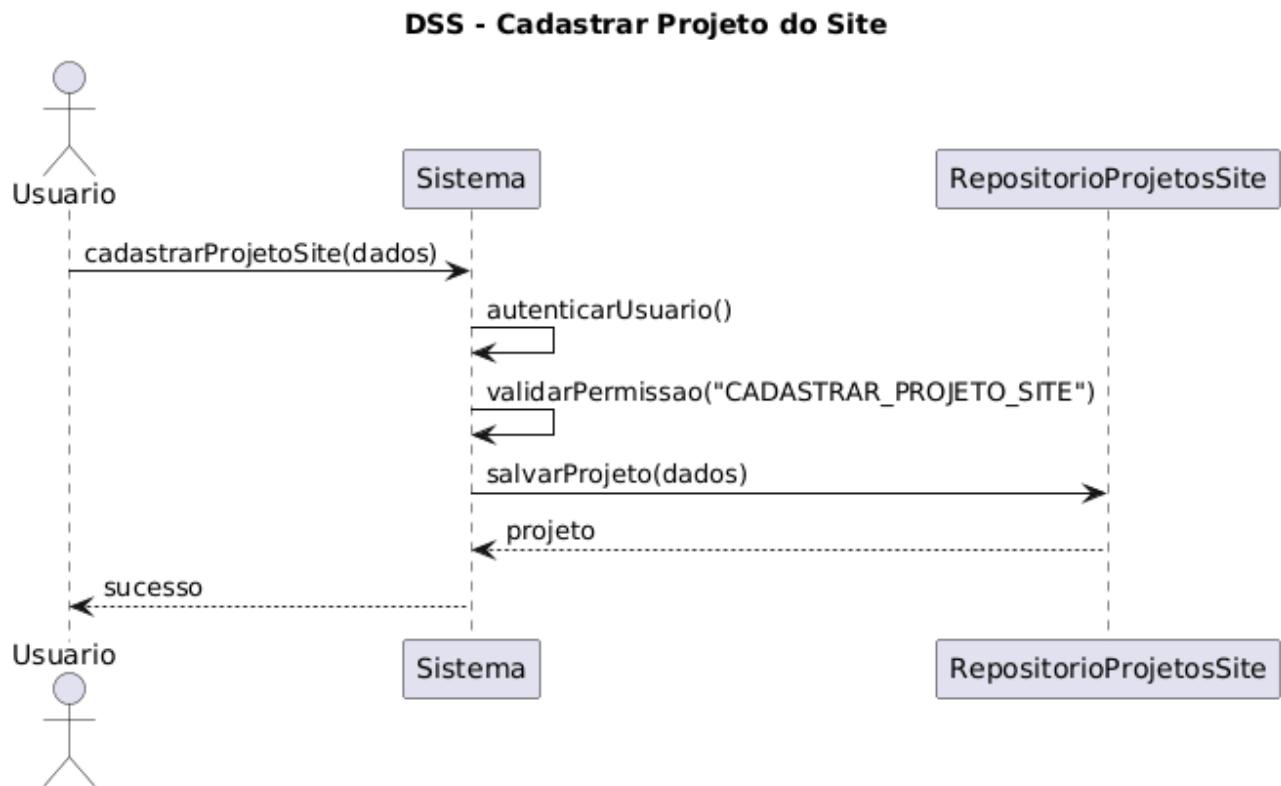


Figura 10 - DSS: Cadastrar Projeto do Site

Operação	cadastrarProjetoSite(dados)
Referências Cruzadas	UC15 – Cadastrar Projetos
Pré-condições	Usuário autenticado Permissão CADASTRAR_PROJETO_SITE
Pós-condições	Projeto salvo

Tabela 16 - Contrato de Operação: Cadastrar Projeto Site

Como mostra a Figura 11, o usuário registra um novo Projeto de Lei. O sistema verifica autenticação, permissões e integridade dos dados (como número e descrição). Após salvamento, o PL fica disponível para consulta e relatórios. O DSS contempla erros por duplicidade ou dados inconsistentes. A Tabela 17 representa o Contrato de Operação deste diagrama.

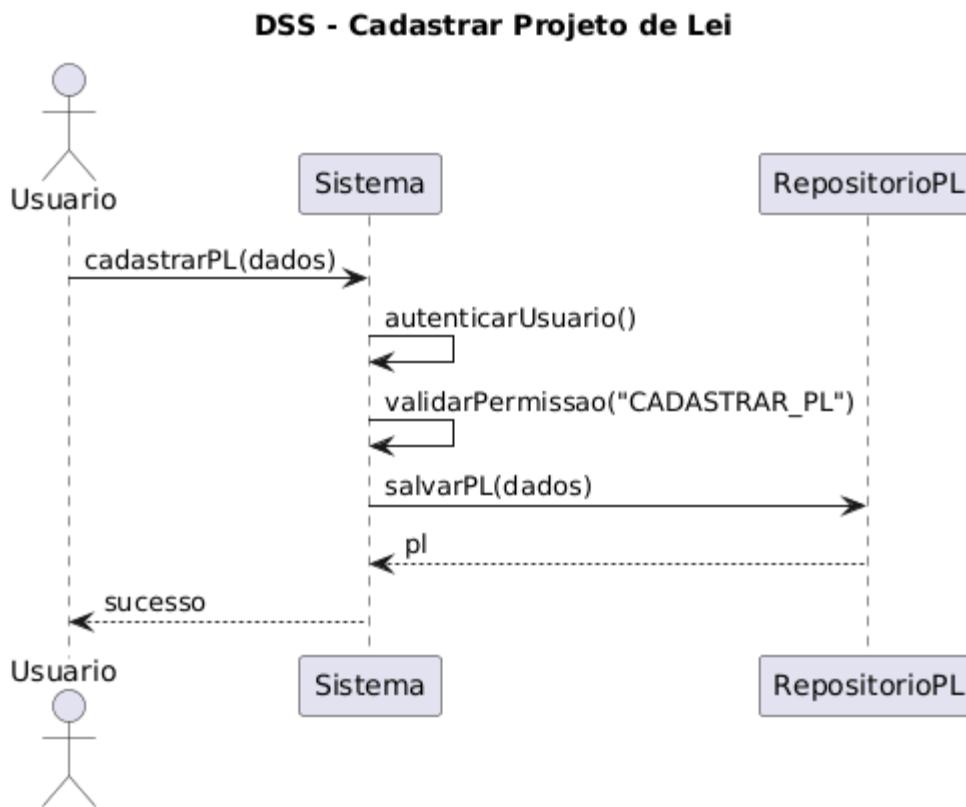


Figura 11 - DSS2: Cadastrar Projeto de Lei

Operação	cadastrarPL(dados)
Referências Cruzadas	UC17 – Cadastrar PL
Pré-condições	Usuário autenticado Permissão CADASTRAR_PL
Pós-condições	PL cadastrado

Tabela 17 - Contrato de Operação: Cadastrar Projeto de Lei

A Figura 12 demonstra a mudança de status de um PL já cadastrado. O sistema valida se o usuário tem permissão e se o novo status é válido. Em seguida, atualiza o registro. O DSS ilustra fluxo de sucesso e falhas por status inválido ou ausência de permissão. O Contrato de Operação equivalente é exibido na Tabela 18.

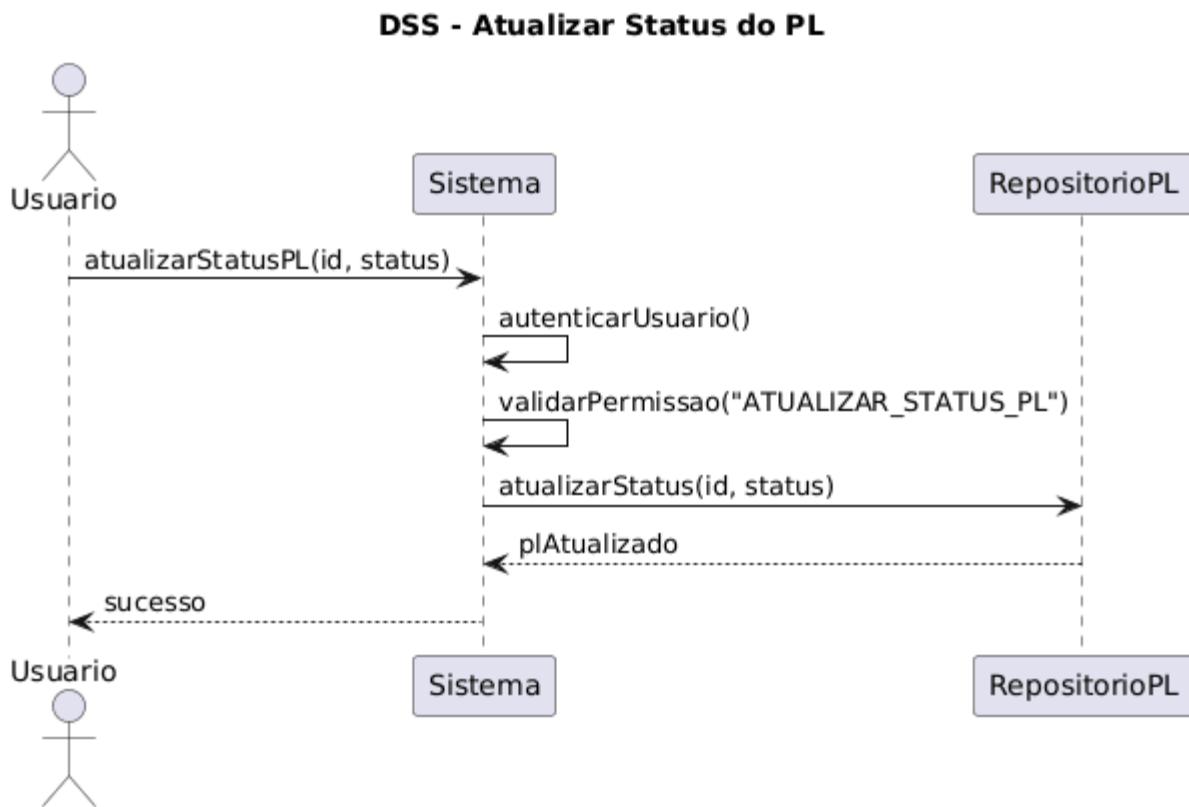


Figura 12 - DSS: Atualizar Status do PL

Operação	atualizarStatusPL(id, status)
Referências Cruzadas	UC18 – Atualizar Status PL
Pré-condições	Usuário autenticado Permissão ATUALIZAR_STATUS_PL
Pós-condições	Status atualizado

Tabela 18 - Contrato de Operação: Atualizar Status Projeto de Lei

Na Figura 13, o ator cadastra uma nova emenda parlamentar. Depois da validação, os dados são persistidos e relacionados ao município. O DSS mostra o fluxo positivo e rejeições por campos obrigatórios faltantes ou vínculo com cidade inexistente, a Tabela 19 exibe o contrato de operação equivalente ao diagrama.

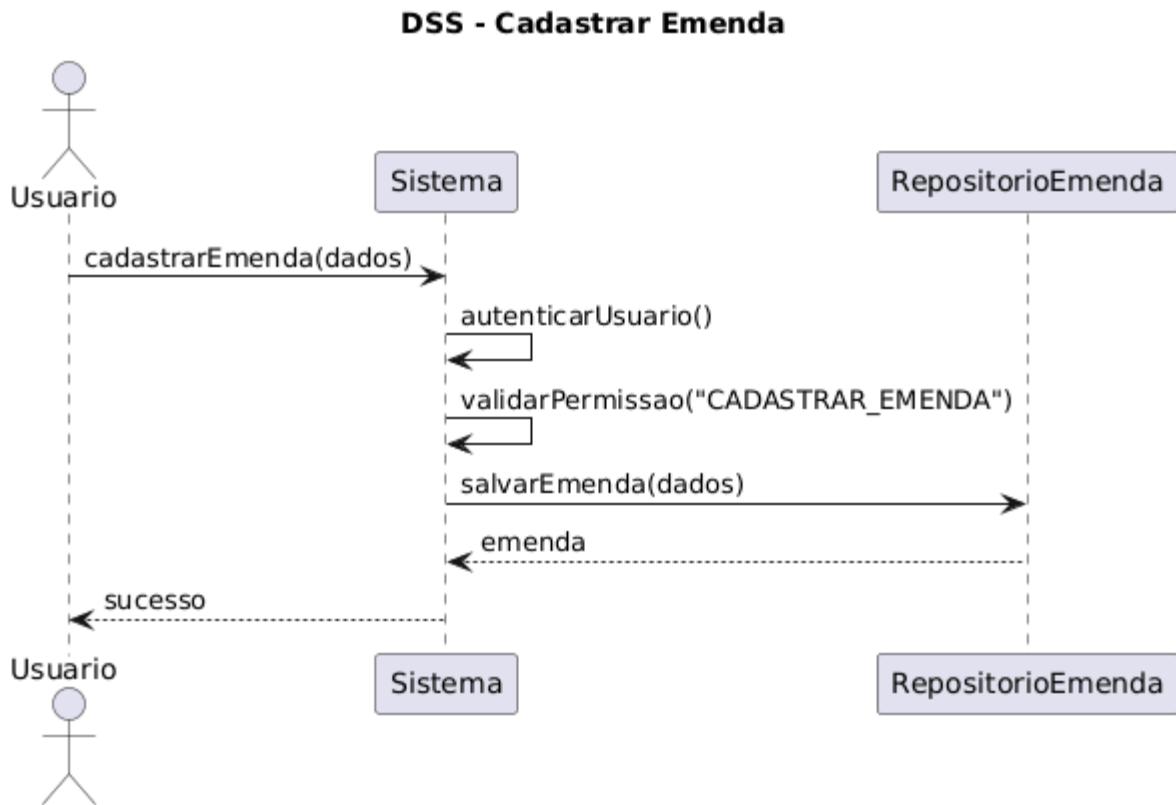


Figura 13 - DSS: Cadastrar Emenda

Operação	cadastrarEmenda(dados)
Referências Cruzadas	UC21 – Cadastrar Emenda
Pré-condições	Usuário autenticado Permissão CADASTRAR_EMENDA
Pós-condições	Emenda cadastrada

Tabela 19 - Contrato de Operação: Autenticar no Sistema

A Figura 14 apresenta a atualização de status de uma emenda e a Tabela 20 o contrato de operação desse fluxo. O sistema exige permissão específica, valida o novo status e registra a alteração. O DSS contempla fluxo normal e erro por transição inválida ou permissão insuficiente.

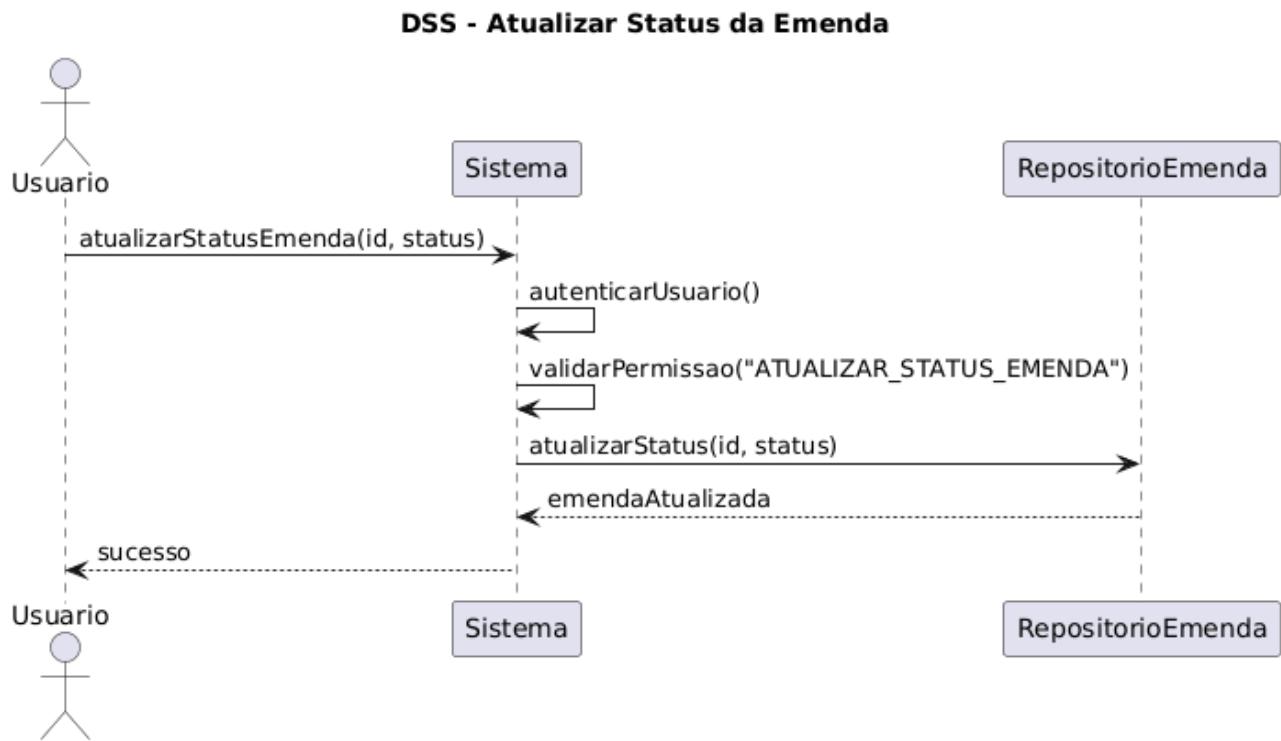


Figura 14 - DSS2: Atualizar Status da Emenda

Operação	atualizarStatusEmenda(id, status)
Referências Cruzadas	UC22 – Atualizar Status da Emenda
Pré-condições	Usuário autenticado Permissão ATUALIZAR_STATUS_EMENDA
Pós-condições	Status atualizado

Tabela 20 - Contrato de Operação: Atualizar Status Emenda

Como mostra a Figura 15, uma demanda é aberta pelo usuário. O sistema valida permissões, cria a demanda com status inicial e registra automaticamente um histórico de criação. O diagrama considera falhas como campos obrigatórios vazios e abaixo está a Tabela 21 referente ao Contrato de Operação deste fluxo..

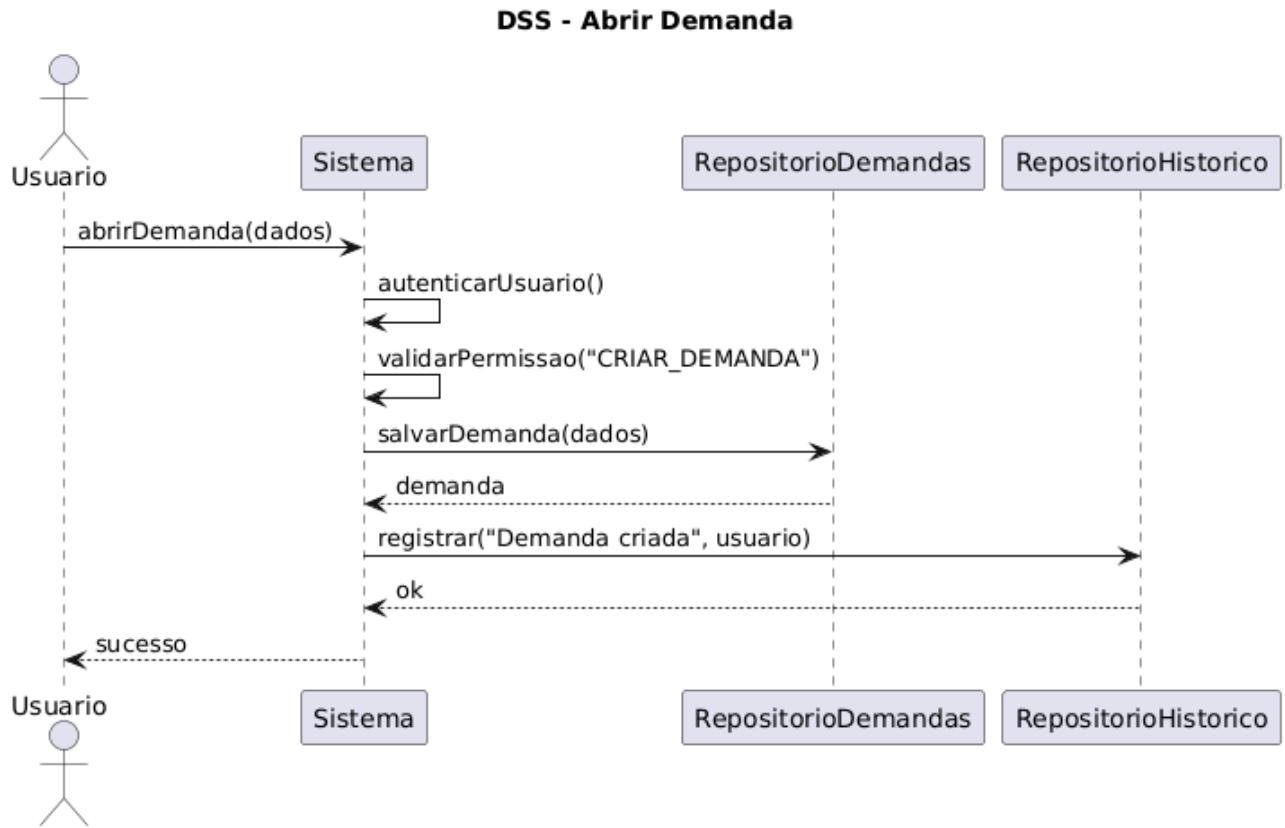


Figura 15 - DSS: Abrir Demanda

Operação	abrirDemanda(dados)
Referências Cruzadas	UC27 – Abrir Demanda
Pré-condições	Usuário autenticado Permissão CRIAR_DEMANDA
Pós-condições	Demand criada Histórico registrado

Tabela 21 - Contrato de Operação: Abrir Demanda

A Figura 16 descreve quando o usuário atribui um responsável a uma demanda. O sistema verifica permissões, atualiza a demanda e registra a mudança no histórico. O DSS mostra erros por usuário inexistente ou demanda inválida. A Tabela 22 exibe o Contrato de Operações equivalente a operação.

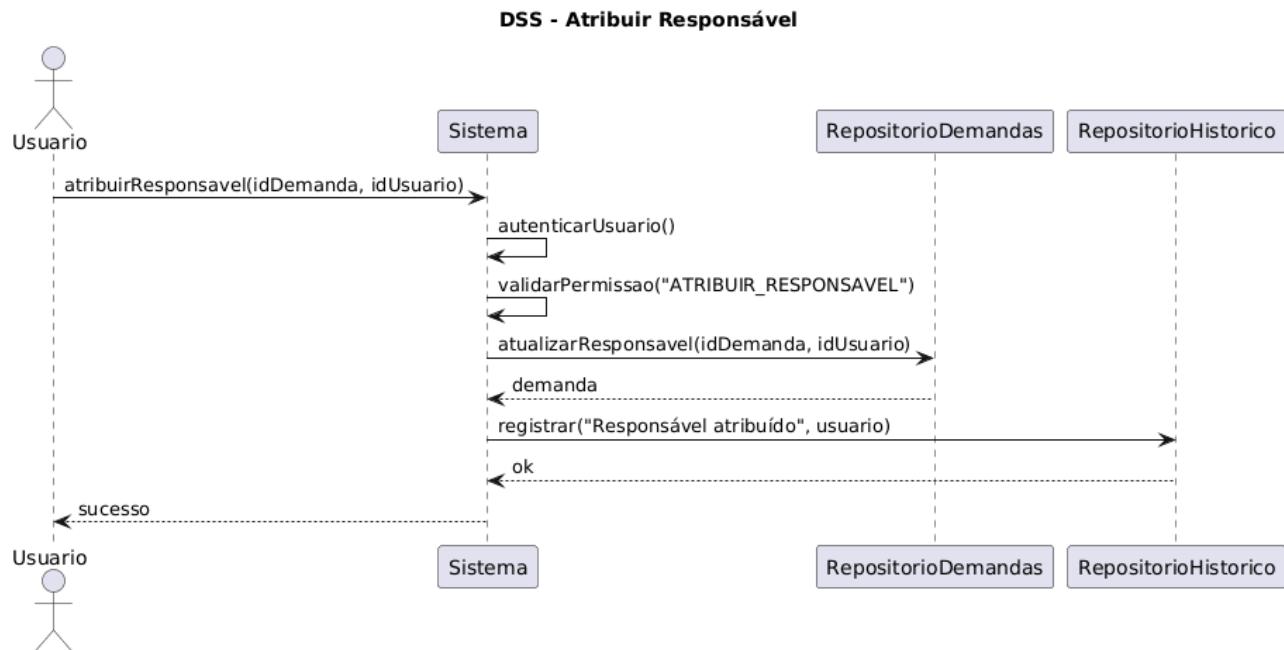


Figura 16 - DSS: Atribuir Responsável

Operação	atribuirResponsavel(id, usuario)
Referências Cruzadas	UC30 – Atribuir Responsável
Pré-condições	Usuário autenticado Permissão ATRIBUIR_RESPONSABEL
Pós-condições	Responsável definido Histórico registrado

Tabela 22 - Contrato de Operação: Atribuir Responsável

A Figura 17 demonstra o momento em que dados de uma pessoa atendida são associados a uma demanda e a Tabela 23 o contrato de operação equivalente. Após validações, a informação é gravada e registrada no histórico. O DSS contempla casos de sucesso e inconsistências nos dados da pessoa.

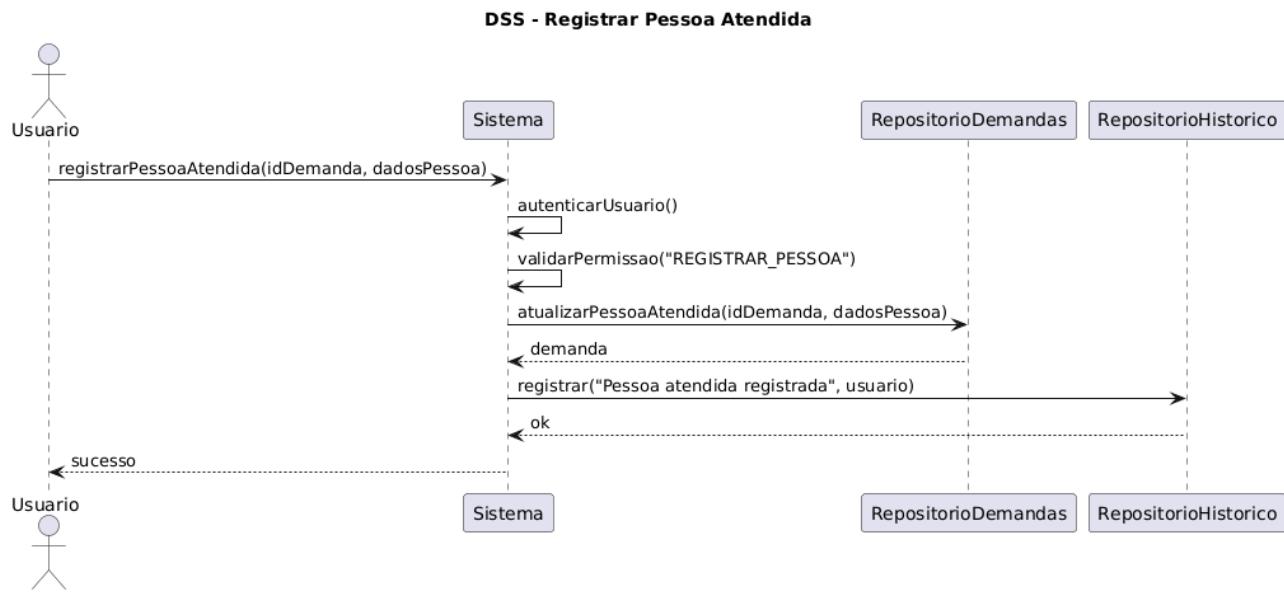


Figura 17 - DSS: Registrar Pessoa Atendida

Operação	<code>registrarPessoaAtendida(id, dadosPessoa)</code>
Referências Cruzadas	UC31 – Registrar Pessoa Atendida
Pré-condições	Usuário autenticado Permissão REGISTRAR_PESSOA
Pós-condições	Pessoa vinculada Histórico registrado

Tabela 23 - Contrato de Operação: Registrar Pessoa

Como mostra a Figura 18, o ator define um prazo para entrega ou acompanhamento da demanda. O sistema valida a data, atualiza o registro e grava histórico. O DSS ilustra condições de erro como datas inválidas ou permissão insuficiente. A Tabela 24 representa o Contrato de Operação equivalente.

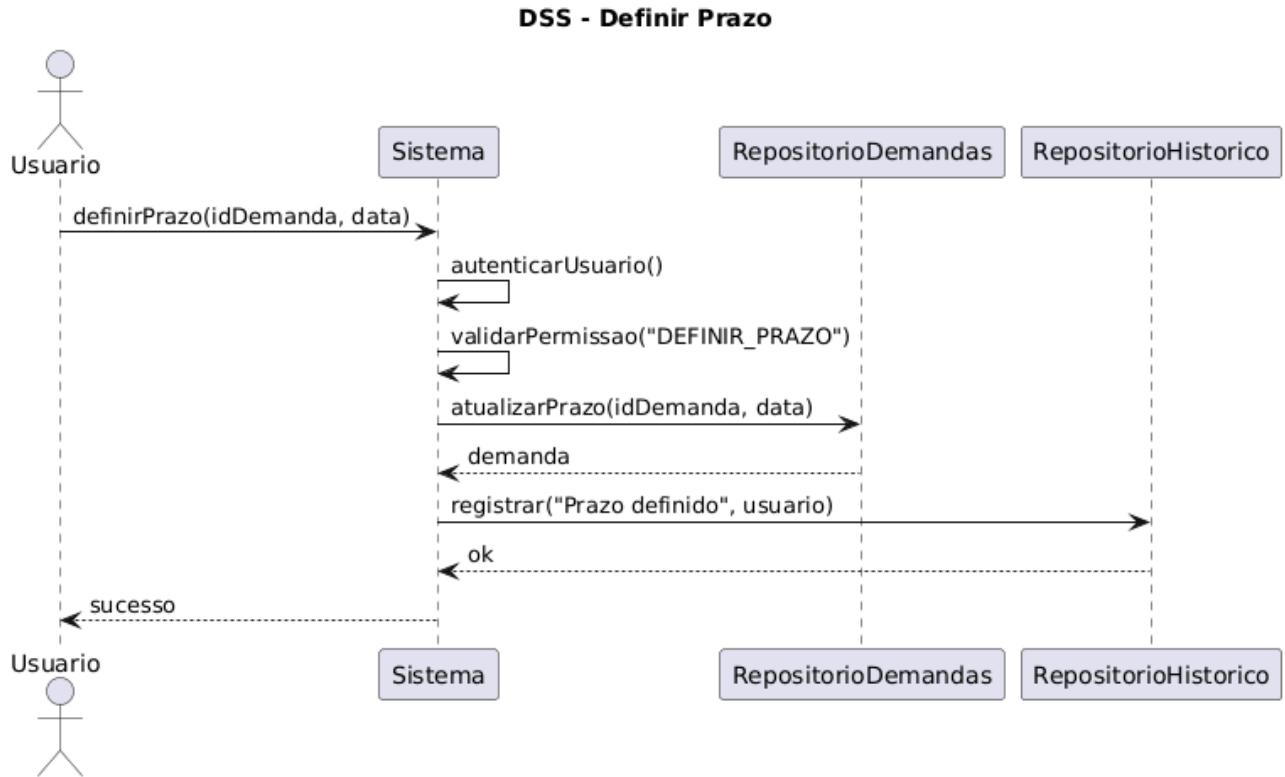


Figura 18 - DSS: Definir Prazo

Operação	definirPrazo(id, data)
Referências Cruzadas	UC32 – Definir Prazo
Pré-condições	Usuário autenticado Permissão DEFINIR_PRAZO
Pós-condições	Prazo atualizado Histórico registrado

Tabela 24 - Contrato de Operação: Autenticar no Sistema

Na Figura 19, o usuário solicita uma mudança de status e a Tabela 25 representa o Contrato de Operação equivalente. O sistema valida a transição segundo regras de negócio, aplica a alteração e registra o histórico. O DSS contempla possíveis erros por status inválido ou transição proibida.

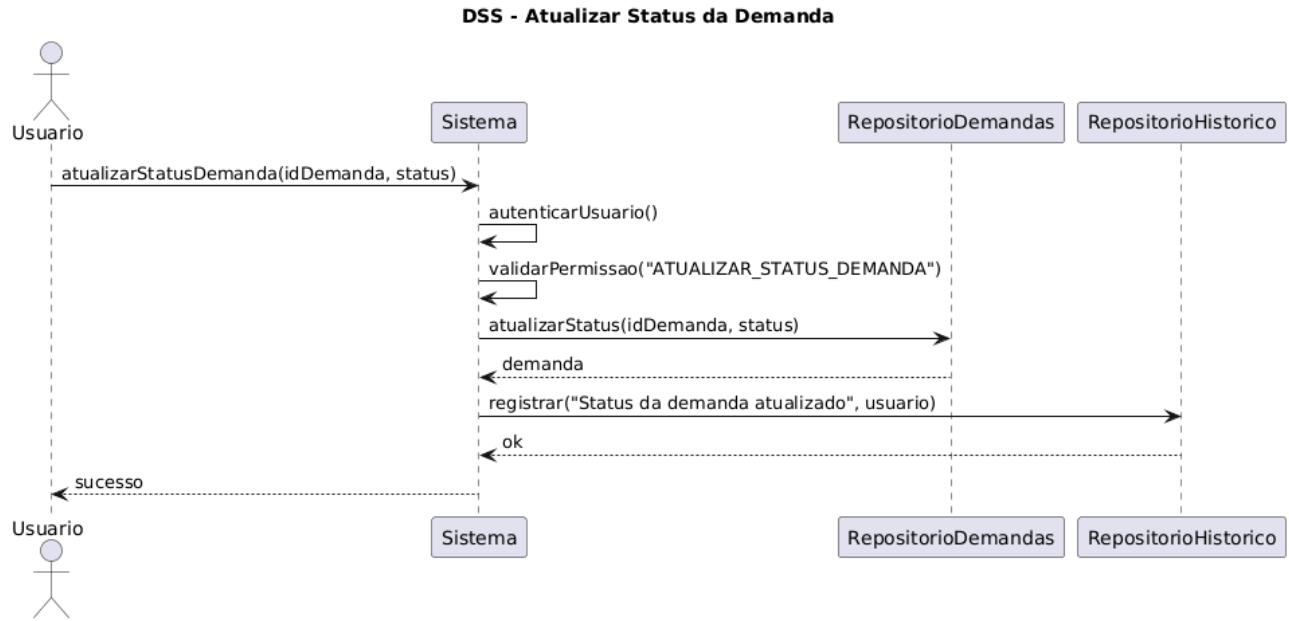


Figura 19 - DSS: Atualizar Status da Demanda

Operação	atualizarStatusDemanda(id, status)
Referências Cruzadas	UC28 – Atualizar Status da Demanda
Pré-condições	Usuário autenticado Permissão ATUALIZAR_STATUS_DEMANDA
Pós-condições	Status alterado Histórico registrado

Tabela 25 - Contrato de Operação: Atualizar Status Demanda

A Figura 20 descreve o fluxo de anexação de um ofício à demanda. O sistema autentica, valida permissões, armazena o arquivo (local ou S3), cria o registro do anexo e adiciona um item no histórico. Considera erros como formato incorreto ou falha de armazenamento. A Tabela 26 exibe o Contrato de Operação referente a esse fluxo.

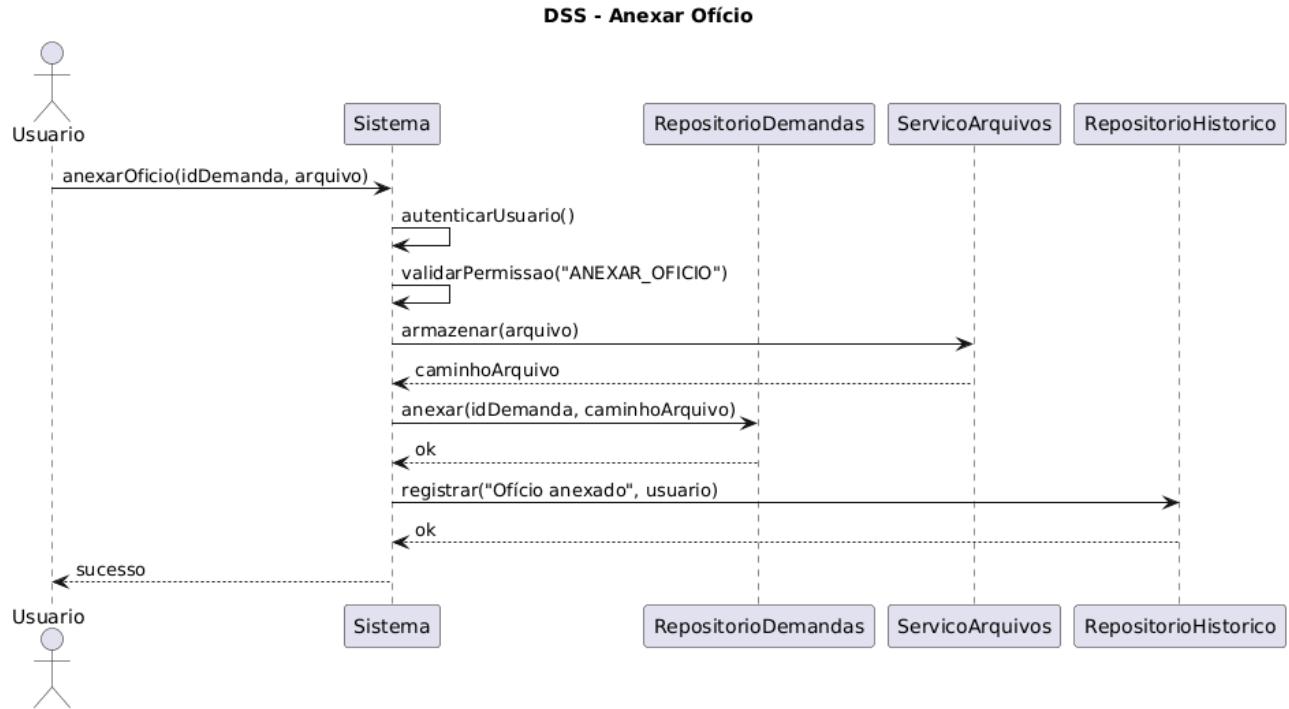


Figura 20 - DSS: Anexar Ofício

Operação	anexarOficio(id, arquivo)
Referências Cruzadas	UC27 – Anexar Ofício
Pré-condições	Usuário autenticado Permissão ANEXAR_OFICIO
Pós-condições	Arquivo salvo Registro criado Histórico registrado

Tabela 26 - Contrato de Operação: Anexar Ofício

A Figura 21 mostra o cadastro de um evento na agenda e a Tabela 27 o Contrato dessa operação. Após autenticação e validação de permissão, o sistema registra o evento e opcionalmente aciona regras de conflito de datas. O DSS prevê erros de datas inconsistentes ou permissões insuficientes.

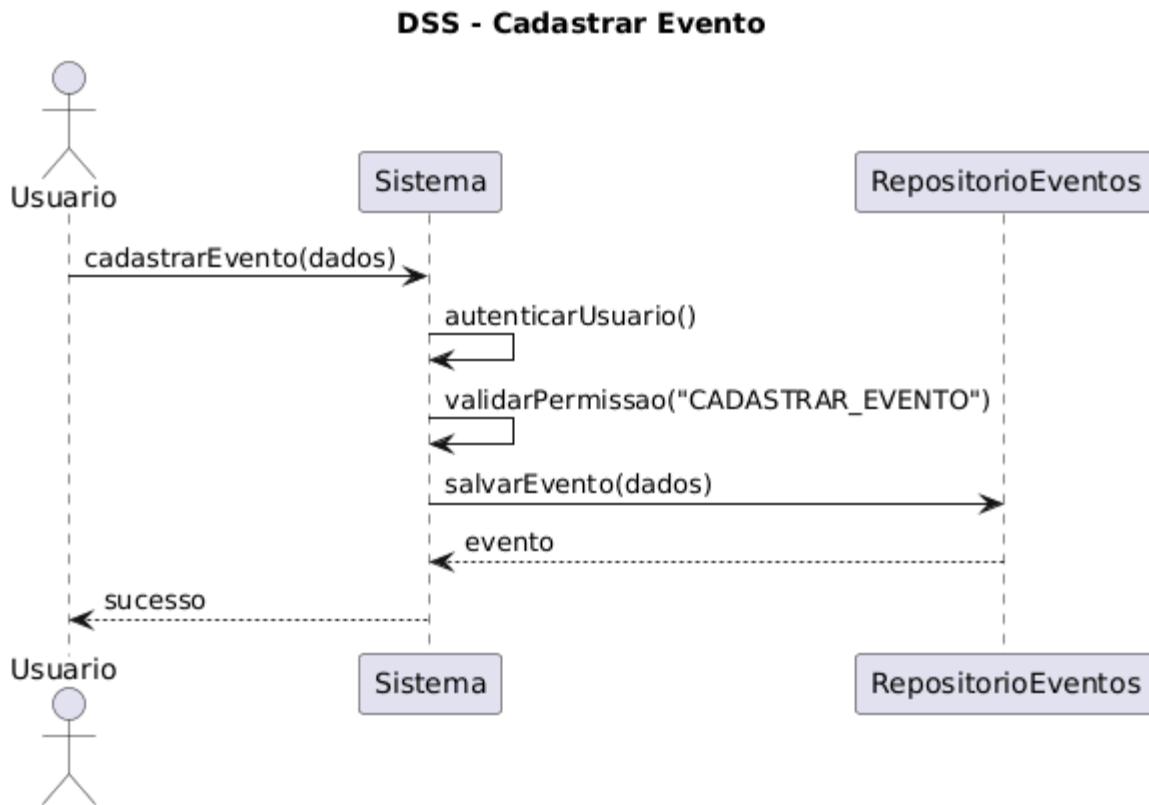


Figura 21 - DSS: Cadastrar Evento

Operação	cadastrarEvento(dados)
Referências Cruzadas	UC33 – Cadastrar Evento
Pré-condições	Usuário autenticado Permissão CADASTRAR_EVENTO
Pós-condições	Evento criado

Tabela 27 - Contrato de Operação: Cadastrar Evento

Como mostra a Figura 22, o usuário associa um alerta a um evento existente. O sistema valida o evento e permissões, registra o alerta e aciona o serviço de notificação para agendá-lo. O DSS contempla falhas como ID de evento inexistente ou horário inválido. A Tabela 28 representa o Contrato de Operação para o diagrama.

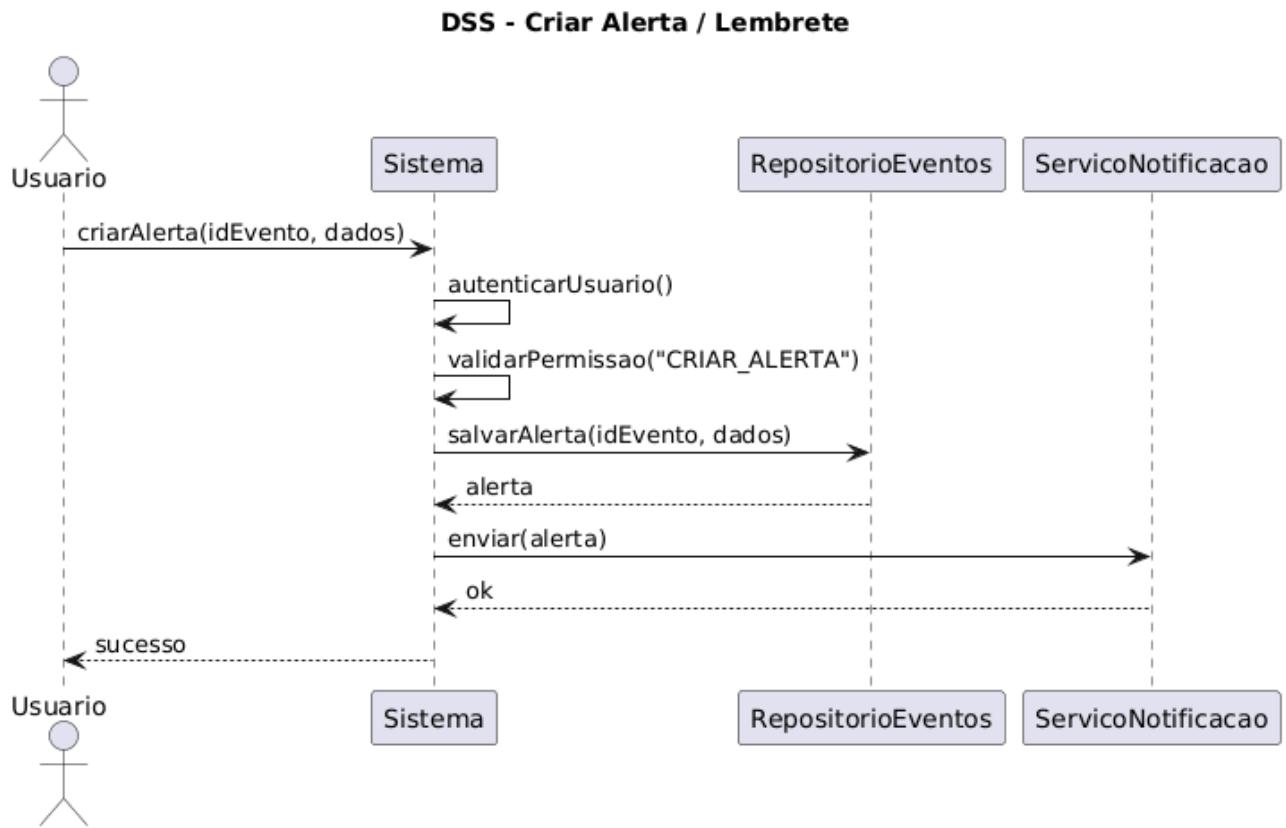


Figura 22 - DSS: Criar Alerta / Lembrete

Operação	<code>criarAlerta(idEvento, dados)</code>
Referências Cruzadas	UC35 – Criar Alerta
Pré-condições	Usuário autenticado Permissão CRIAR_ALERTA
Pós-condições	Alerta criado Notificação agendada

Tabela 28 - Contrato de Operação: Criar Alerta

3. Modelos de Projeto

Os modelos de projeto detalham a estrutura lógica, arquitetural e funcional do sistema, servindo como base para a implementação e documentação técnica. Nesta seção, são apresentados diagramas UML e C4 que descrevem o comportamento interno do sistema, suas camadas, componentes, fluxos de informação e infraestrutura de implantação.

3.1 Diagrama de Classes

A Figura 23 apresenta o diagrama de classes do sistema Gabinete Virtual, organizado em pacotes para evidenciar a divisão lógica das responsabilidades. O pacote Autorização contém PerfilAcesso e Permissao, que estruturam o modelo de controle de acesso, permitindo verificar dinamicamente se cada usuário possui privilégios específicos.

No pacote Pessoas está a classe Usuario, que representa os integrantes do gabinete e centraliza métodos de autenticação e validação de permissões. As classes Lideranca e Instituicao também fazem parte desse pacote, mantendo informações de entidades externas e compartilhando vínculo com Municipio. O pacote Demandas inclui a classe Demanda, responsável por representar todo o ciclo de atendimento, e HistoricoDemanda, que registra alterações de status, responsáveis e prazos, garantindo rastreabilidade. Cada demanda também se relaciona com Municipio, reforçando a natureza territorial do sistema. O pacote Cidades contém a classe Municipio, utilizada em diversos módulos, enquanto o pacote Agenda reúne Evento e VinculoContexto, permitindo associar compromissos a outros elementos do sistema. O pacote Emendas inclui a classe Emenda, que representa o fluxo administrativo das emendas parlamentares, e o pacote CMS contém a classe Noticia, vinculada ao usuário responsável pela publicação.

Por fim, os enums PerfilEnum, StatusDemanda, Prioridade e StatusEmenda padronizam valores do domínio e reforçam a consistência das regras do sistema.

O diagrama, portanto, resume de forma clara a estrutura conceitual da aplicação, seus principais domínios e as relações necessárias para sustentar as operações modeladas nos casos de uso.

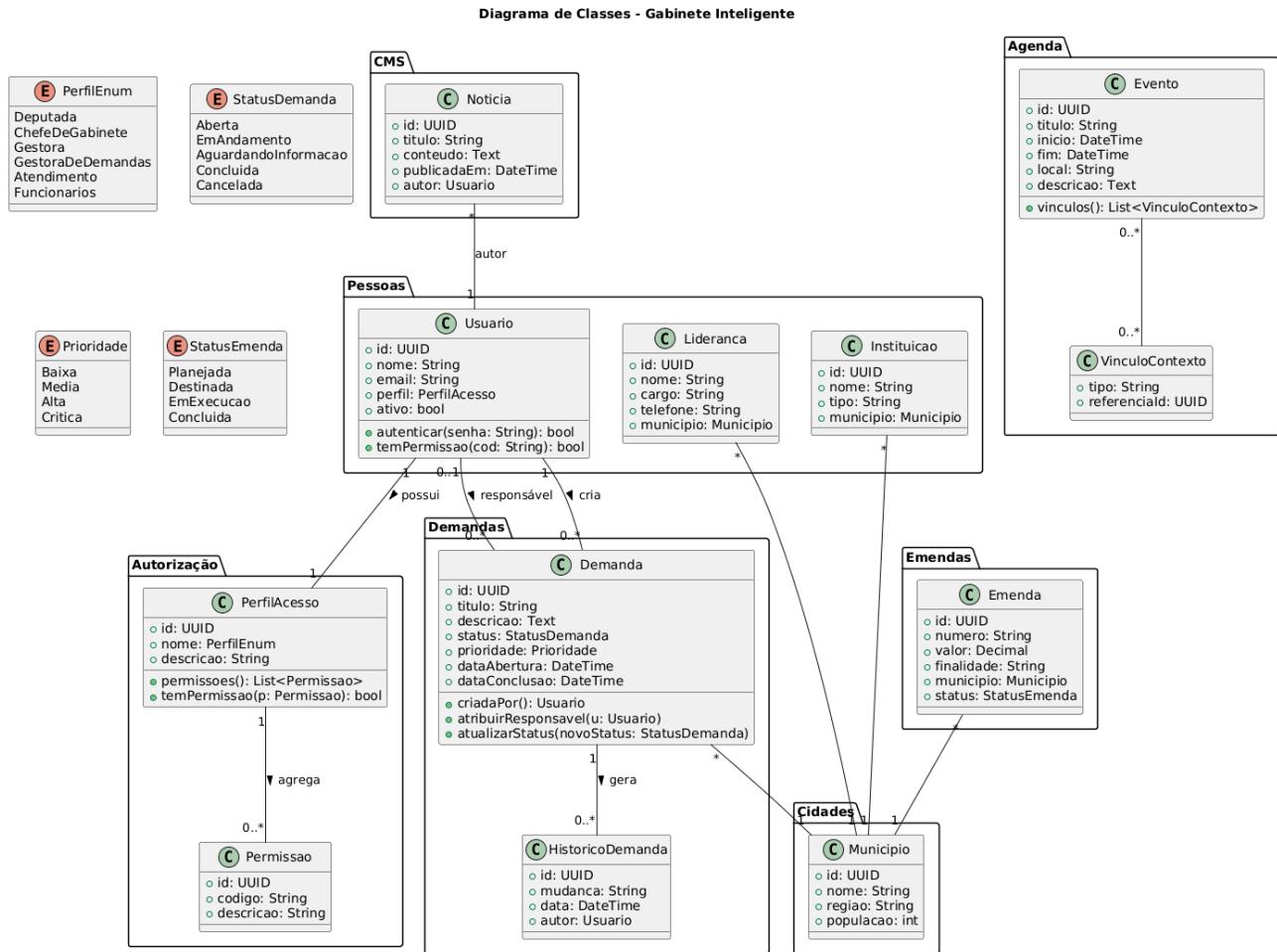


Figura 23 - Diagrama de Classes do Sistema Gabinete Virtual

3.2 Diagramas de Sequência

A presente subseção descreve os Diagramas de Sequência elaborados para cada uma das operações do sistema no contexto do projeto Gabinete Virtual. Esses diagramas têm por objetivo detalhar, de forma clara e sistemática, o fluxo de mensagens trocadas entre as diferentes camadas da aplicação: *Frontend* (React), Controladores (API Laravel), Serviços, Repositórios/Modelos, além de serviços auxiliares, como validações, autenticação, registro de histórico e armazenamento de arquivos.

O diagrama da Figura 24 representa o fluxo do UC01 - Autenticar no Sistema + Validar Permissão, descrevendo como o usuário realiza *login* e como o sistema valida credenciais, perfil e permissões iniciais. O processo inicia no *Frontend* (React) quando o usuário submete seu e-mail e senha. O

AuthController recebe a requisição e repassa para o AuthService, que valida credenciais e consulta o registro do usuário no modelo Usuario. Caso a senha esteja incorreta ou o usuário esteja inativo, o sistema retorna erro ao *Frontend*.

Após autenticar, o AuthService executa uma validação de permissões iniciais por meio do AuthorizationService, garantindo que o usuário tenha um perfil válido (como Deputada, Chefe de Gabinete, Gestora de Demandas etc.). Em caso de sucesso, o serviço gera o *token* JWT, envia ao controller e, por fim, retorna ao *frontend* para iniciar a sessão.

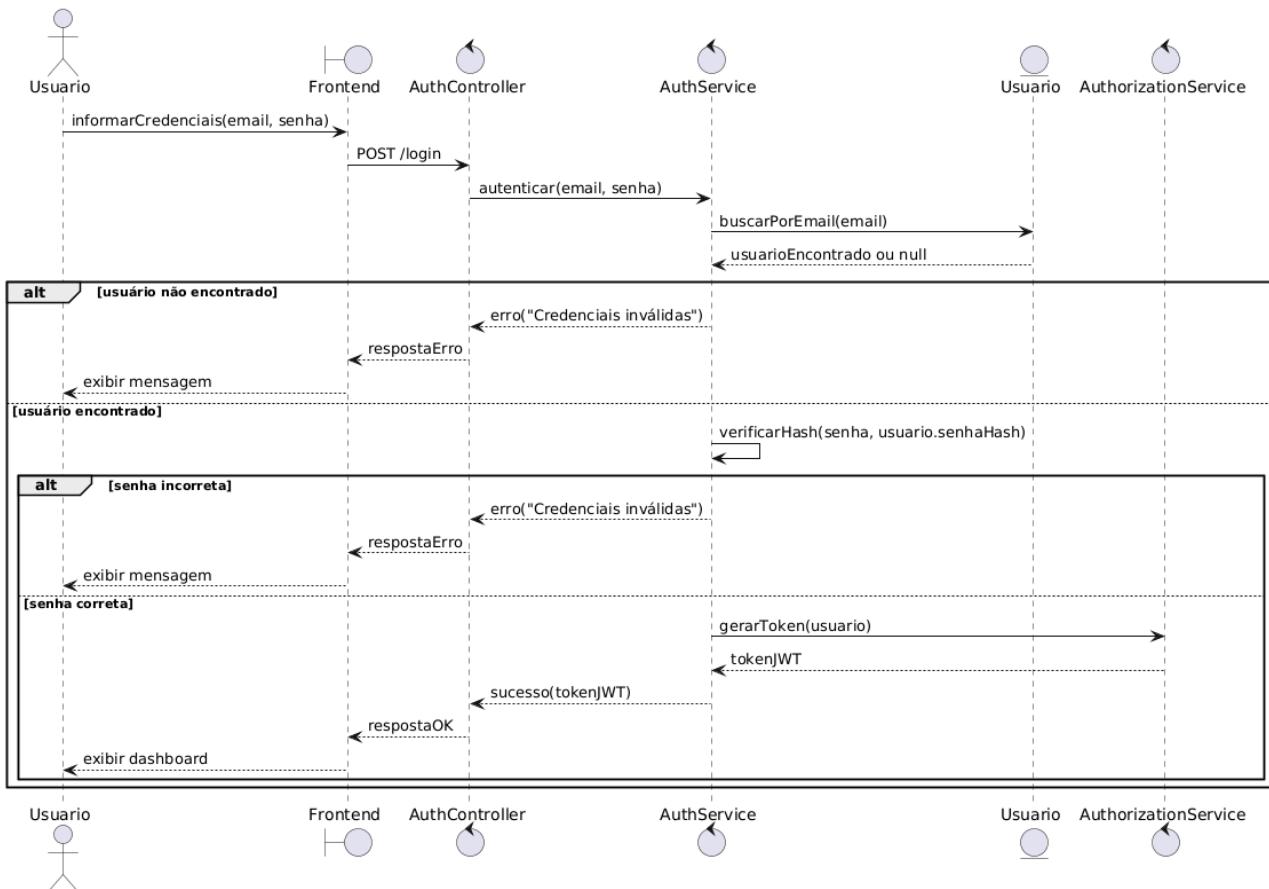


Figura 24 - Diagrama de Sequência do Sistema:Autenticar no Sistema

A Figura 25 representa um dos fluxos correspondentes ao UC02 – Gerenciar Cadastros, descrevendo como o usuário preenche dados de uma instituição no *Frontend* e como o *backend* processa, valida e persiste essas informações. O InstituicaoController recebe a requisição e chama o AuthorizationService para garantir que o perfil do usuário pode executar o cadastro.

O ValidationService valida campos, como nome, tipo e município. Em seguida, o InstituicaoService verifica previamente se o município informado existe, consultando o modelo Municipio. Caso esteja tudo correto, cria-se a nova instituição via modelo Instituição, registra-se o histórico com o HistoryService, e a operação é finalizada com resposta de sucesso ao *Frontend*.

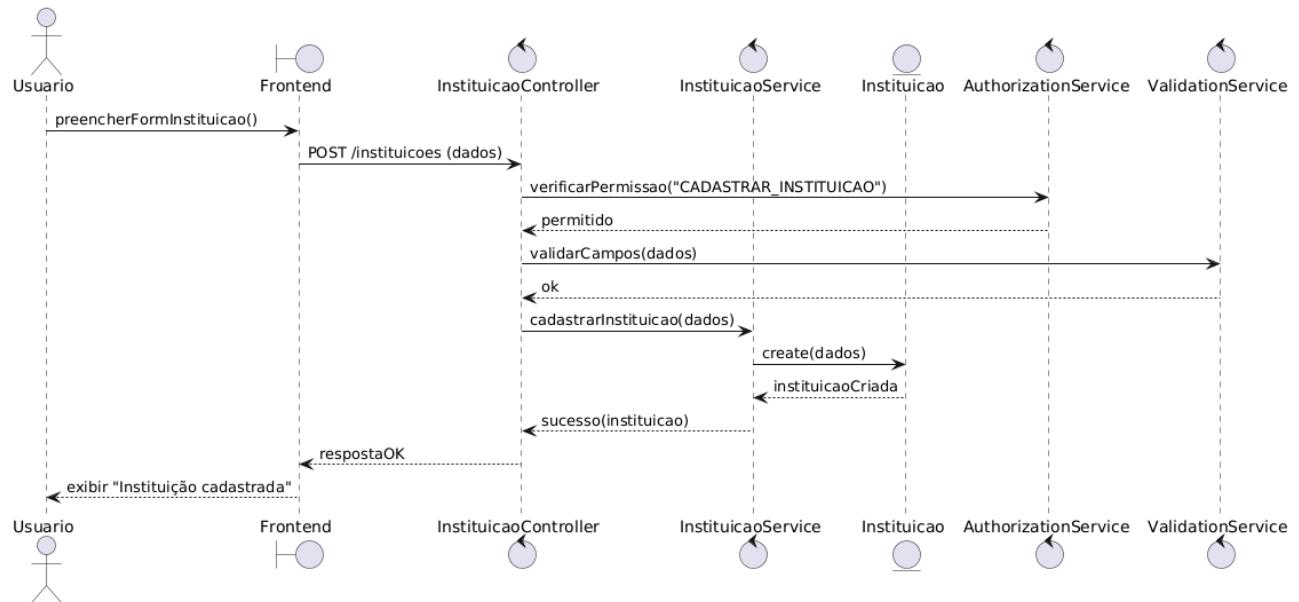


Figura 25 – Diagrama de Sequência: Cadastrar Instituição

A Figura 26 apresenta o fluxo de consulta de instituições, uma operação prevista dentro do caso de uso UC02. O usuário solicita a listagem através do *Frontend*, e o *InstituicaoController* valida permissões antes de acionar o *service*. O *InstituicaoService* aplica filtros recebidos e consulta o modelo *Instituicao*, retornando o conjunto de resultados ao *controller*, que o repassa ao usuário. Como é uma operação de leitura, não há registro de histórico.

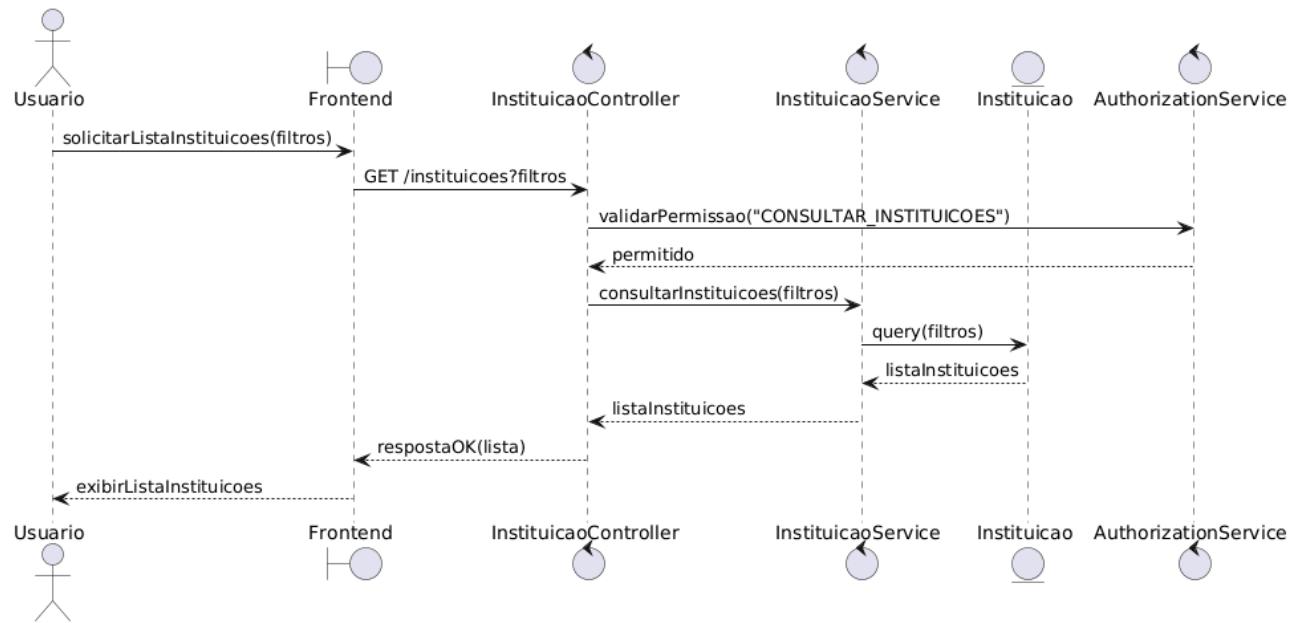


Figura 26 – Diagrama de Sequência: Consultar Instituição

Na Figura 27, é representado o fluxo de criação de uma liderança, conforme o caso de uso UC02. Após o envio dos dados pelo *Frontend*, o *controller* realiza a verificação de permissão e aciona o *ValidationService*. O *LiderancaService* valida a existência da instituição vinculada e, caso não haja inconsistências, cria o registro no modelo *Lideranca*. O *HistoryService* registra o evento e a resposta de sucesso é enviada ao usuário.

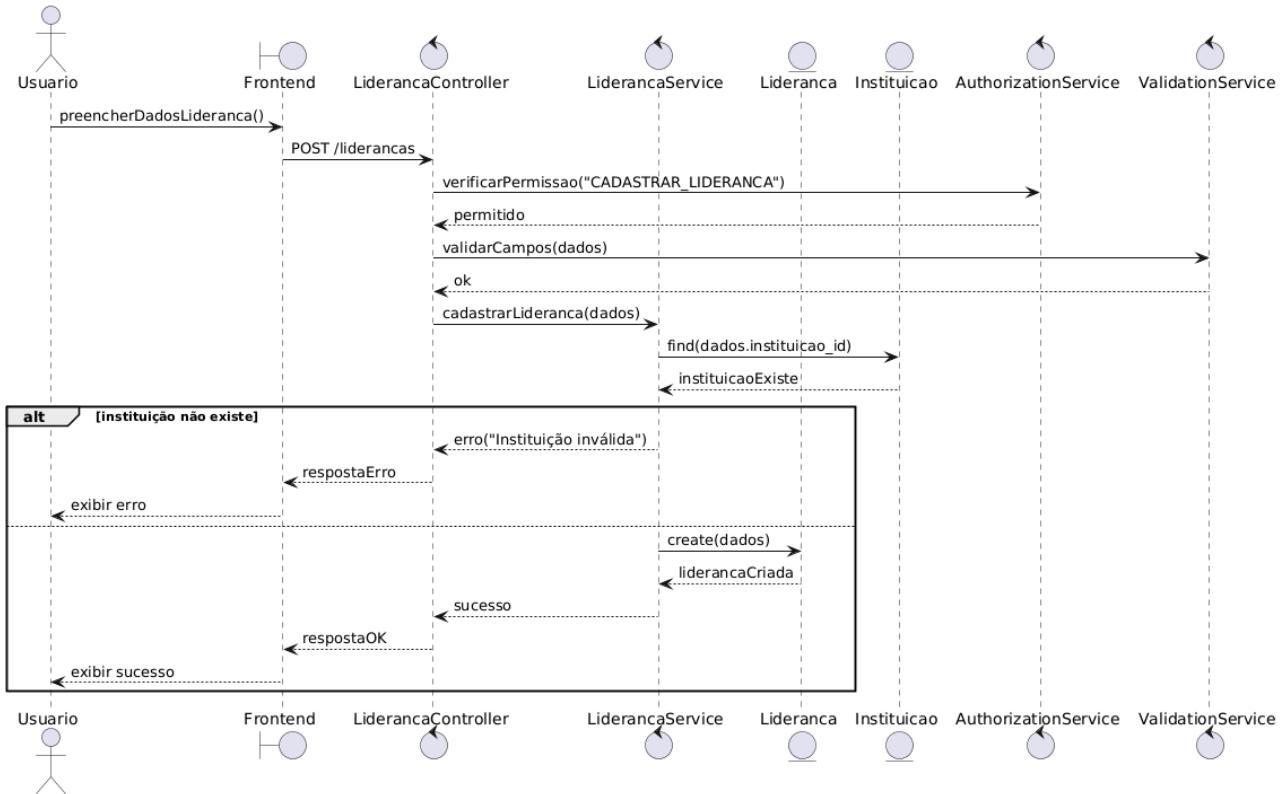


Figura 27 – Diagrama de Sequência: Cadastrar Lideranças

Como mostra a Figura 28, o diagrama de sequência de consulta de lideranças descreve o fluxo previsto no UC02. O *Frontend* solicita a listagem ao *controller*, que primeiro valida permissões e então aciona o *service*. O *LiderancaService* monta a consulta com base nos filtros informados e retorna os resultados ao *controller*, que envia a resposta ao cliente.

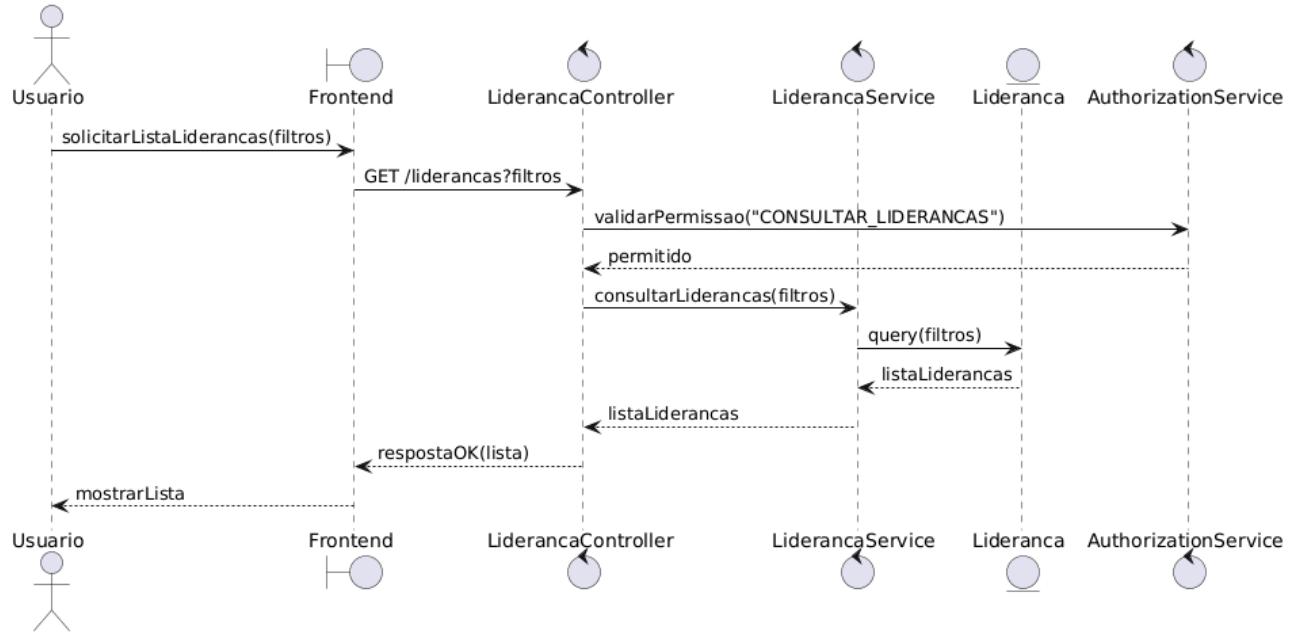


Figura 28 – Diagrama de Sequência: Consultar Lideranças

Na Figura 29, observa-se o fluxo associado ao UC02, onde o usuário requisita o cadastro de uma cidade. Após validação de permissão e integridade dos dados, o MunicipioService cria o novo registro no modelo Municipio e registra a ação no HistoryService. O sistema responde ao *Frontend* confirmando a criação.

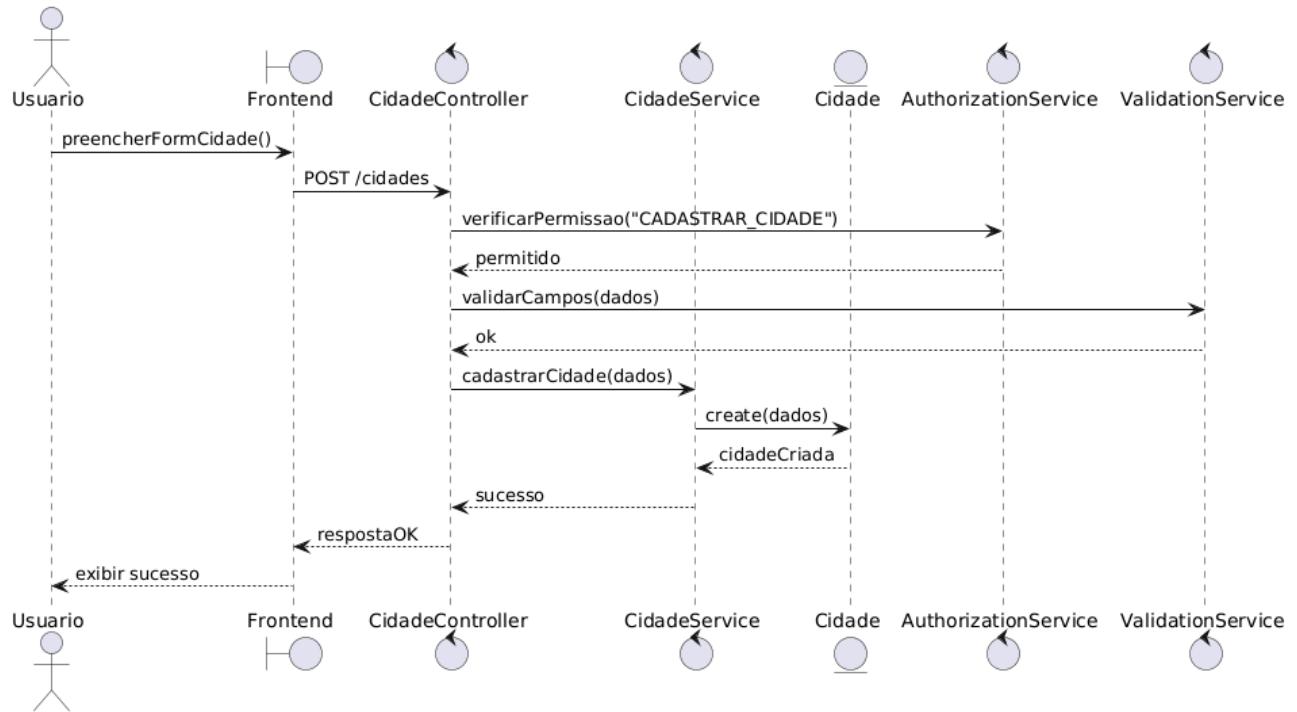


Figura 29 – Diagrama de Sequência: Cadastrar Cidades

A Figura 30 descreve a operação de consulta de cidades prevista no UC02. O *Frontend* solicita a listagem ao *controller*, que chama o *MunicipioService*. Este realiza a consulta no modelo *Municipio* e retorna a lista final ao usuário. Não há alteração de dados nem registro de histórico.

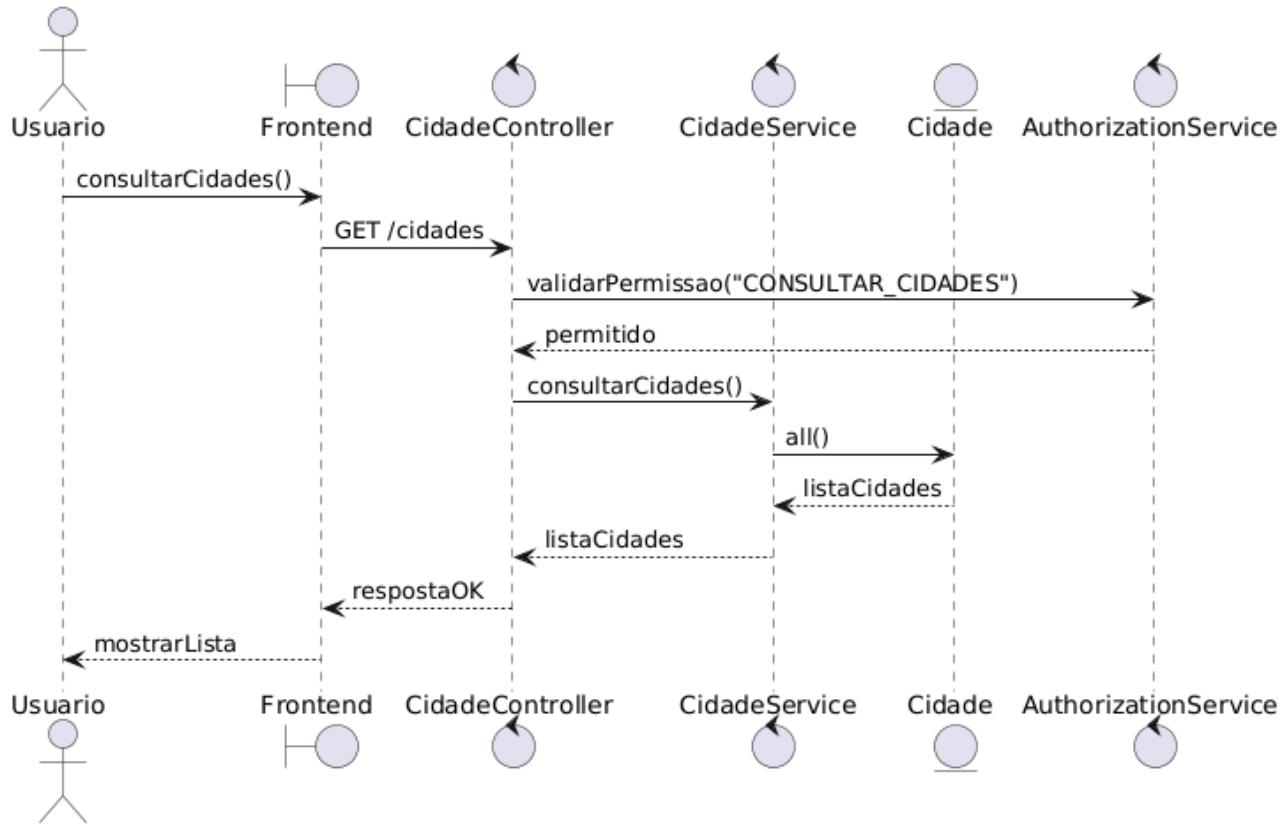


Figura 30 – Diagrama de Sequência: Consultar Cidades

Na Figura 31, o fluxo evidencia como o caso de uso UC02 trata o cadastro de uma igreja. O *controller* valida as permissões e os dados recebidos. O IgrejaService verifica o município informado e, caso seja válido, cria a nova igreja no modelo Igreja e registra o evento no HistoryService. A confirmação retorna ao usuário.

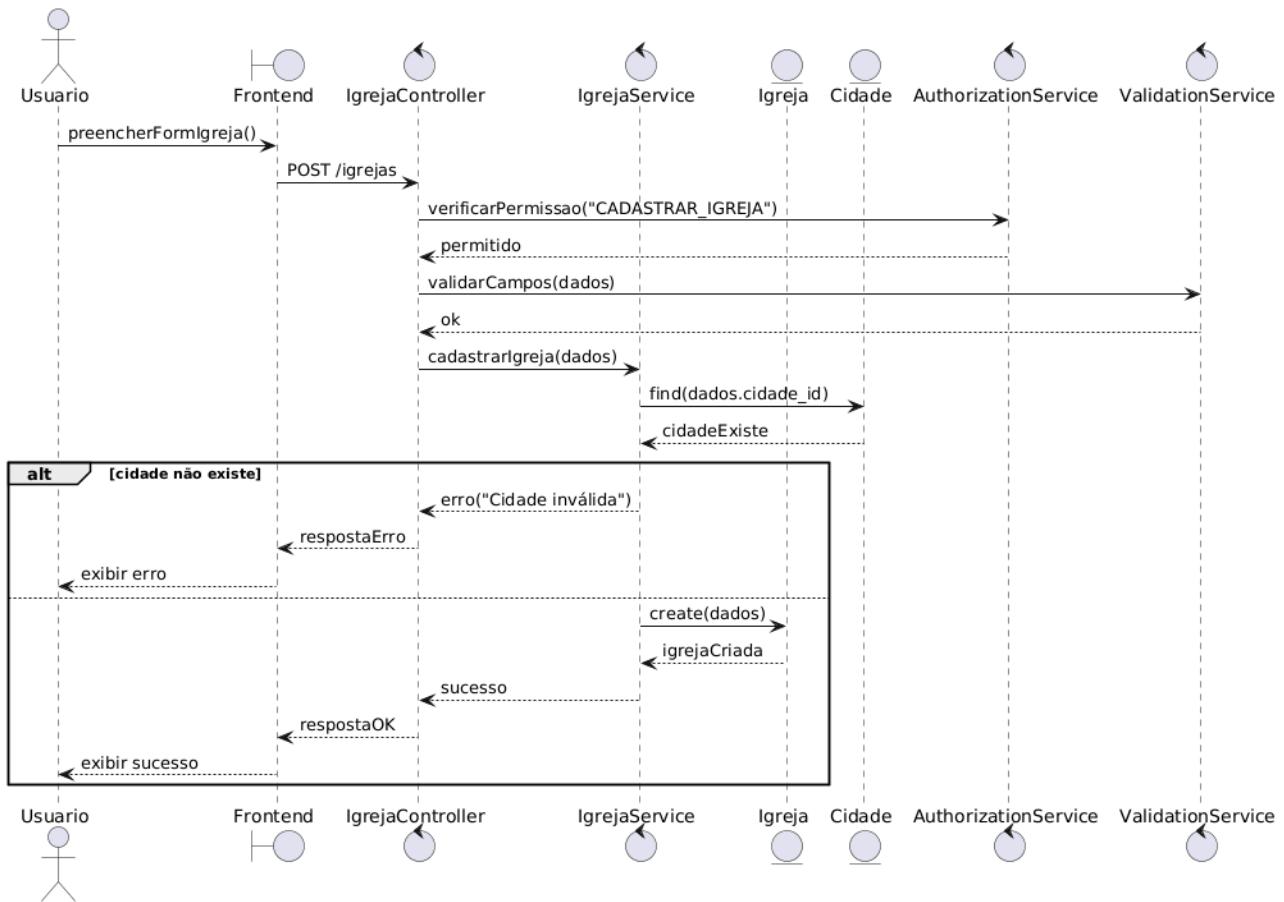


Figura 31 – Diagrama de Sequência: Cadastrar Igrejas

Como mostra a Figura 32, o diagrama detalha a operação de consulta de igrejas, presente no UC02. O *Frontend* envia os filtros ao *controller*, que repassa a requisição ao *IgrejaService*. O *service* constrói a consulta baseada nos filtros e retorna a listagem ao usuário.

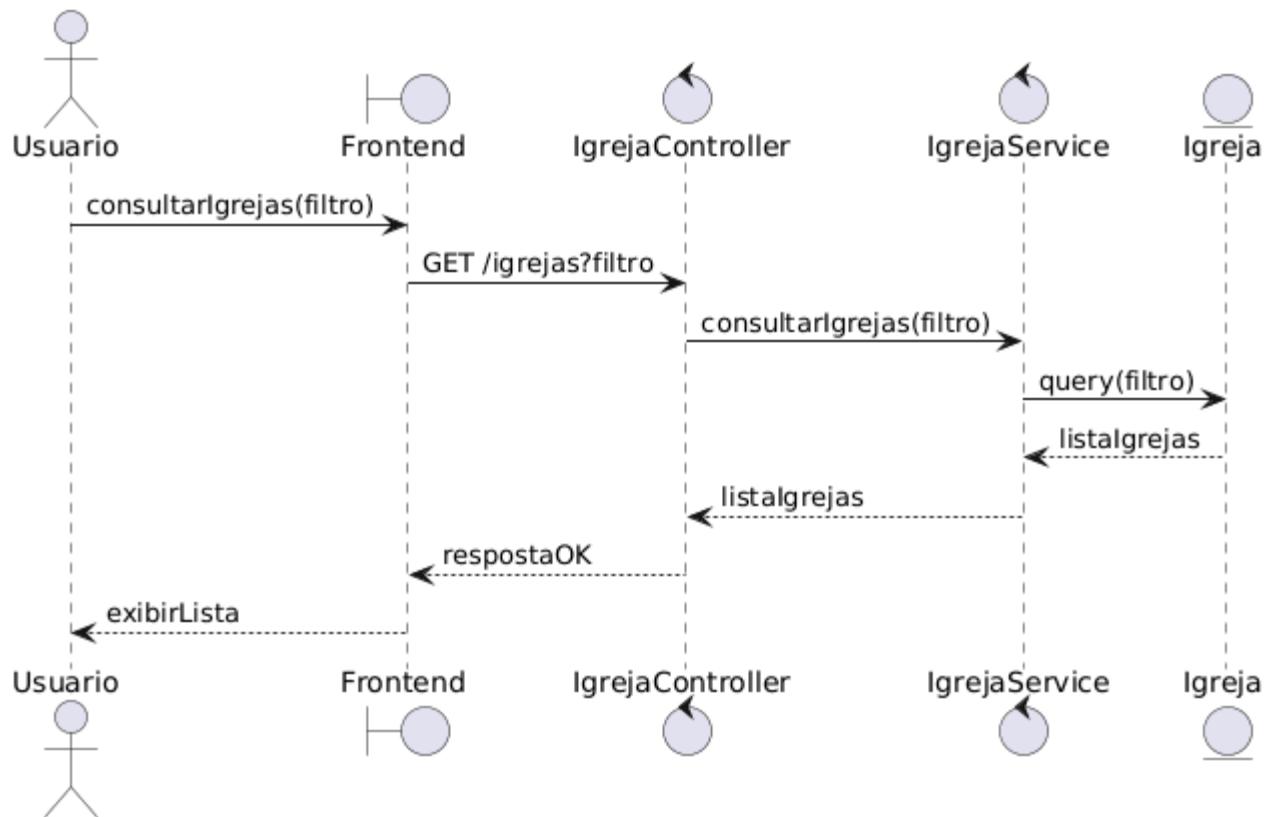


Figura 32 – Diagrama de Sequência: Consultar Igrejas

Na Figura 33, é exibido o fluxo de criação de uma notícia, conforme definido no caso de uso UC03. O usuário envia dados como título e conteúdo ao *controller*, que valida permissões e aciona o ValidationService. O NoticiaService cria o registro no modelo Noticia e registra o evento no HistoryService. A confirmação é enviada ao *Frontend*.

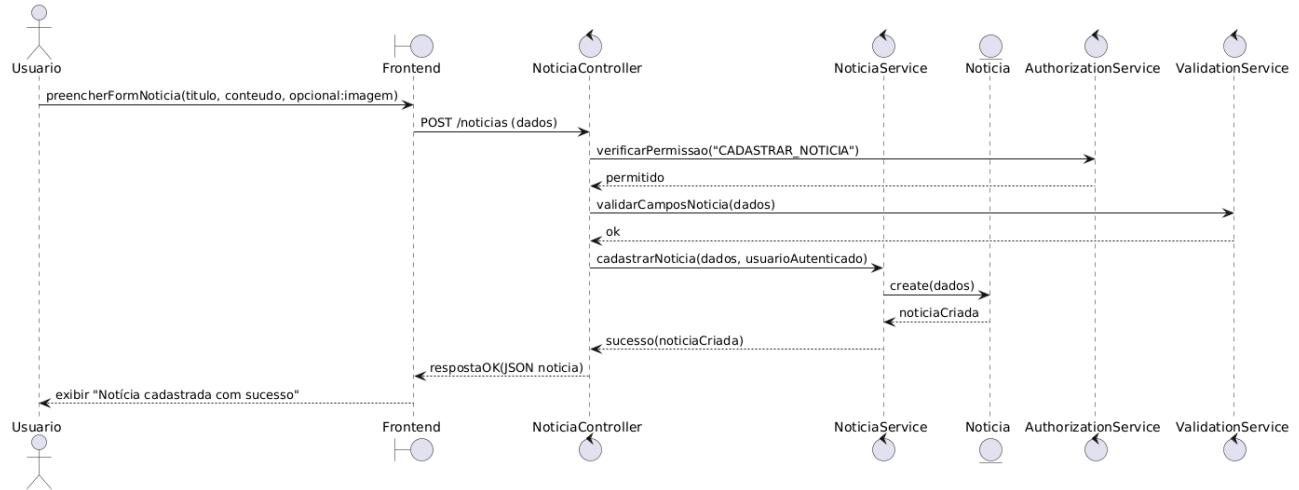


Figura 33 – Diagrama de Sequência: Publicar Notícias

Como mostra a Figura 34, o diagrama de sequência representa a operação de consulta de notícias prevista no caso de uso UC03. O usuário inicia a requisição a partir do *Frontend*, informando filtros opcionais como título, data ou autor. O *NoticiaController* recebe a solicitação, valida permissões conforme o perfil do usuário e aciona o *NoticiaService*, que constrói a consulta no modelo *Noticia* aplicando eventuais filtros. O resultado é retornado ao *controller* e enviado ao *Frontend*, que exibe a lista ao usuário. Por se tratar de uma operação de leitura, não há alterações no estado do sistema nem registro de histórico.

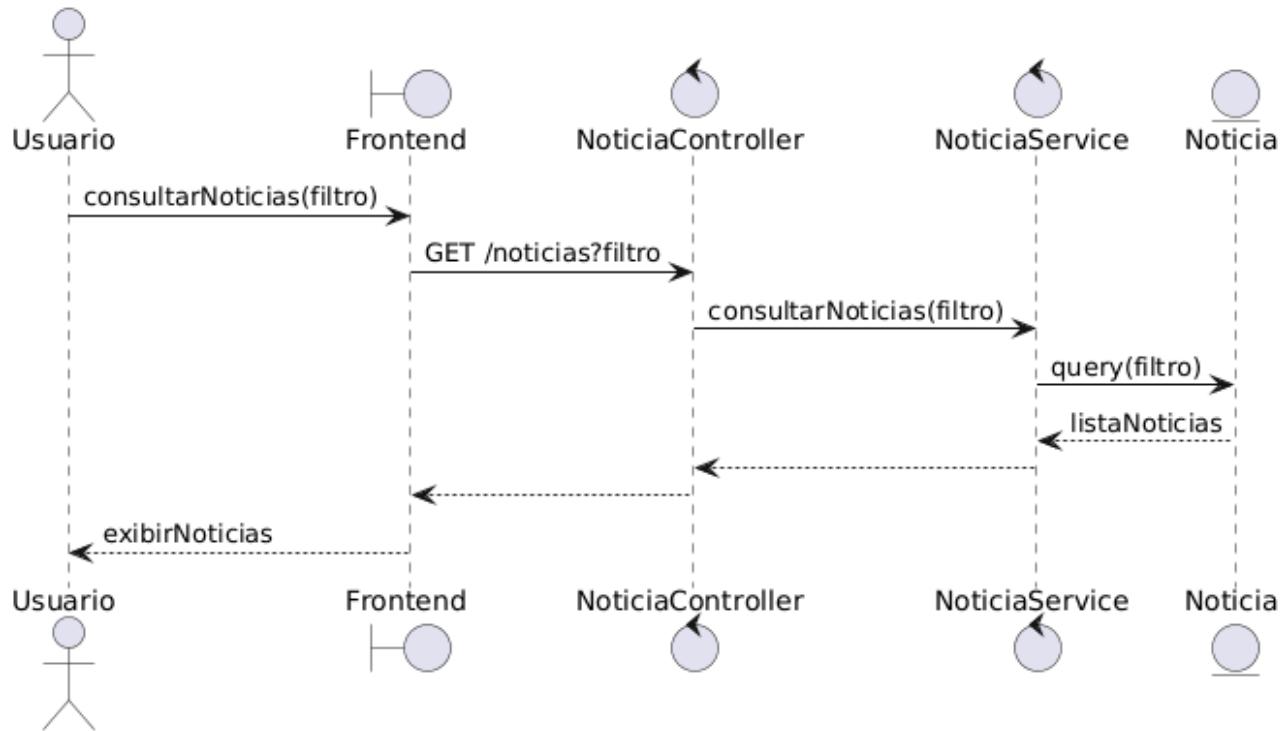


Figura 34 – Diagrama de Sequência: Consultar Notícias

Na Figura 35, o diagrama de sequência demonstra o fluxo da operação de atualização do conteúdo institucional de missão, conforme previsto no caso de uso UC03. Após o envio dos novos dados pelo *Frontend*, o *controller* valida permissões e utiliza o *ValidationService* para assegurar que o texto fornecido atende aos requisitos mínimos. O *ConteudoService* atualiza o registro no modelo correspondente e o *HistoryService* registra a alteração para auditoria. O *Frontend* recebe a confirmação de sucesso ao final do processo.

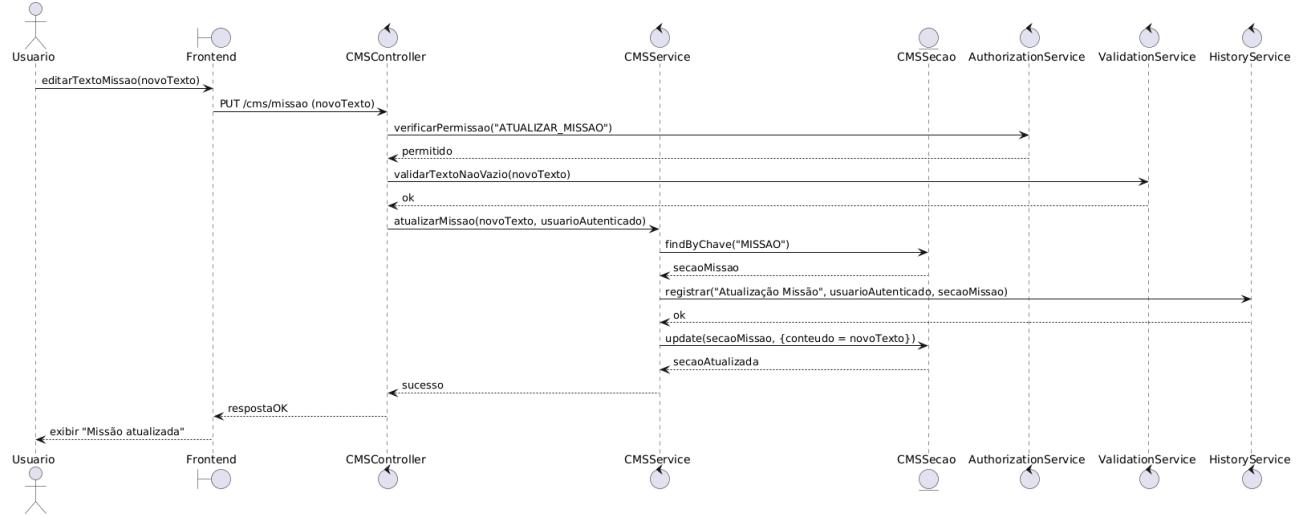


Figura 35 – Diagrama de Sequência: Atualizar Missão

A Figura 36 detalha a operação de atualização da seção de trajetória institucional, integrante do caso de uso UC03. O fluxo é iniciado pelo *Frontend* e passa pela validação de permissão, seguido pela verificação da integridade dos dados no *ValidationService*. O *ConteudoService* atualiza o registro e envia o evento ao *HistoryService*, garantindo rastreabilidade da mudança. Finalmente, o *controller* retorna o status de sucesso ao usuário.

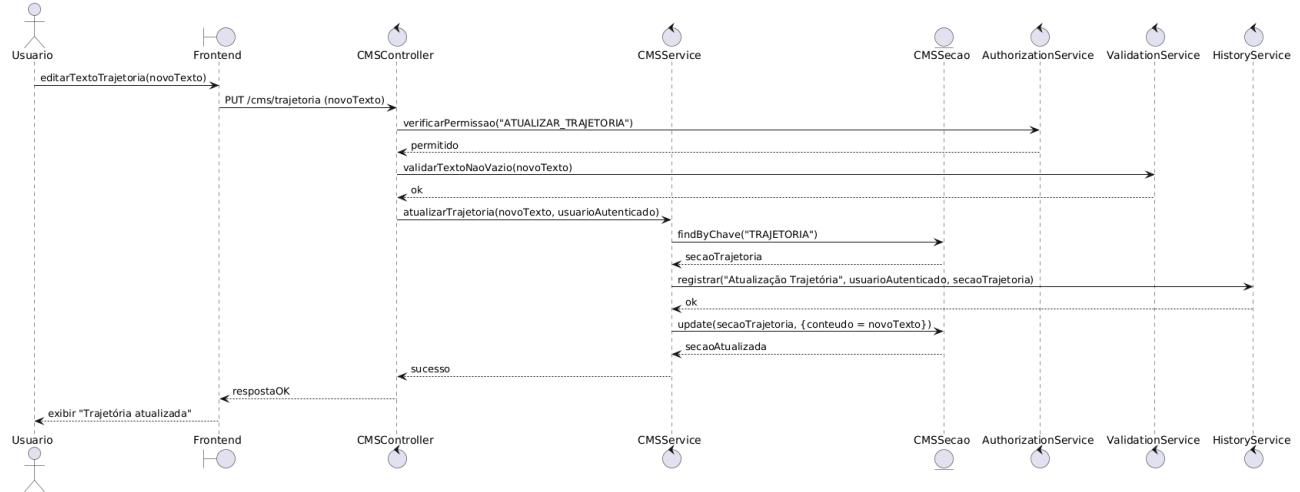


Figura 36 – Diagrama de Sequência: Atualizar Trajetória

Como mostra a Figura 37, o fluxo de criação de um projeto exibido no site faz parte do caso de uso UC03 e envolve permissão, validação e persistência. O *controller* recebe os dados enviados pelo

Frontend, valida as permissões e aciona o ValidationService. O ProjetoSiteService cria o novo registro no modelo ProjetoSite e registra o evento no HistoryService. O usuário recebe a confirmação da operação.

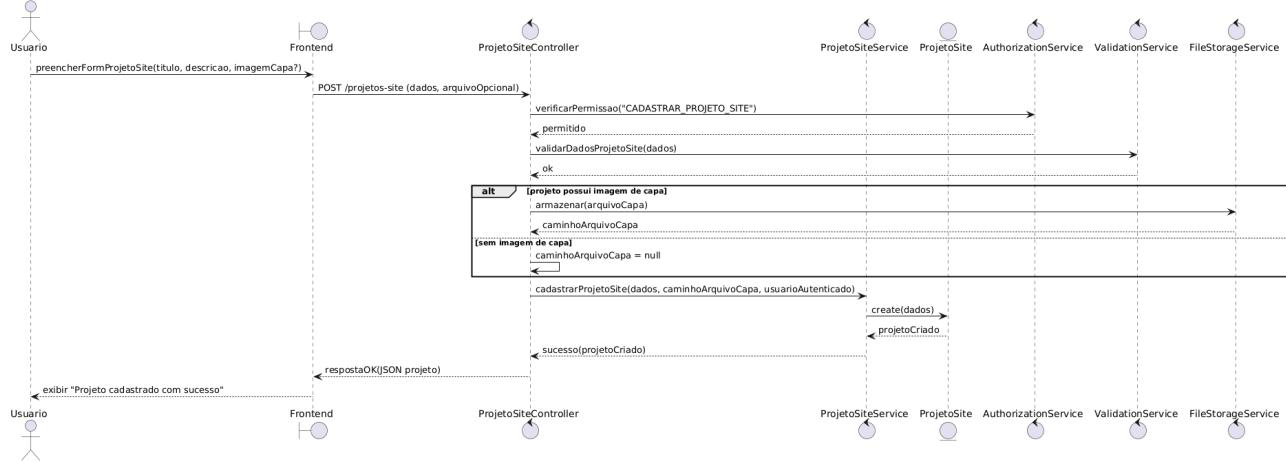


Figura 37 – Diagrama de Sequência: Cadastrar Projetos Site

Na Figura 38, o diagrama de sequência demonstra a operação de consulta de projetos exibidos no portal, pertencente ao caso de uso UC03. O *Frontend* requisita a listagem ao *controller*, que aciona o ProjetoSiteService para consultar o modelo ProjetoSite. A lista retornada é repassada ao usuário sem alterações no estado do sistema.

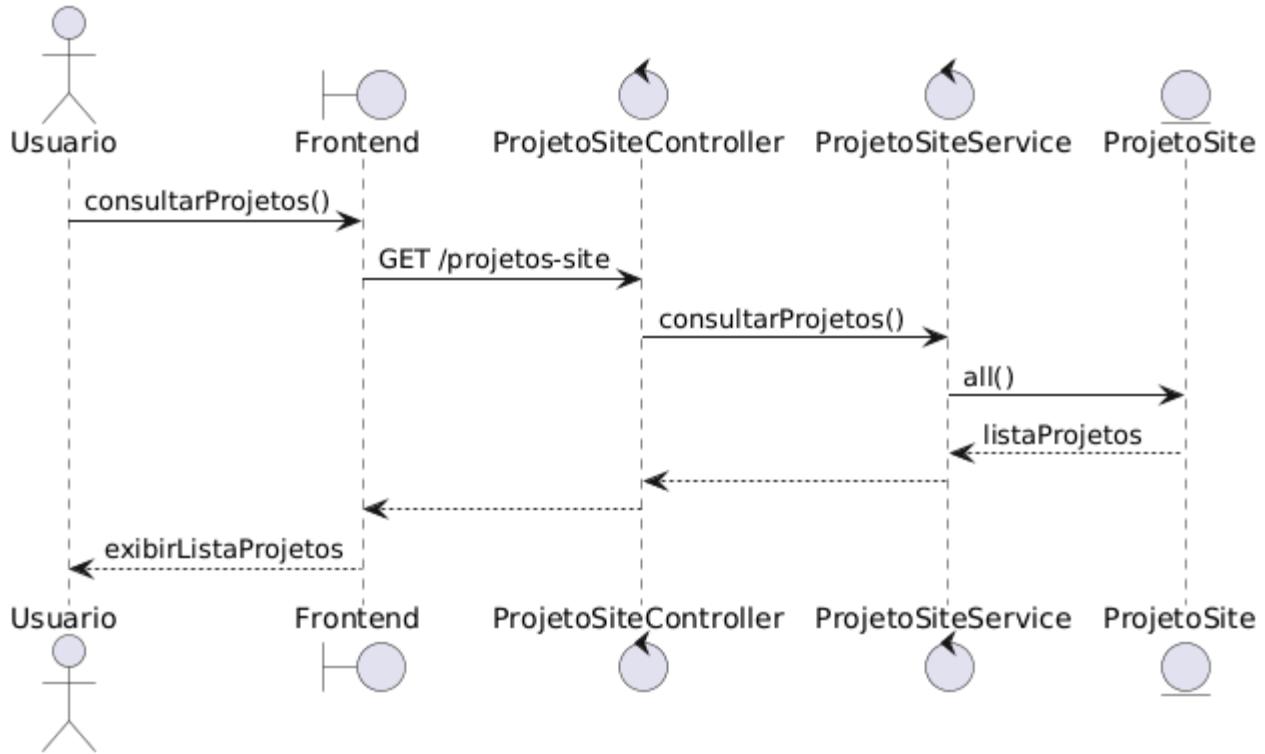


Figura 38 – Diagrama de Sequência: Consultar Projetos Site

A Figura 39 representa o fluxo de criação de um Projeto de Lei, conforme descrito no caso de uso UC04. Após validação de permissão e integridade dos dados, o ProjetoLeiService verifica se o número informado já existe. Em caso positivo, retorna erro ao usuário; caso contrário, cria o registro no modelo ProjetoLei e registra o histórico da operação. O *Frontend* recebe a resposta confirmando a criação do PL.

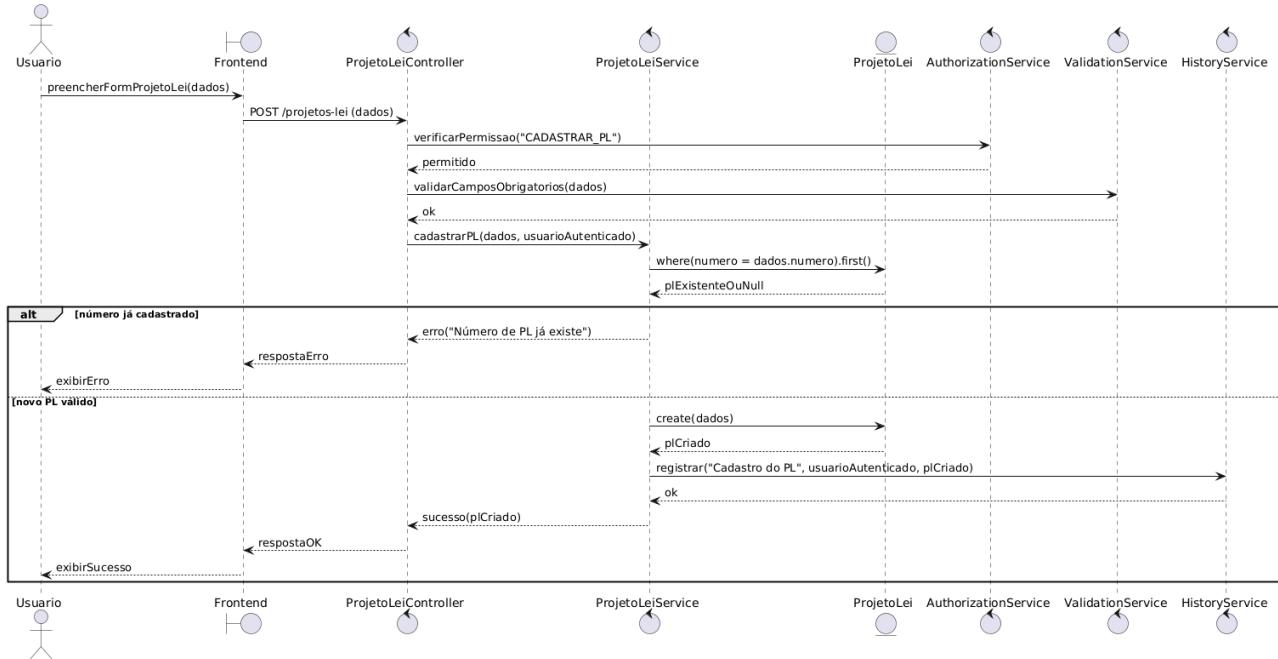


Figura 39 – Diagrama de Sequência: Cadastrar Projetos de Lei

Na Figura 40, observamos o fluxo da operação de atualização de status de um Projeto de Lei prevista no caso de uso UC04. O controller valida permissões e aciona o ValidationService para confirmar se o novo status é permitido. O ProjetoLeiService localiza o PL e atualiza o status no modelo ProjetoLei. O HistoryService registra a alteração e o resultado é enviado ao *Frontend*. Caso o PL não seja encontrado, retorna-se erro.

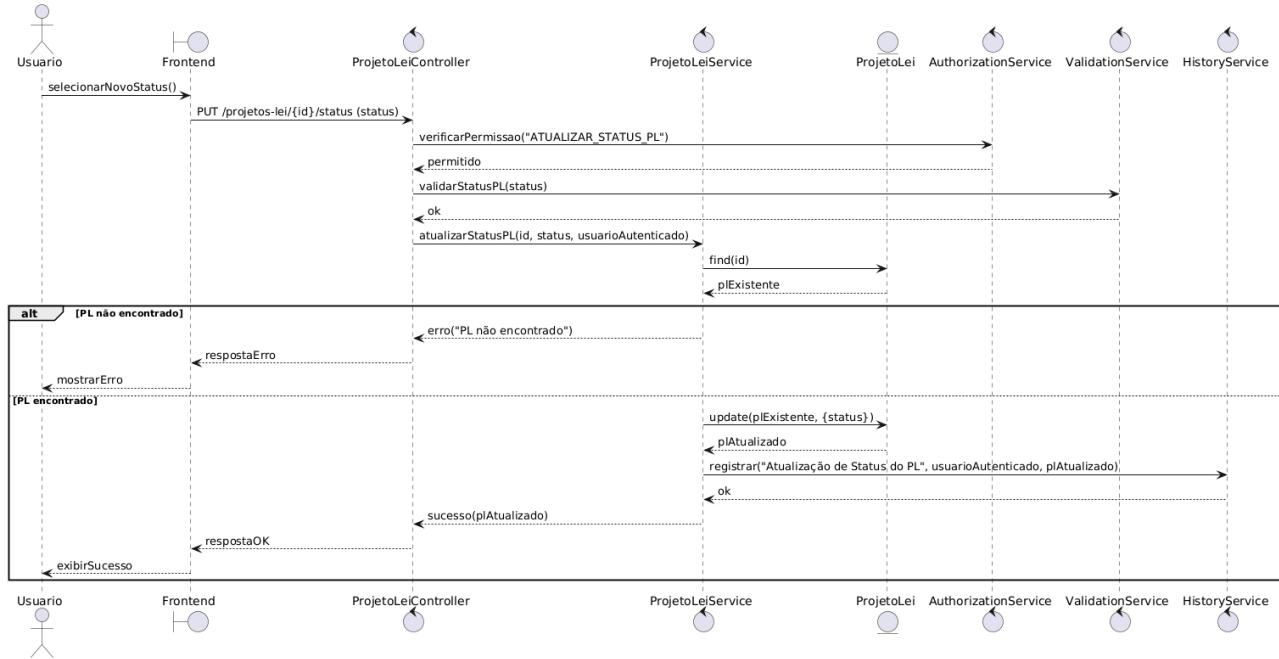


Figura 40 – Diagrama de Sequência: Atualizar Status Projeto de Lei

Como indica a Figura 41, o diagrama de sequência demonstra a operação de consulta de Projetos de Lei, parte integrante do caso de uso UC04. O usuário envia filtros através do *Frontend*; o *controller* valida permissões e aciona o *ProjetoLeiService*, que consulta o modelo *ProjetoLei* com base nos filtros. Os resultados são retornados ao usuário sem alteração de estado.

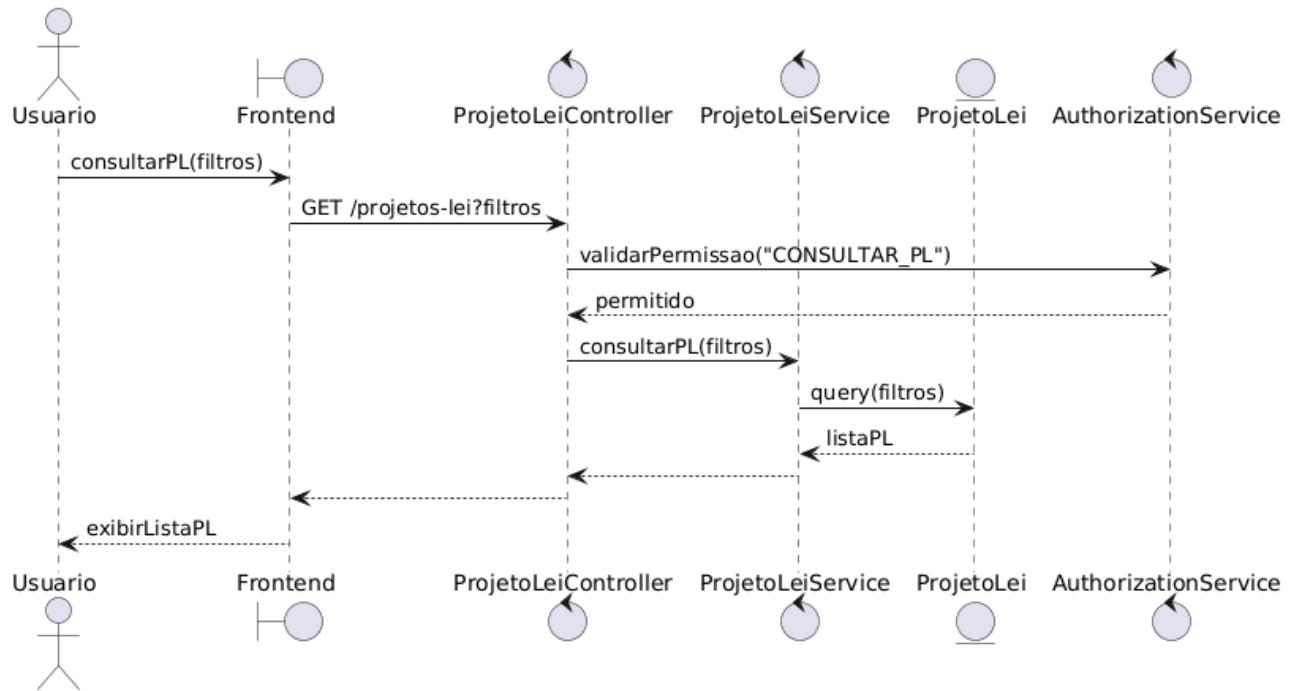


Figura 41 – Diagrama de Sequência: Consultar Projetos de Lei

Na Figura 42, o diagrama de sequência reflete o processo de cadastro de emenda previsto no caso de uso UC05. Após validação de permissão e integridade, o EmendaService confirma a existência da cidade associada e cria o registro no modelo Emenda. O HistoryService registra o evento e uma resposta de sucesso é enviada ao usuário.

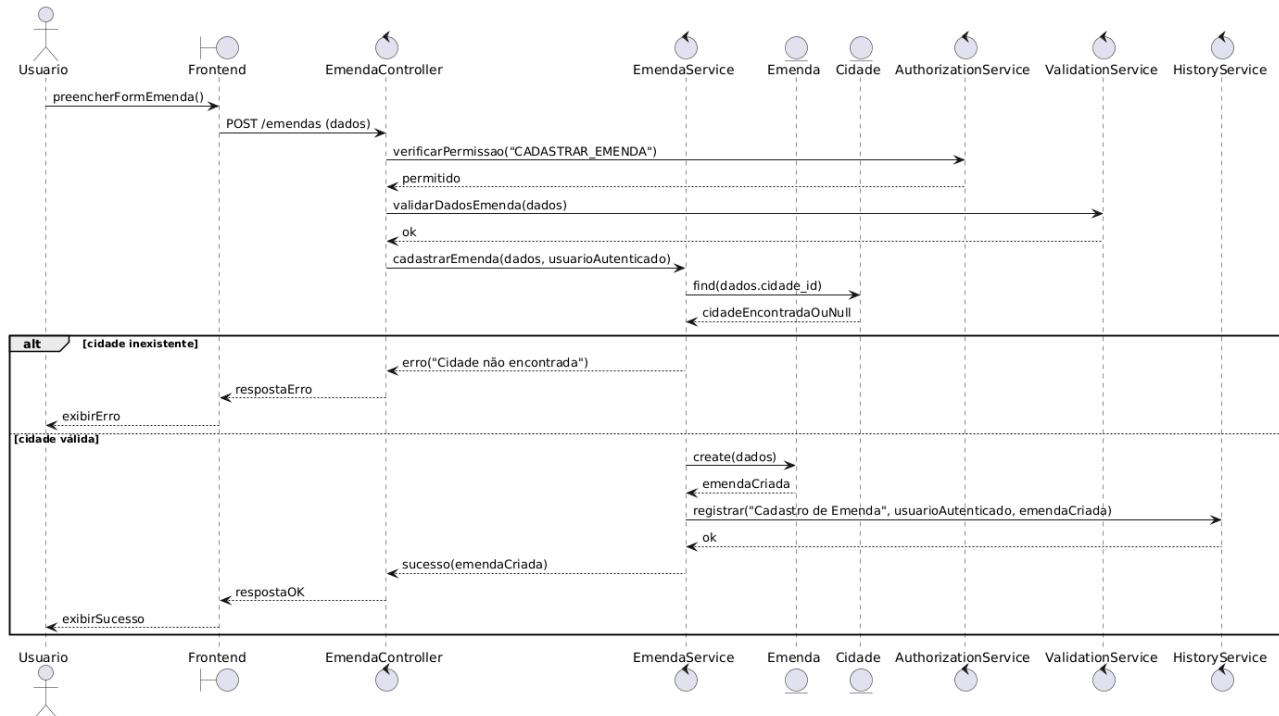


Figura 42 – Diagrama de Sequência: Cadastrar Emenda

A Figura 43 ilustra a mudança de status de uma emenda conforme o caso de uso UC05. Após permissão e validação do novo status, o EmendaService localiza a emenda e atualiza o campo de status. O histórico é gravado pelo HistoryService e o resultado retorna ao *Frontend*. Emendas inexistentes resultam em erros.

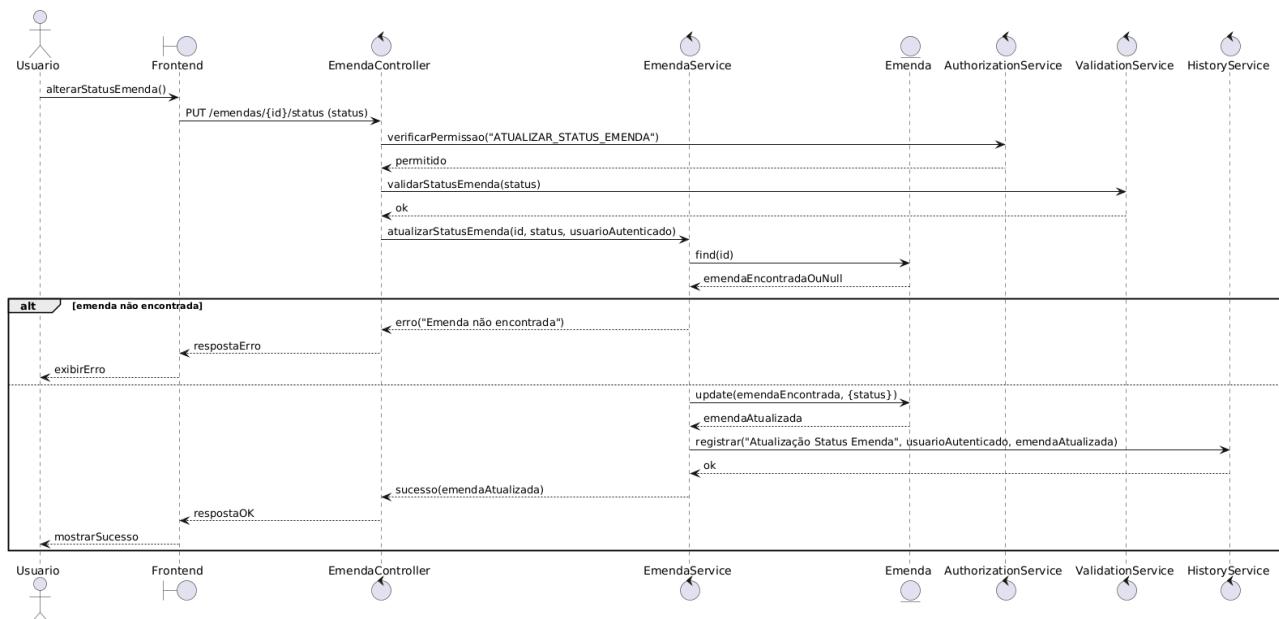


Figura 43 – Diagrama de Sequência: Atualizar Status Emenda

Na Figura 44, observa-se a operação de consulta de emendas, prevista no caso de uso UC05. O *controller* recebe filtros do *Frontend* e aciona o *EmendaService*, que monta a *query* no modelo *Emenda*. O resultado é exibido ao usuário. Como é uma operação de leitura, não há histórico nem atualização de estado.

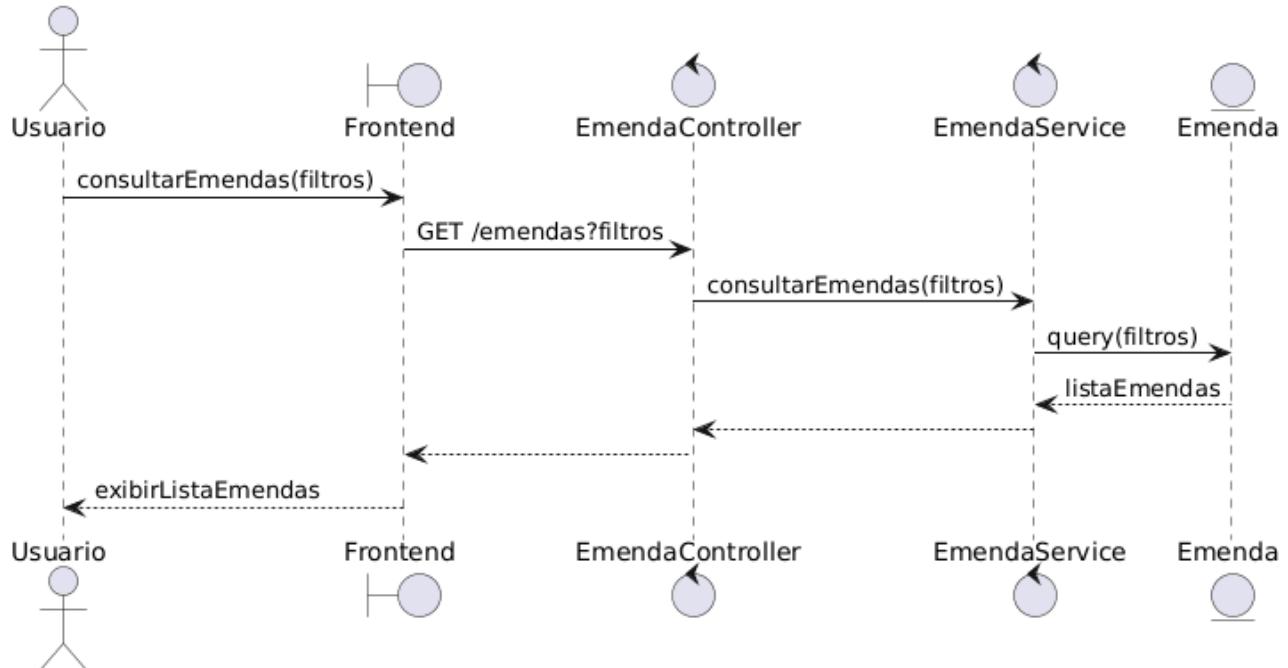


Figura 44 – Diagrama de Sequência: Consultar Emendas

Como mostra a Figura 45, o fluxo de abertura de demanda faz parte do caso de uso UC06. O *Frontend* envia os dados ao *controller*, que valida permissões e aciona o *ValidationService*. O *DemandasService* confirma a existência de cidade e instituição, cria o registro no modelo *Demandas* e aciona o *HistoryService* para registrar a criação. O retorno de sucesso é enviado ao usuário.

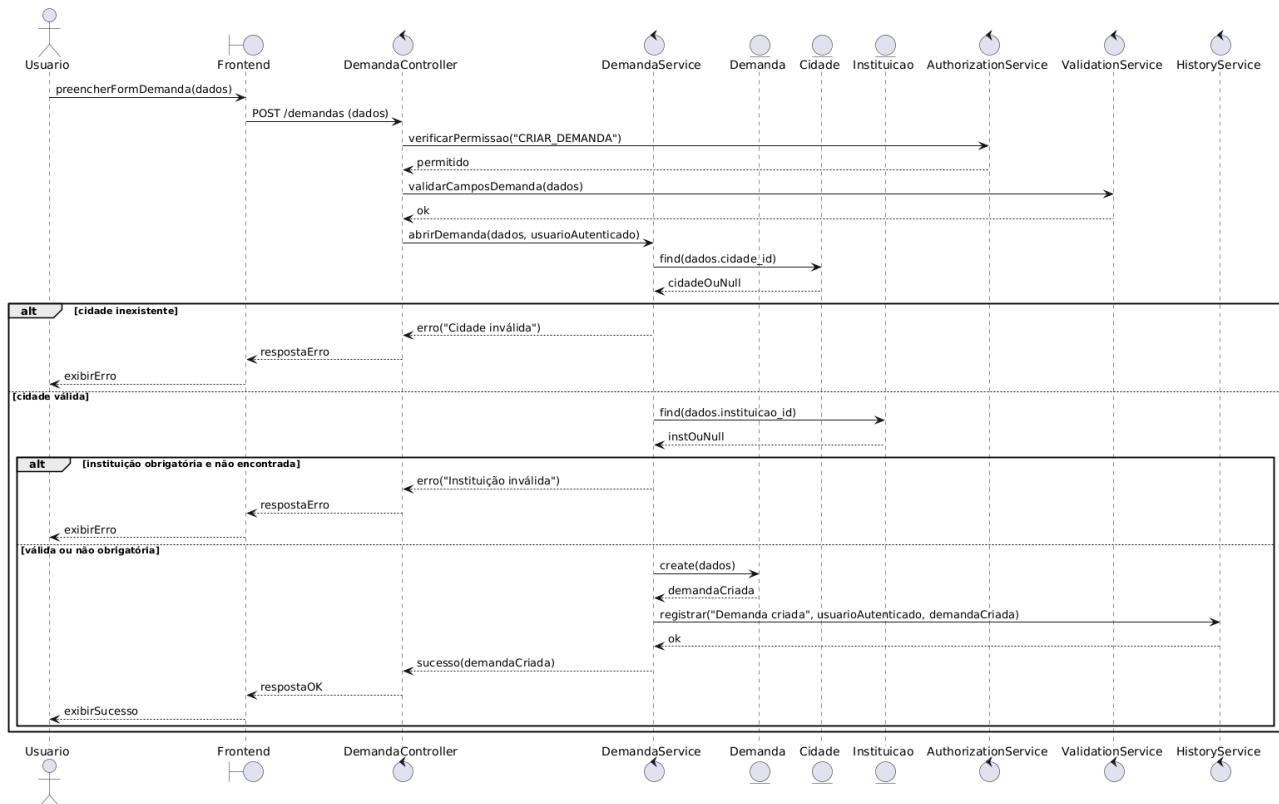


Figura 45 – Diagrama de Sequência: Abrir Demanda

Na Figura 46, o diagrama demonstra a operação de atribuição de responsável prevista no caso de uso UC06. O *controller* valida se o usuário tem permissão para atribuir responsáveis e o ValidationService verifica se os dados são válidos. O DemandaService confirma se o usuário destino existe antes de atualizar a demanda. O histórico é registrado e a resposta retornada ao *Frontend*.

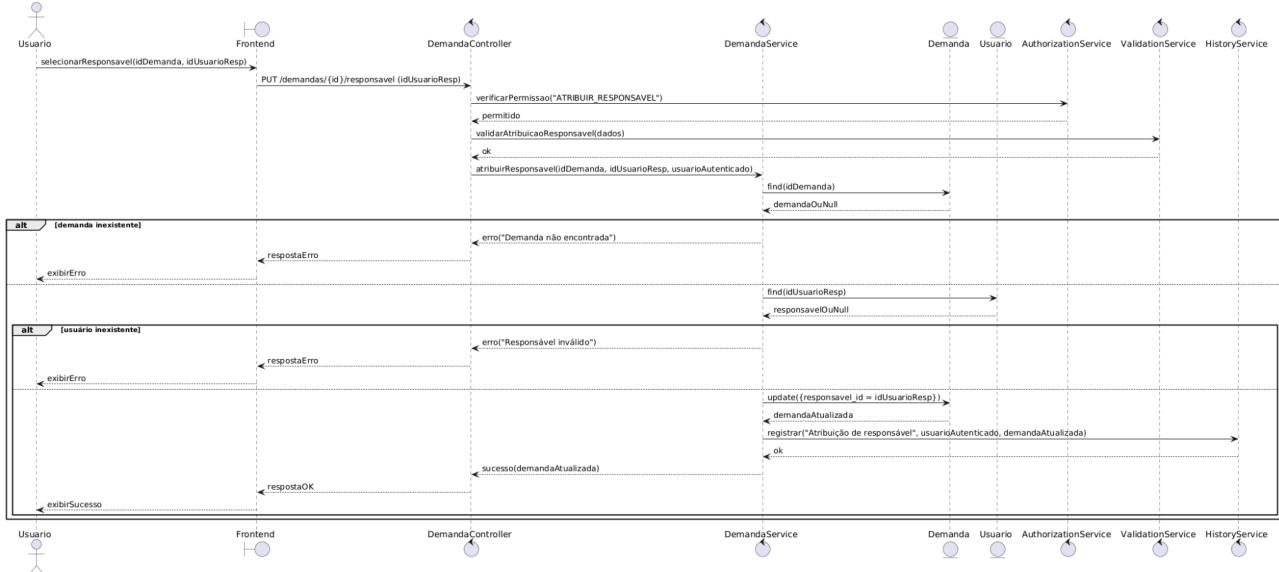


Figura 46 – Diagrama de Sequência: Atribuir Responsável

A Figura 47 representa o fluxo de registro de pessoa atendida na demanda, pertencente ao caso de uso UC06. Após validação de permissão e integridade dos dados, o DemandService atualiza a demanda com nome, CPF e contato da pessoa atendida. O evento é registrado no histórico e o usuário recebe a confirmação da operação.

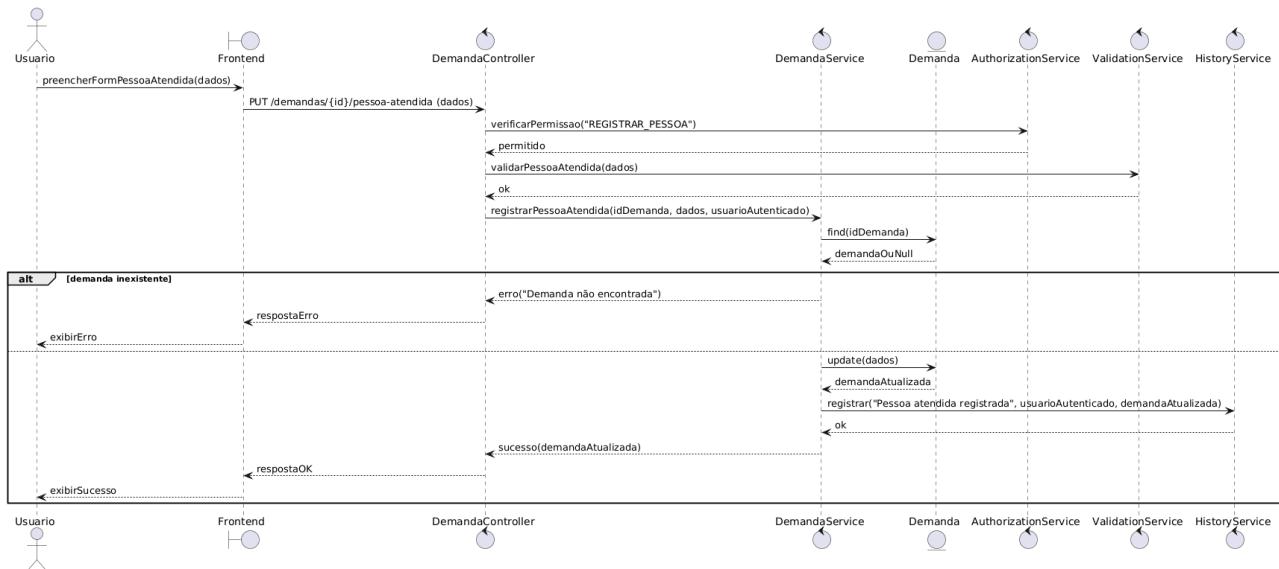


Figura 47 – Diagrama de Sequência: Registrar Pessoa Atendida

Na Figura 48, o diagrama mostra o fluxo de definição de prazo, integrante do caso de uso UC06. O Frontend envia a nova data ao *controller*, que valida permissões e usa o ValidationService para validar o formato e coerência da data. O DemandaService encontra a demanda e atualiza seu prazo, acionando o HistoryService para registrar a alteração. O resultado é retornado ao usuário.

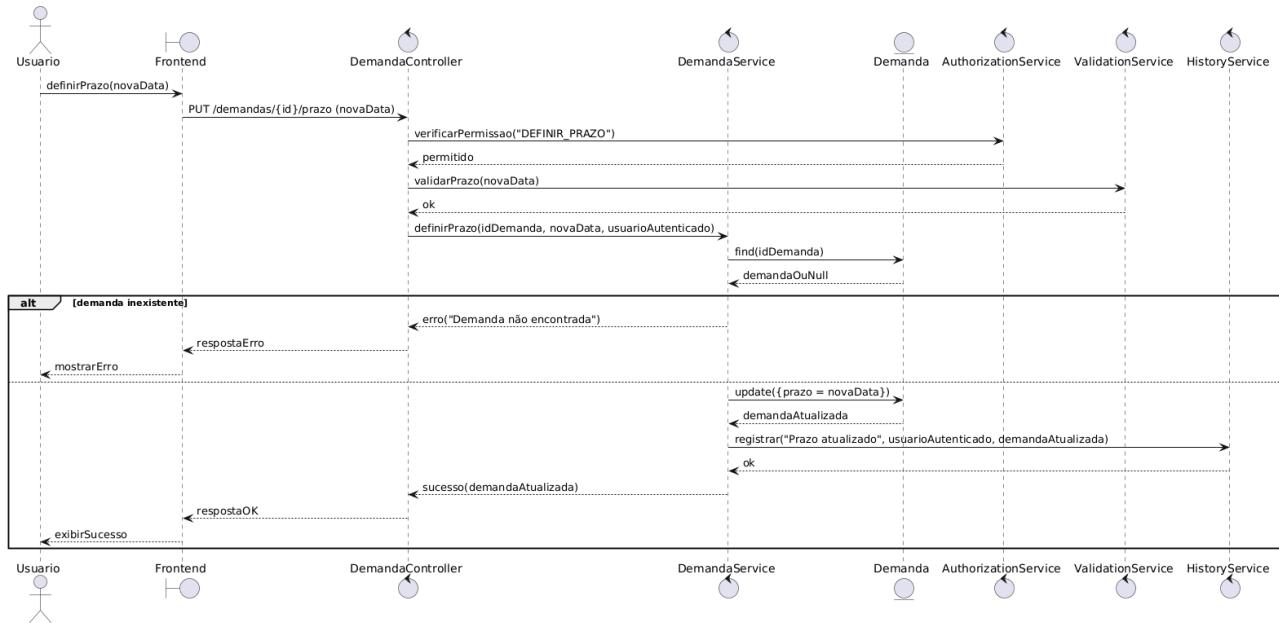


Figura 48 – Diagrama de Sequência: Definir Prazo

Como mostra a Figura 49, o diagrama de sequência descreve o fluxo de atualização de status pertencente ao caso de uso UC06. O *controller* valida permissões, o ValidationService verifica a transição de status e o DemandaService atualiza o registro no modelo Demanda. O histórico é registrado e o resultado enviado ao usuário. Caso a demanda não exista, um erro é retornado.

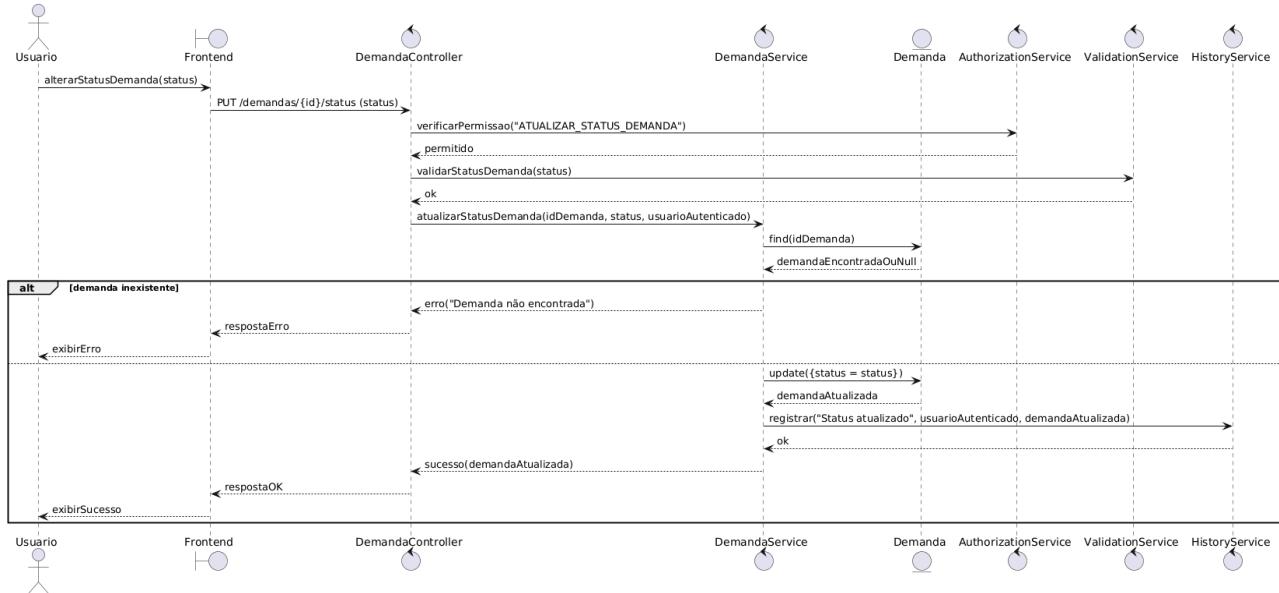


Figura 49 – Diagrama de Sequência: Alterar Status Demanda

Na Figura 50, observa-se o fluxo mais complexo do caso de uso UC06, que trata do anexo de ofícios às demandas. Após validação de permissões e integridade do arquivo, o FileStorageService armazena o documento e retorna o caminho ao DemandService, que cria o registro de anexo e registra o histórico da operação. O usuário recebe a confirmação final.

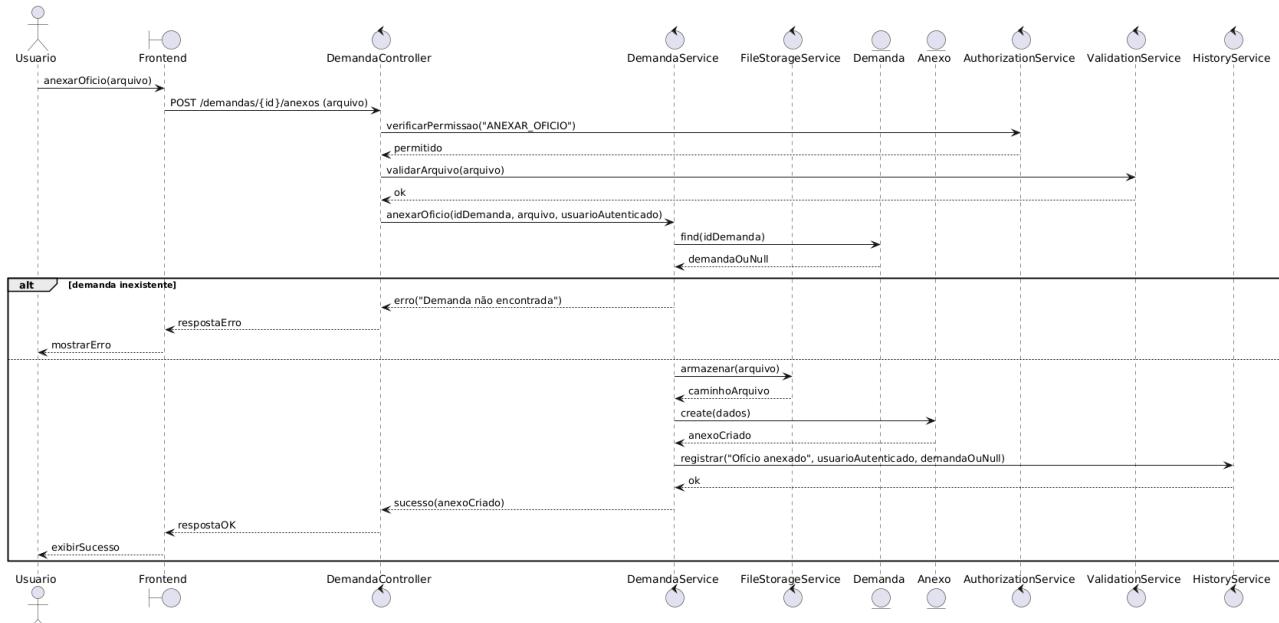


Figura 50 – Diagrama de Sequência: Anexar Ofício

A Figura 51 apresenta a operação de criação de evento prevista no caso de uso UC07. Após validação de permissão e dados, o AgendaService cria o registro no modelo Evento e registra o histórico. O usuário recebe uma confirmação de que o evento foi criado com sucesso.

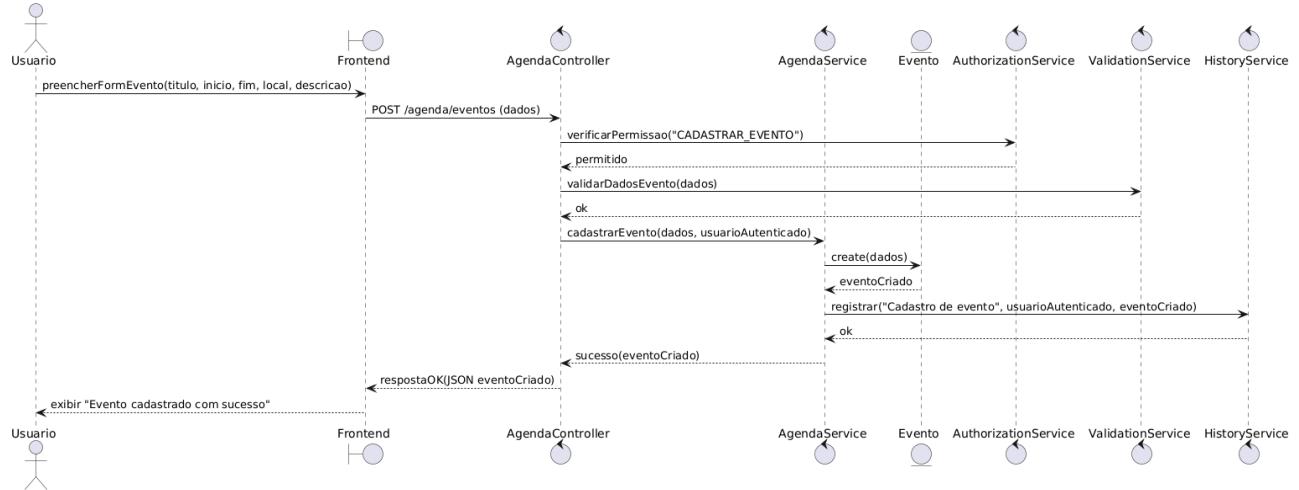


Figura 51 – Diagrama de Sequência: Cadastrar Evento

Como mostra a Figura 52, o diagrama descreve a criação de alertas vinculados a eventos, conforme o caso de uso UC07. Após validação de permissão e integridade, o AgendaService registra o alerta e aciona o NotificationService para agendar o envio da notificação. A confirmação retorna ao *Frontend*.

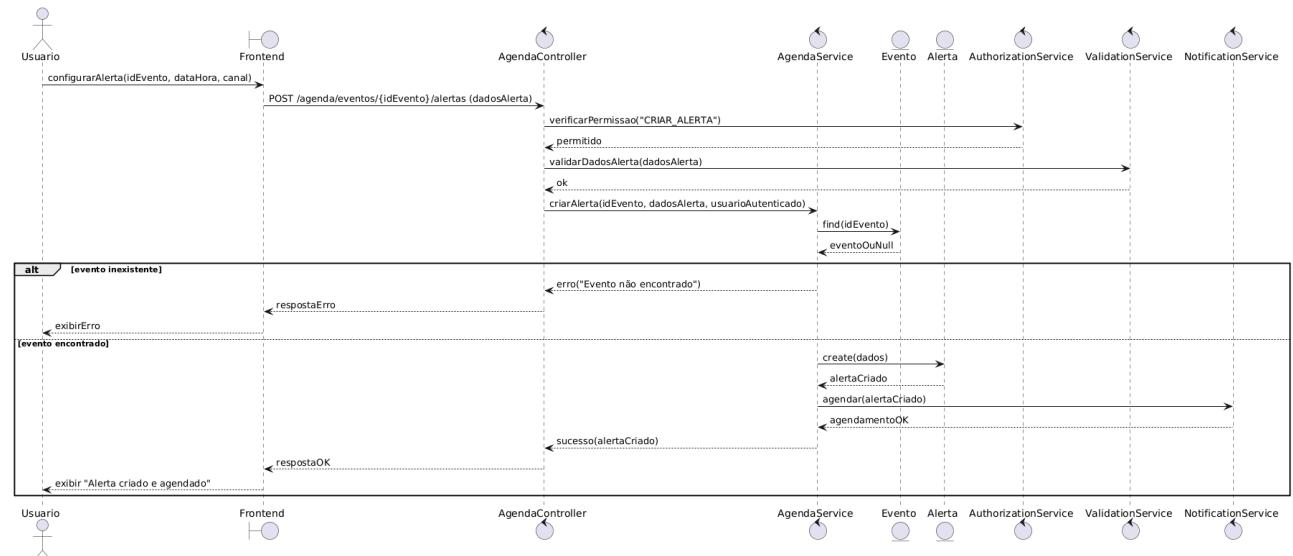


Figura 52 – Diagrama de Sequência: Criar Alerta

Na Figura 53, observa-se o fluxo de consulta do calendário, integrante do caso de uso UC07. O *controller* recebe a requisição do *Frontend* e aciona o *AgendaService* para recuperar os eventos do mês. Os dados retornam ao usuário que visualiza o calendário atualizado.

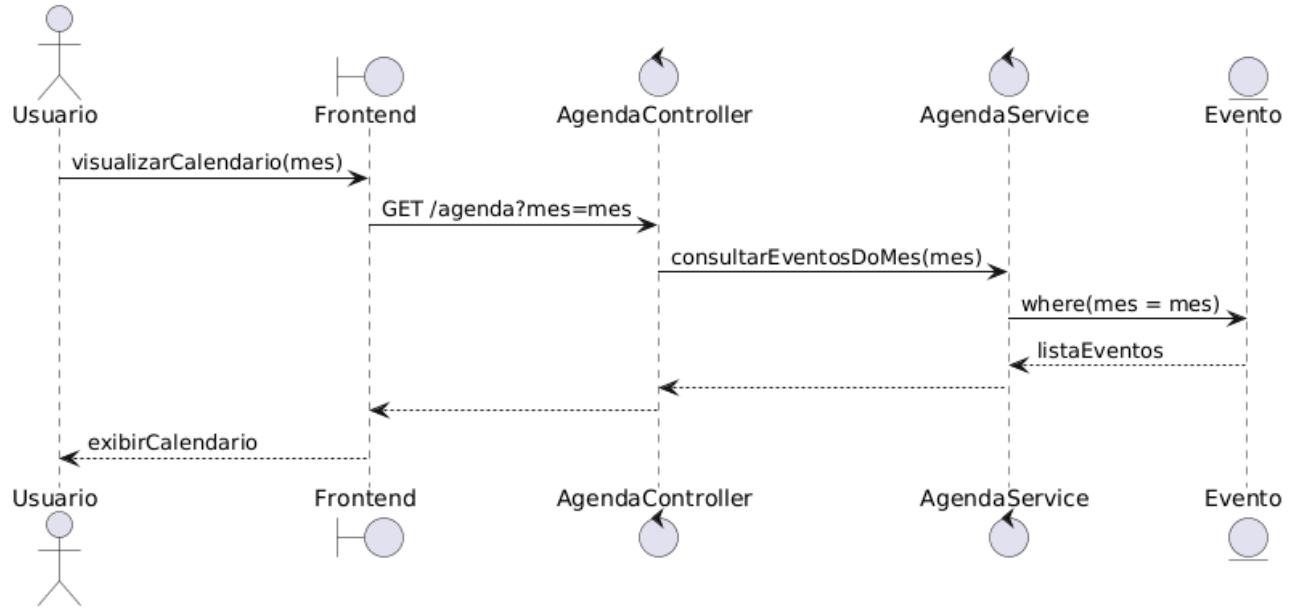


Figura 53 – Diagrama de Sequência: Visualizar Agenda

A Figura 54 apresenta a operação de geração do relatório de Projetos de Lei, prevista no caso de uso UC08. Após validação de permissão específica, o RelatorioService consulta todos os PLs e gera o documento consolidado. O relatório é devolvido ao *Frontend* para *download*.

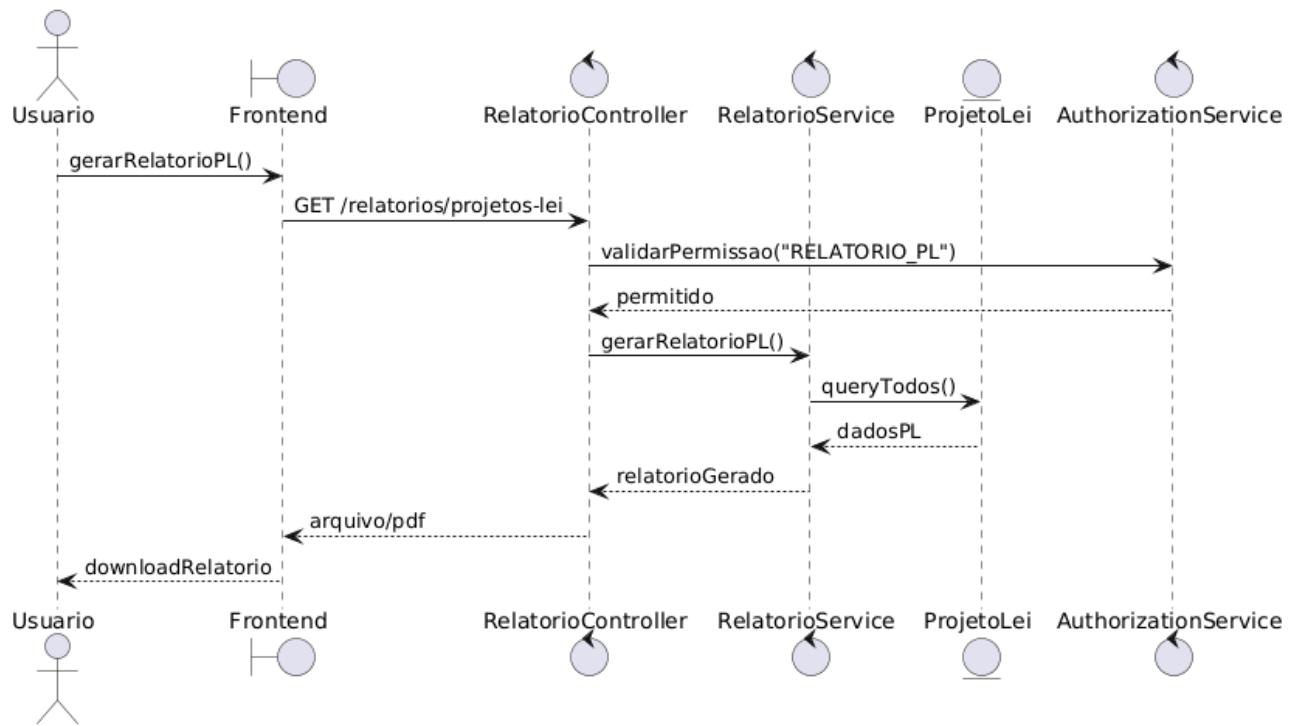


Figura 54 – Diagrama de Sequência: Relatório de Projetos de Lei

Como mostra a Figura 55, o *controller* valida a permissão do usuário, aciona o RelatorioService e o serviço consulta o conjunto de emendas antes de gerar o relatório. O *Frontend* recebe o arquivo gerado para *download*.

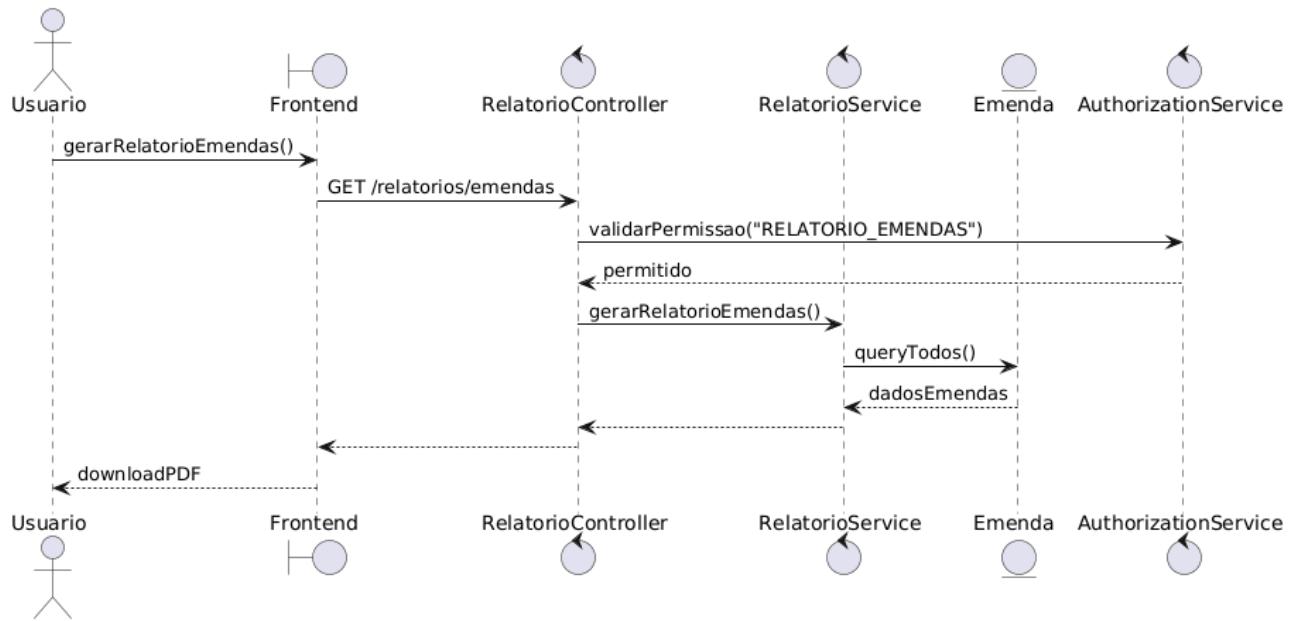


Figura 55 – Diagrama de Sequência: Relatório de Emendas

Na Figura 56, o diagrama evidencia a geração do relatório de demandas conforme o caso de uso UC08, com validação de permissão, consulta de dados e geração do documento consolidado.

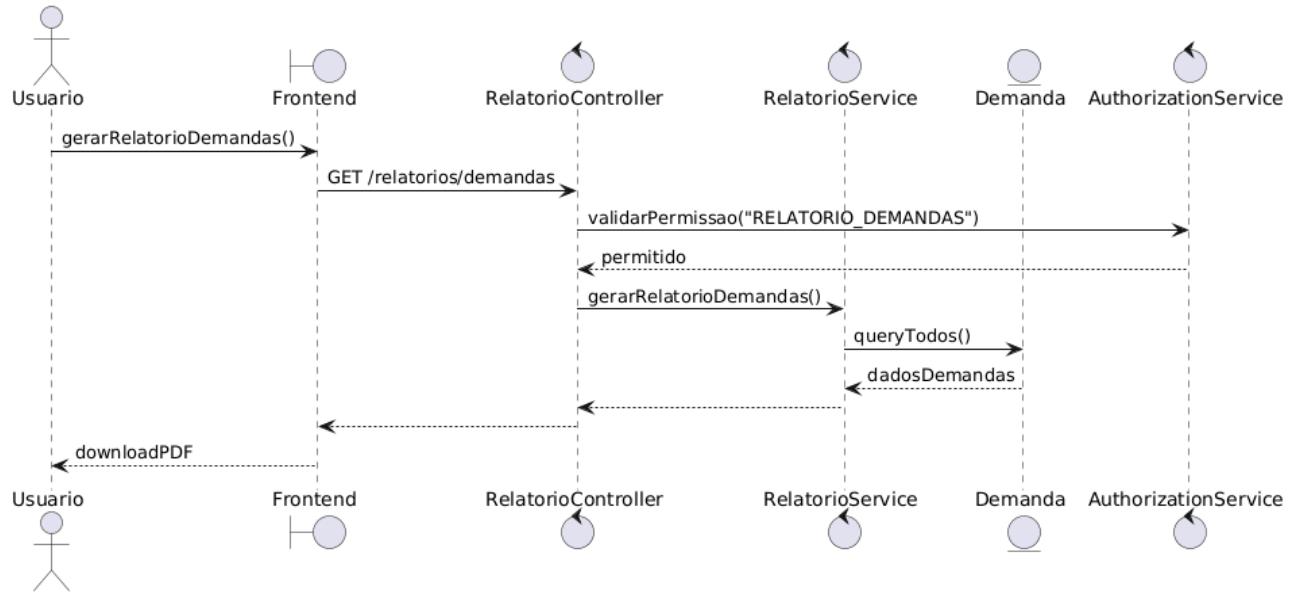


Figura 56 – Diagrama de Sequência: Relatório de Demandas

A Figura 57 demonstra a operação mais ampla do caso de uso UC08, na qual o RelatorioService consolida dados de emendas, projetos de lei e demandas em um único documento acessível apenas à Deputada. O *controller* valida a permissão, o serviço consulta todas as entidades e gera o relatório final para *download*.

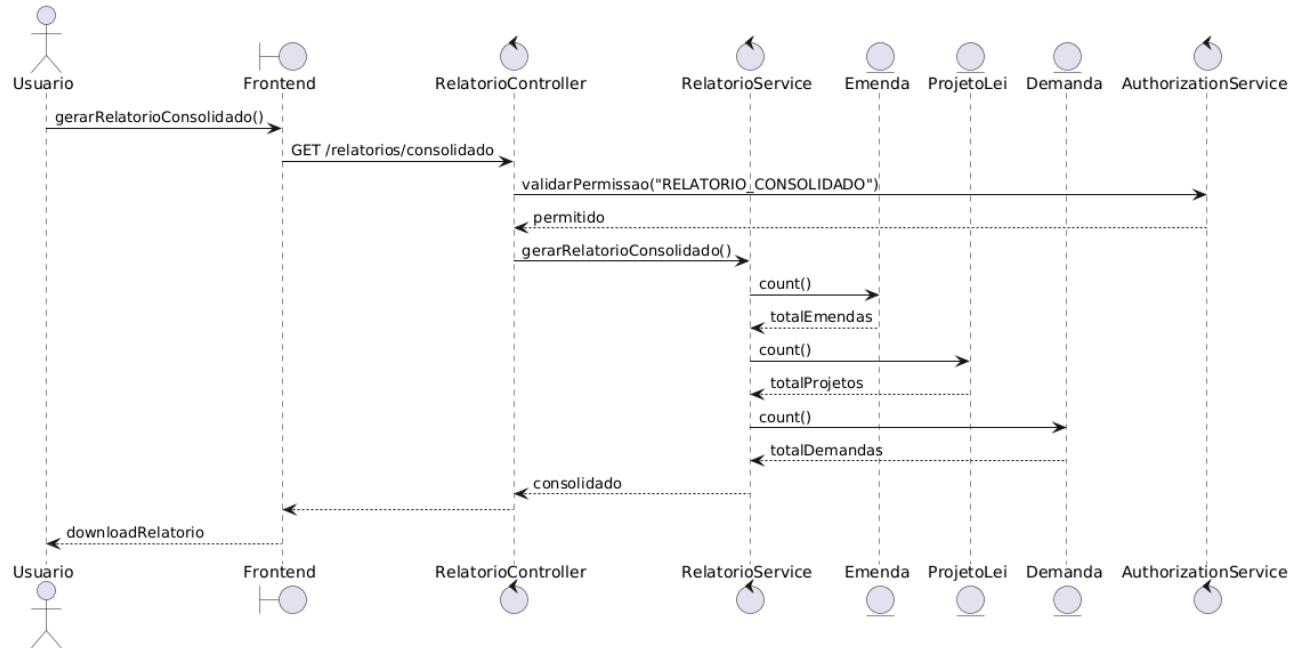


Figura 57 – Diagrama de Sequência: Relatório Consolidado

3.3 Diagramas de Comunicação

Representam o fluxo de mensagens entre os objetos participantes dos casos de uso, destacando as relações estruturais e a cooperação entre componentes durante as operações do sistema.

Como mostra a Figura 58, o diagrama de comunicação da operação de autenticação, relacionada ao caso de uso UC01 – Autenticar no Sistema + Validar Permissão, evidencia a cooperação entre o usuário, a interface *Frontend*, o *AuthController*, o *AuthService*, o repositório de usuários e o serviço de autorização. O usuário envia suas credenciais pela interface *web*, que encaminha a requisição ao controlador. Este delega ao serviço de autenticação a validação do e-mail e da senha no repositório de usuários e, em seguida, aciona o serviço de autorização para conferir o perfil e as permissões básicas. Apenas após a validação conjunta desses componentes é que o *token* de acesso é gerado e retornado ao *Frontend*, ou, em caso de falha, uma mensagem de erro é devolvida.

Comunicação - UC01 Autenticar no Sistema

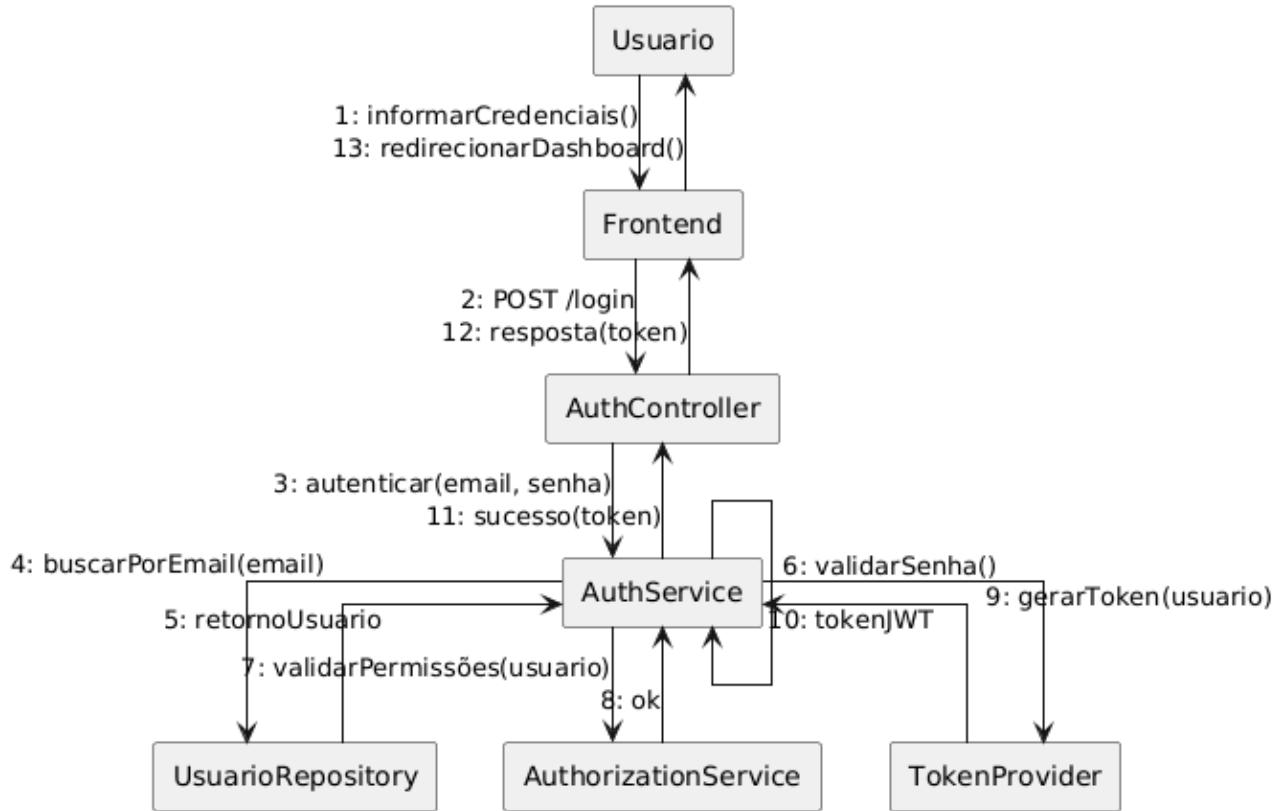


Figura 58 – Diagrama de Comunicação: Autenticar no Sistema

Na Figura 59, o diagrama de comunicação da operação de cadastro de cidade, associada ao caso de uso UC02 – Gerenciar Cadastros, mostra o usuário interagindo com o **Frontend**, que envia os dados para o **CidadeController**. O controlador consulta o serviço de autorização para verificar se o perfil possui permissão de cadastro e, em seguida, aciona o **CidadeService** para validar os campos obrigatórios. Estando tudo correto, o serviço persiste a nova cidade no repositório/modelo **Cidade** e retorna a confirmação ao controlador e ao **Frontend**.

Comunicação - UC02 Cadastrar Cidade

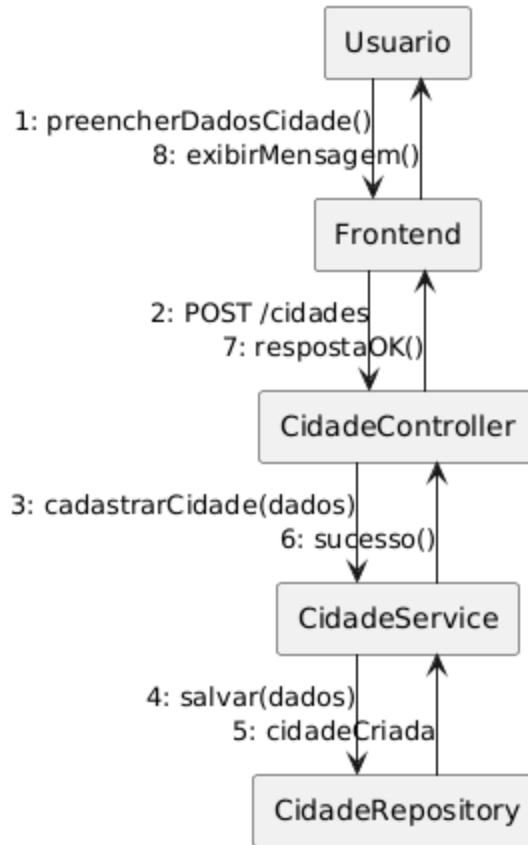


Figura 59 – Diagrama de Comunicação: Cadastrar Cidade

A Figura 60 apresenta o diagrama de comunicação da operação de consulta de cidades, também pertencente ao caso de uso UC02 – Gerenciar Cadastros. Nesse cenário, o usuário aciona a *interface* para listar ou filtrar cidades, o Frontend encaminha a requisição ao CidadeController, que valida a permissão e delega ao CidadeService a montagem da consulta. O serviço acessa o repositório Cidade com filtros de pesquisa e devolve a lista ao controlador, que repassa os dados ao *Frontend*. Como se trata de uma operação de leitura, não há registro de histórico nem alteração de estado.

Comunicação - UC02 Consultar Cidades

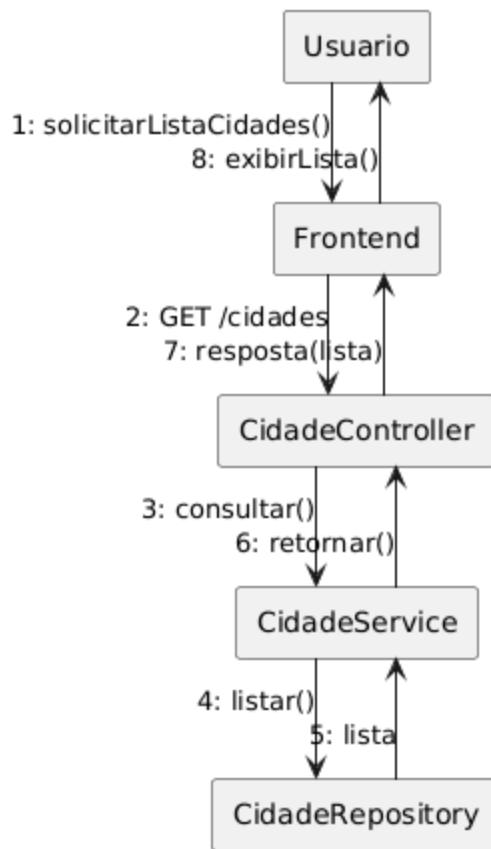


Figura 60 – Diagrama de Comunicação: Consultar Cidades

Como mostra a Figura 61, o diagrama de comunicação da operação de cadastro de instituição, relacionada ao caso de uso UC02 – Gerenciar Cadastros, evidencia a sequência de mensagens entre usuário, *Frontend*, *InstituicaoController*, *InstituicaoService* e repositório *Instituicao*. O controlador primeiro verifica se o usuário tem permissão para cadastrar instituições, depois o serviço valida os dados (nome, tipo, cidade associada) e consulta o repositório de cidades, quando necessário. Em seguida, a instituição é criada no banco, e a resposta de sucesso retorna ao usuário.

Comunicação - UC02 Cadastrar Instituição

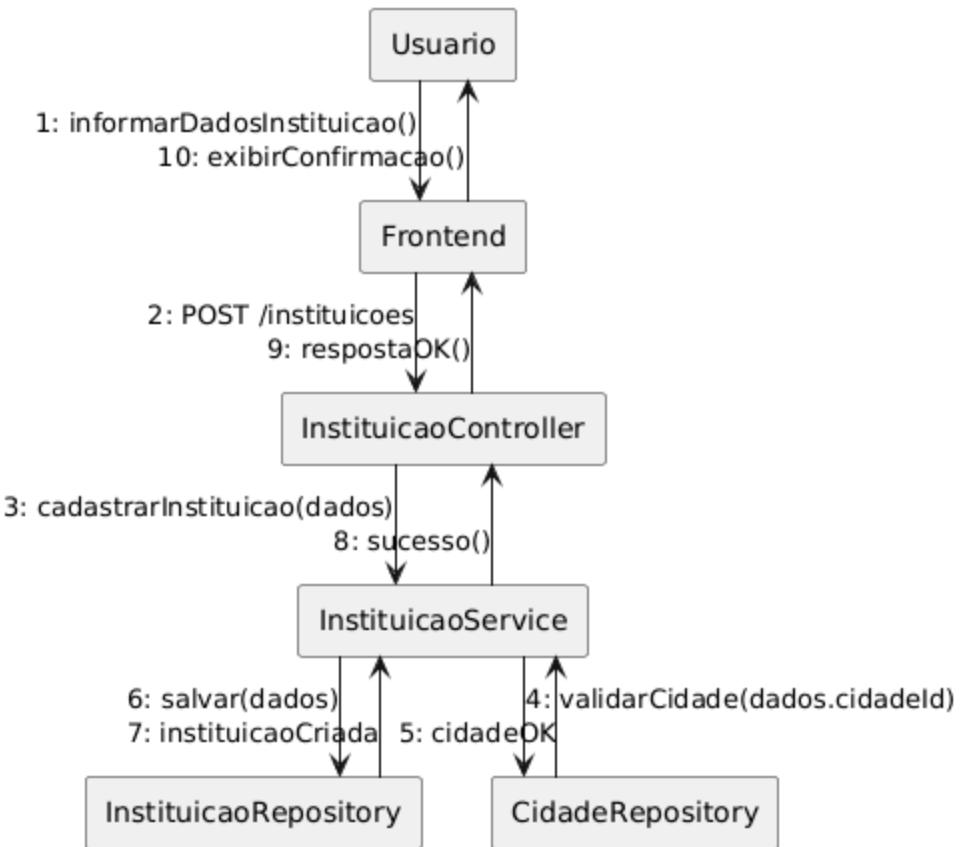


Figura 61 – Diagrama de Comunicação: Cadastrar Instituição

Na Figura 62, o diagrama de comunicação da operação de consulta de instituições, ainda dentro do caso de uso UC02 – Gerenciar Cadastros, mostra um fluxo mais enxuto: o *Frontend* envia os filtros ao *InstituicaoController*, que valida a permissão e repassa ao *InstituicaoService*. O serviço monta a consulta sobre o repositório *Instituicao*, obtém os resultados e os devolve ao controlador, que responde ao *Frontend*. Nenhum dado é alterado, apenas consultado.

Comunicação - UC02 Consultar Instituições

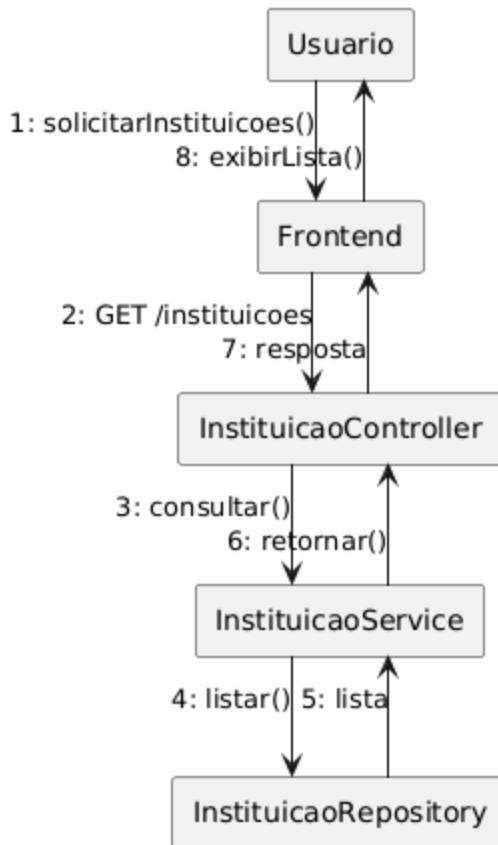


Figura 62 – Diagrama de Comunicação: Consultar Instituições

A Figura 63 apresenta o diagrama de comunicação da operação de cadastro de igreja, pertencente ao caso de uso UC02 – Gerenciar Cadastros. O usuário preenche os dados da igreja no *Frontend*; o *IgrejaController* manda os dados para *IgrejaService* consulta o repositório de cidades para garantir que o município informado existe e, em seguida, grava a nova igreja no repositório *Igreja*. O resultado da operação é devolvido ao usuário.

Comunicação - UC02 Cadastrar Igreja

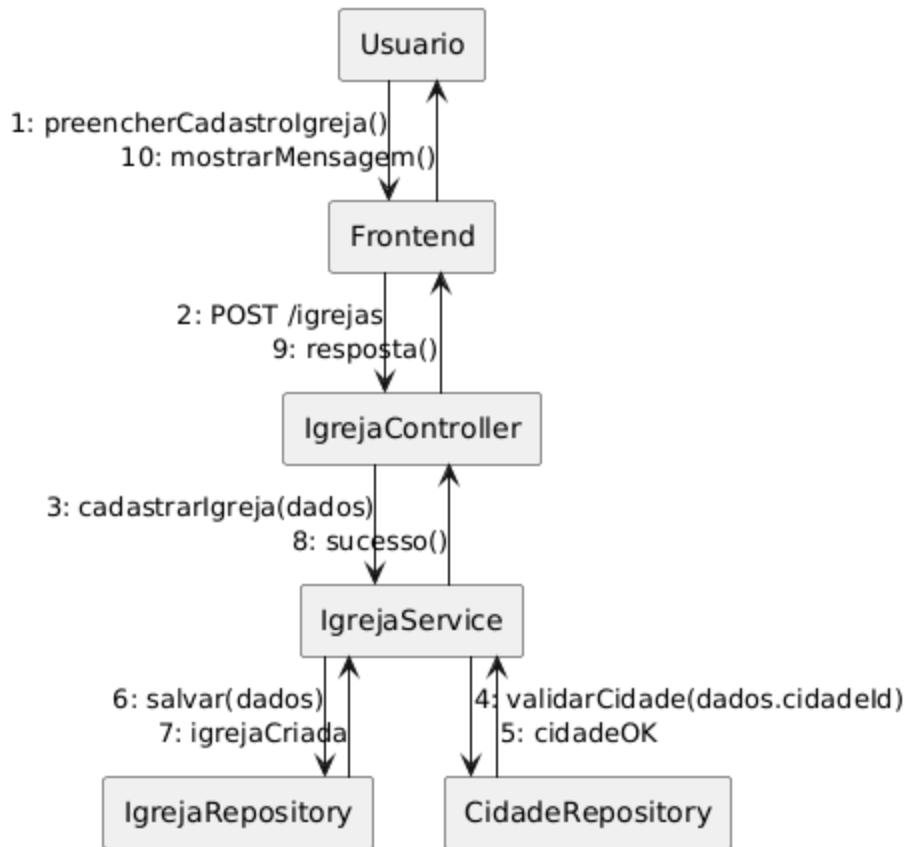


Figura 63 – Diagrama de Comunicação: Cadastrar Igreja

Como mostra a Figura 64, o diagrama de comunicação de consulta de igrejas, ligado ao caso de uso UC02 – Gerenciar Cadastros, ilustra o *Frontend* solicitando a listagem ao *IgrejaController*, que verifica permissão e aciona o *IgrejaService*. Este aplica filtros de pesquisa e consulta o repositório *Igreja*, devolvendo a lista ao controlador e, por fim, à interface. O foco aqui é apenas leitura dos registros existentes.

Comunicação - UC02 Consultar Igrejas

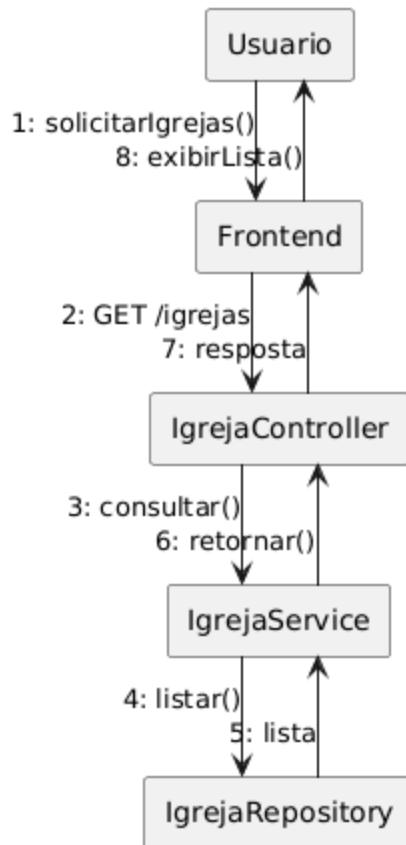


Figura 64 – Diagrama de Comunicação: Consultar Igrejas

Na Figura 65, o diagrama de comunicação de cadastro de liderança, ainda no contexto do caso de uso UC02 – Gerenciar Cadastros, mostra a interação entre o usuário, o *Frontend*, o LiderancaController, o LiderancaService e os repositórios Instituicao e Lideranca. Além de validar permissão e dados básicos, o serviço precisa confirmar a existência da instituição vinculada antes de criar a liderança no banco. O resultado é uma nova liderança persistida e retornada ao usuário.

Comunicação - UC02 Cadastrar Liderança

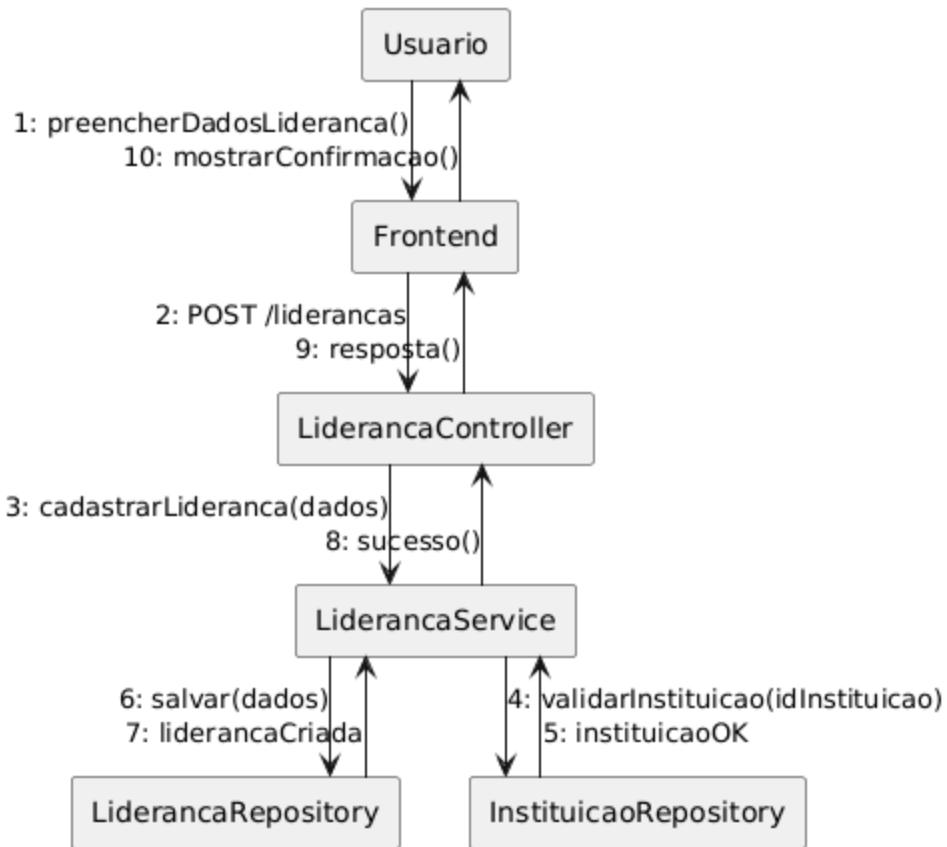


Figura 65 – Diagrama de Comunicação: Cadastrar Liderança

A Figura 66 apresenta o diagrama de comunicação para consulta de lideranças, também vinculado ao caso de uso UC02 – Gerenciar Cadastros. O *Frontend* envia filtros (como instituição, município ou nome), o *LiderancaController* valida a permissão e o *LiderancaService* consulta o repositório *Lideranca* com esses critérios. A lista de lideranças filtradas é devolvida ao controlador e apresentada ao usuário.

Comunicação - UC02 Consultar Lideranças

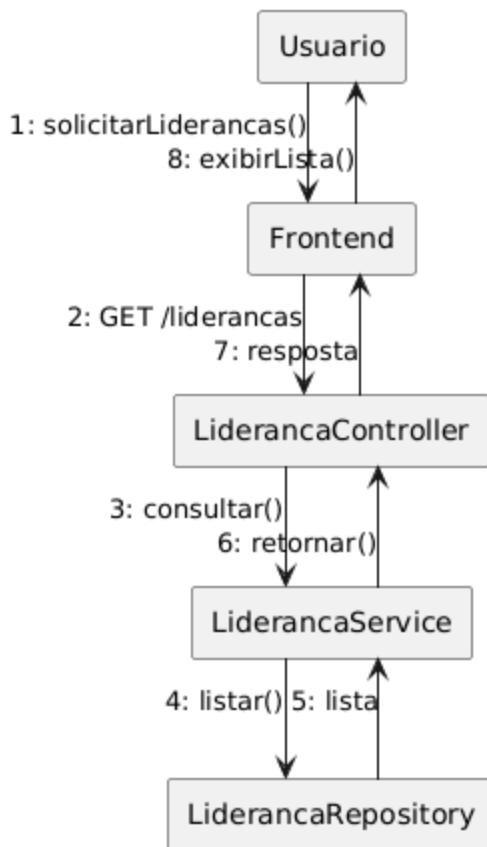


Figura 66 – Diagrama de Comunicação: Consultar Lideranças

Como mostra a Figura 67, o diagrama de comunicação para cadastro de notícia, associado ao caso de uso UC03 – Gerenciar Conteúdo do Site, destaca a cooperação entre o usuário, o *Frontend*, o *NoticiaController*, o *NoticiaService* e o repositório *Noticia*. O controlador verifica se o perfil tem permissão editorial, o serviço valida título e conteúdo e, em seguida, grava a notícia vinculando o autor (usuário autenticado). A resposta retorna ao *Frontend* com a confirmação da publicação.

Comunicação - UC03 Cadastrar Notícia

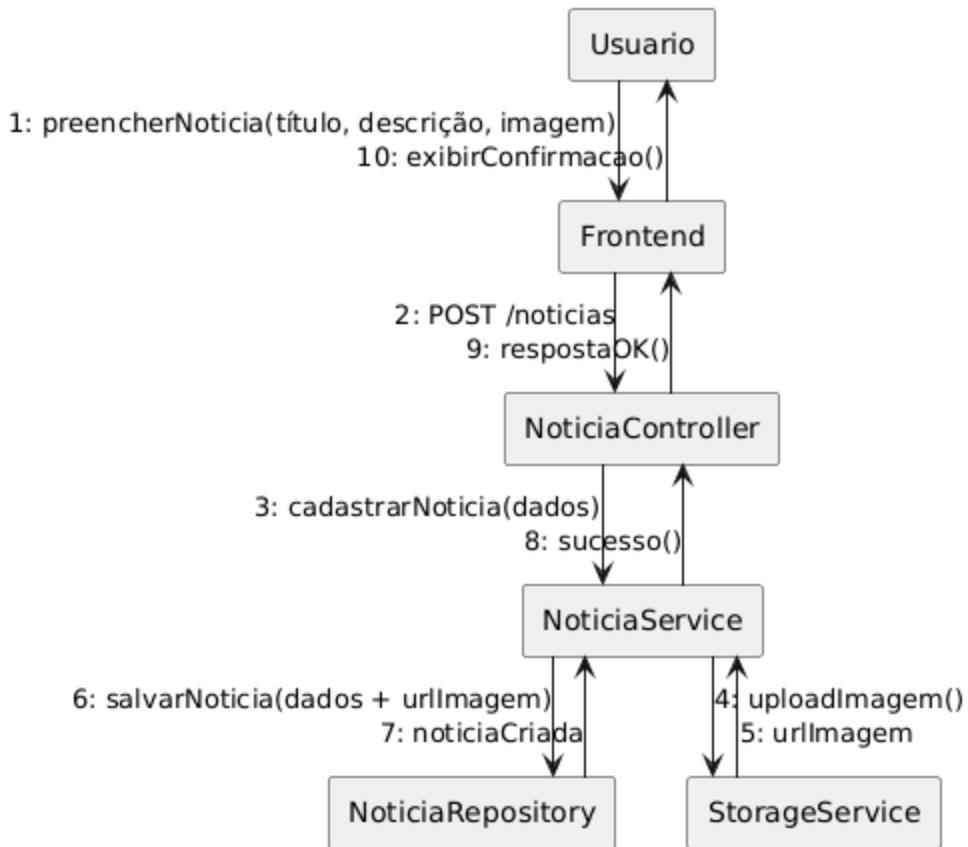


Figura 67 – Diagrama de Comunicação: Cadastrar Notícia

Na Figura 68, o diagrama de comunicação de consulta de notícias, ainda dentro do caso de uso UC03 – Gerenciar Conteúdo do *Site*, representa a interface solicitando notícias ao **NoticiaController**. Após a verificação de permissão (que pode ser mais permissiva para leitura), o **NoticiaService** constrói a consulta no repositório **Noticia** com base em filtros (data, título, autor), retornando o conjunto de notícias exibidas ao usuário.

Comunicação - UC03 Consultar Notícias

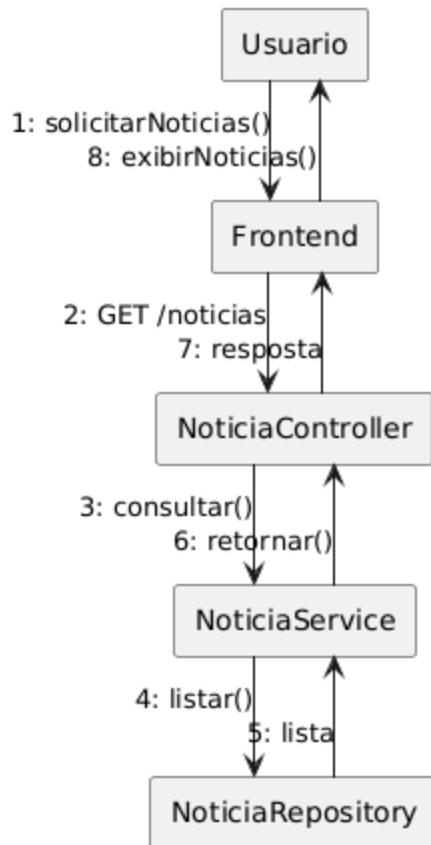


Figura 68 – Diagrama de Comunicação: Consultar Notícias

A Figura 69 mostra o diagrama de comunicação da operação de atualização da seção de Missão, ligada ao caso de uso UC03 – Gerenciar Conteúdo do Site. O usuário envia o novo texto pelo *Frontend*, o *ConteudoController* verifica permissões específicas e realiza a checagem de campos obrigatórios. O *ConteudoService* atualiza o registro da missão no repositório correspondente e registra a alteração no histórico, quando aplicável. O controlador então devolve ao usuário a confirmação de que o conteúdo foi atualizado.

Comunicação - UC03 Atualizar Missão

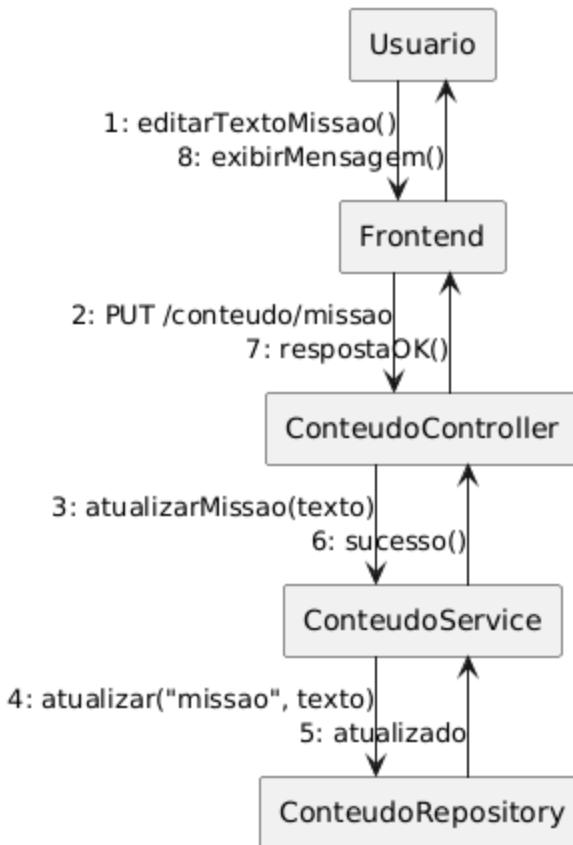


Figura 69 – Diagrama de Comunicação: Atualizar Missão

Como mostra a Figura 70, o diagrama de comunicação de atualização da Trajetória, também relacionado ao caso de uso UC03 – Gerenciar Conteúdo do Site, repete a mesma estrutura de colaboração: o *Frontend* chama o *ConteudoController*, que valida permissão e dados; o *ConteudoService* persiste a nova trajetória e aciona o serviço de histórico, garantindo rastreabilidade. A resposta de sucesso é retornada ao usuário.

Comunicação - UC03 Atualizar Trajetória

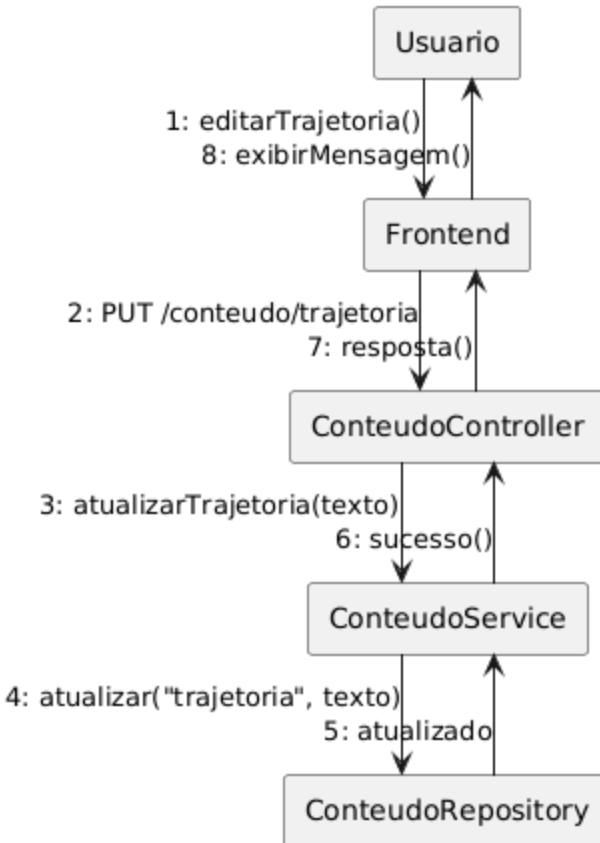


Figura 70 – Diagrama de Comunicação: Atualizar Trajetória

Na Figura 71, o diagrama de comunicação de cadastro de projeto do *site*, ainda no escopo do caso de uso UC03 – Gerenciar Conteúdo do *Site*, mostra o *Frontend* enviando os dados ao ProjetoSiteController. Após validação de permissão e campos, o ProjetoSiteService grava o projeto no repositório ProjetoSite e registra a operação no histórico. O usuário recebe a confirmação com as informações do novo projeto institucional.

Comunicação - UC03 Cadastrar Projeto do Site

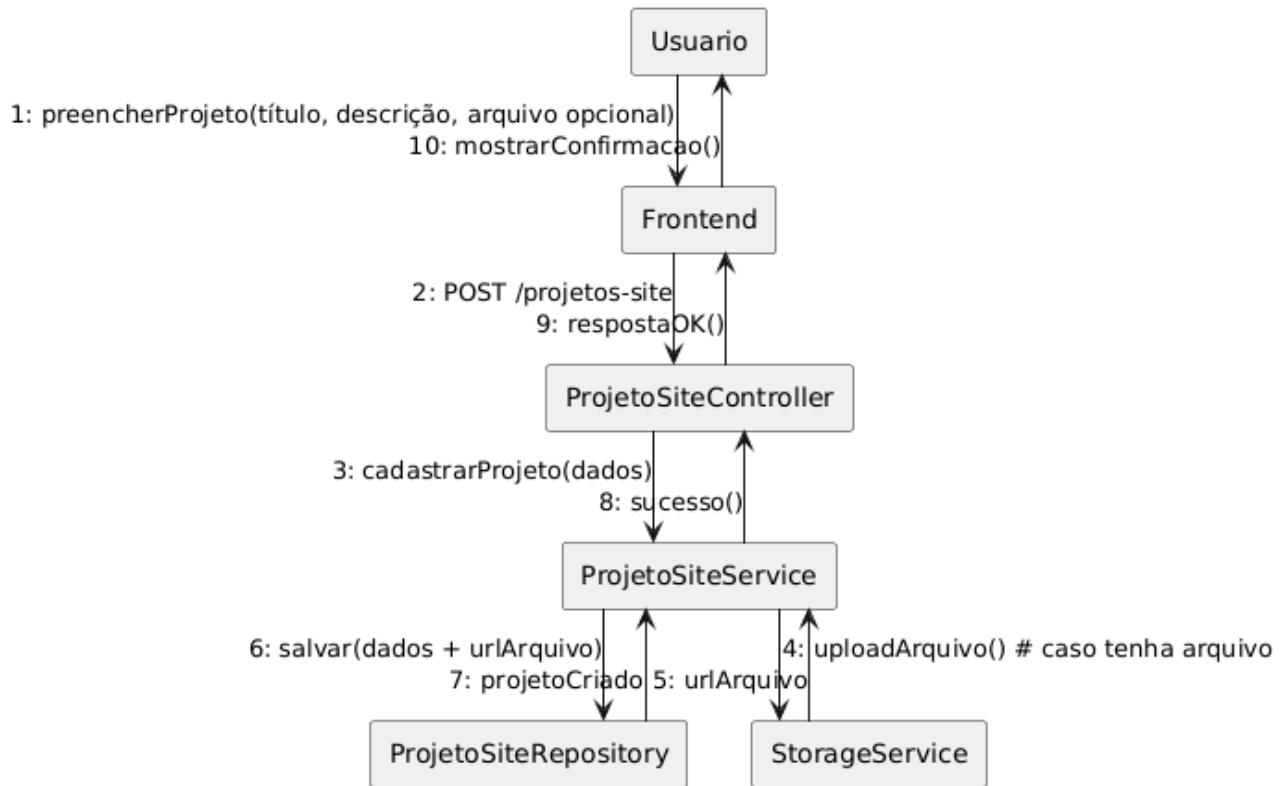


Figura 71 – Diagrama de Comunicação: Cadastrar Projeto Site

A Figura 72 apresenta o diagrama de comunicação da consulta de projetos do *site*, vinculada ao caso de uso UC03 – Gerenciar Conteúdo do *Site*. O usuário aciona filtros na interface, o ProjetoSiteController valida permissões e encaminha ao ProjetoSiteService, que consulta o repositório ProjetoSite. A lista de projetos é retornada ao controlador e exibida ao usuário sem alteração de estado.

Comunicação - UC03 Consultar Projetos do Site

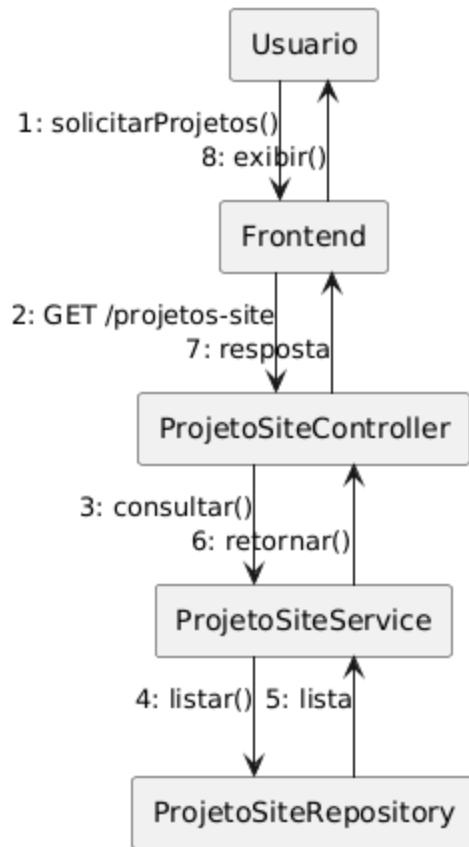


Figura 72 – Diagrama de Comunicação: Consultar Projetos Site

Como mostra a Figura 73, o diagrama de comunicação da operação de cadastro de Projeto de Lei, associada ao caso de uso UC04 – Gerenciar Projetos de Lei, evidencia a interação entre *Frontend*, *ProjetoLeiController*, *ProjetoLeiService* e repositório *ProjetoLei*. O serviço verifica se o número do PL já existe, valida os dados e, se tudo estiver correto, cria o registro e registra o histórico da criação.

Comunicação - UC04 Cadastrar Projeto de Lei

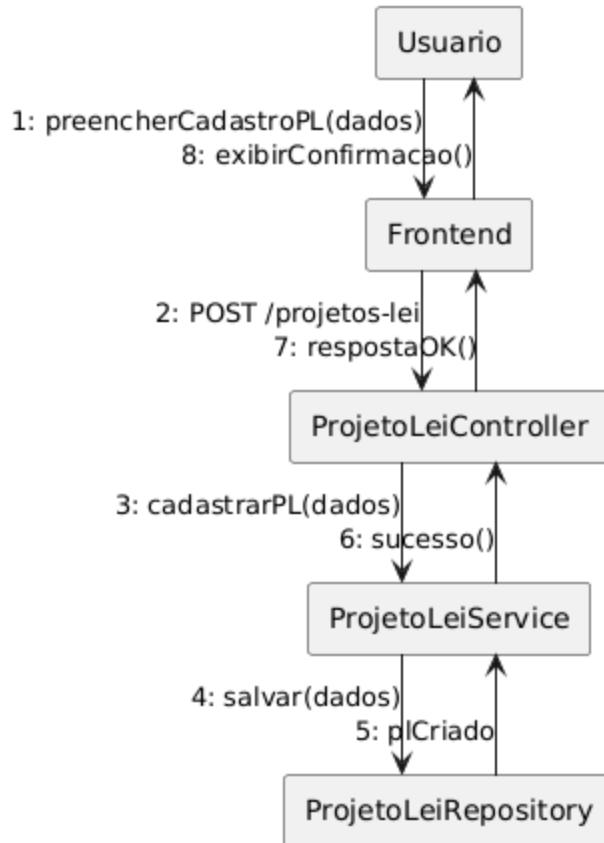


Figura 73 – Diagrama de Comunicação: Cadastrar Projeto de Lei

Na Figura 74, o diagrama de comunicação de consulta de Projetos de Lei, ainda no contexto do caso de uso UC04 – Gerenciar Projetos de Lei, mostra o *Frontend* enviando a requisição de pesquisa, o controlador validando a permissão e o *ProjetoLeiService* consultando o repositório *ProjetoLei* com filtros (*status*, número, tema etc.). O conjunto de PLs encontrado retorna ao *Frontend* para visualização.

Comunicação - UC04 Consultar Projetos de Lei

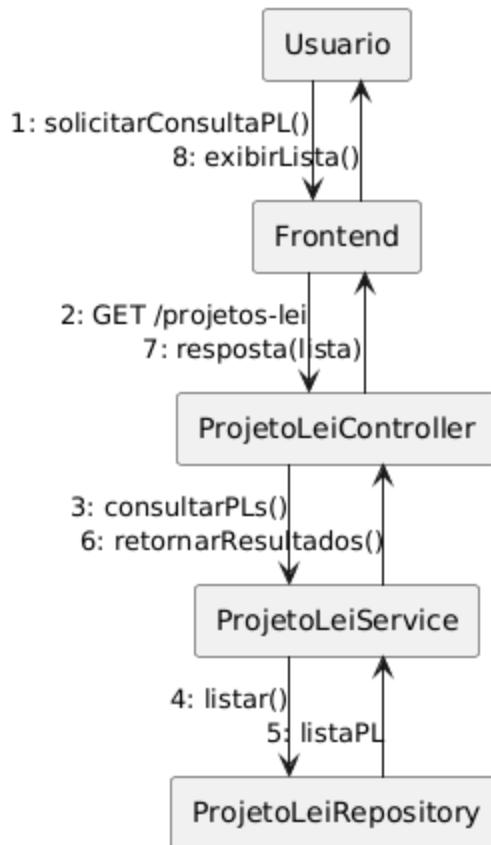


Figura 74 – Diagrama de Comunicação: Consultar Projeto de Lei

A Figura 75 apresenta o diagrama de comunicação da operação de atualização do *status* de um Projeto de Lei, também vinculada ao caso de uso UC04 – Gerenciar Projetos de Lei. O *Frontend* envia o identificador do PL e o novo *status* ao *ProjetoLeiController*, que valida a permissão e aciona o *ProjetoLeiService* que carrega o PL no repositório, atualiza o *status* e registra a mudança no histórico. A confirmação é devolvida ao usuário.

Comunicação - UC04 Atualizar Status de Projeto de Lei

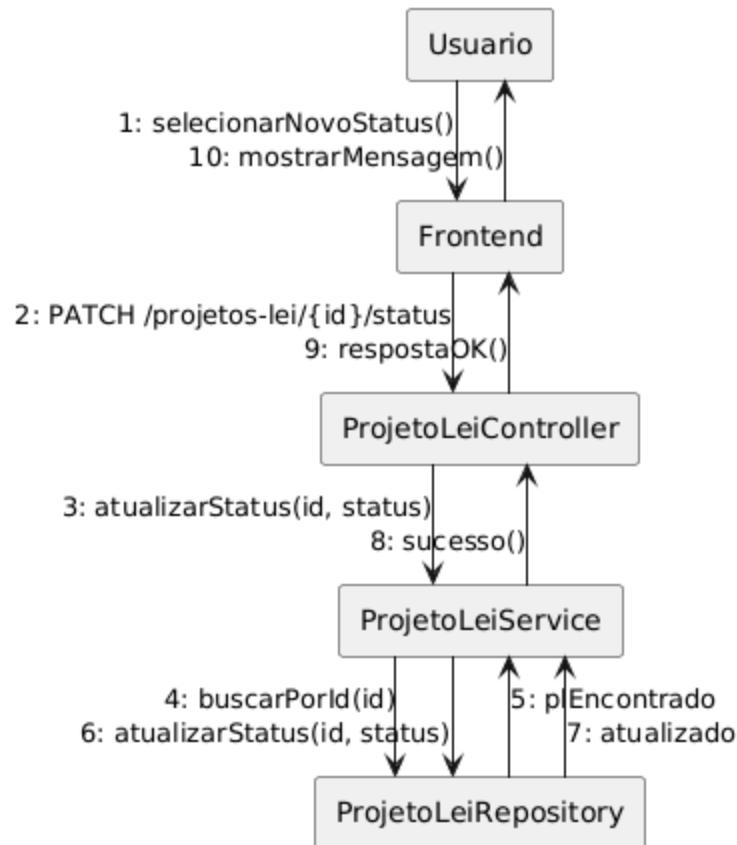


Figura 75 – Diagrama de Comunicação: Atualizar Status Projeto de Lei

Como mostra a Figura 76, o diagrama de comunicação de cadastro de emenda, relativo ao caso de uso UC05 – Gerenciar Emendas, exibe o fluxo entre *Frontend*, *EmendaController*, *EmendaService*, repositório *Emenda* e repositório *Cidade*. O serviço valida dados financeiros e referenciais, garante que a cidade informada existe e então grava a emenda, registrando o histórico da criação.

Comunicação - UC05 Cadastrar Emenda

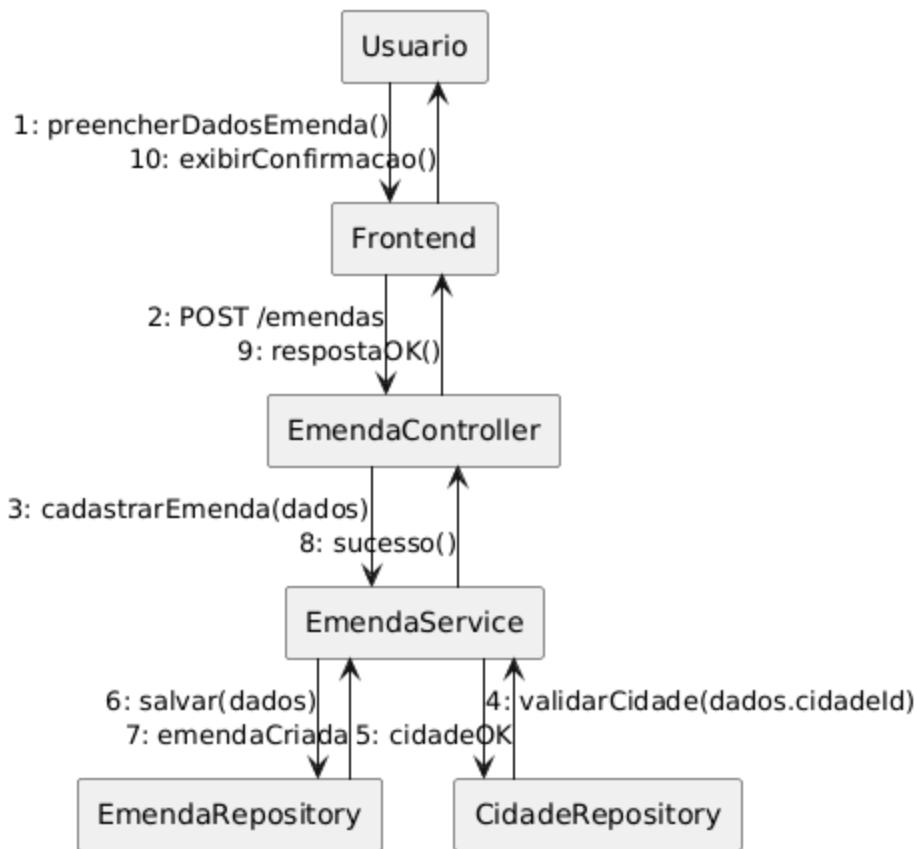


Figura 76 – Diagrama de Comunicação: Cadastrar Emenda

Na Figura 77, o diagrama de comunicação de consulta de emendas, também associado ao caso de uso UC05 – Gerenciar Emendas, mostra o *Frontend* solicitando a listagem ao *EmendaController*. Após validar permissões, o *EmendaService* consulta o repositório *Emenda* aplicando filtros (*cidade*, área de aplicação, *status*) e retorna a lista de emendas ao usuário.

Comunicação - UC05 Consultar Emendas

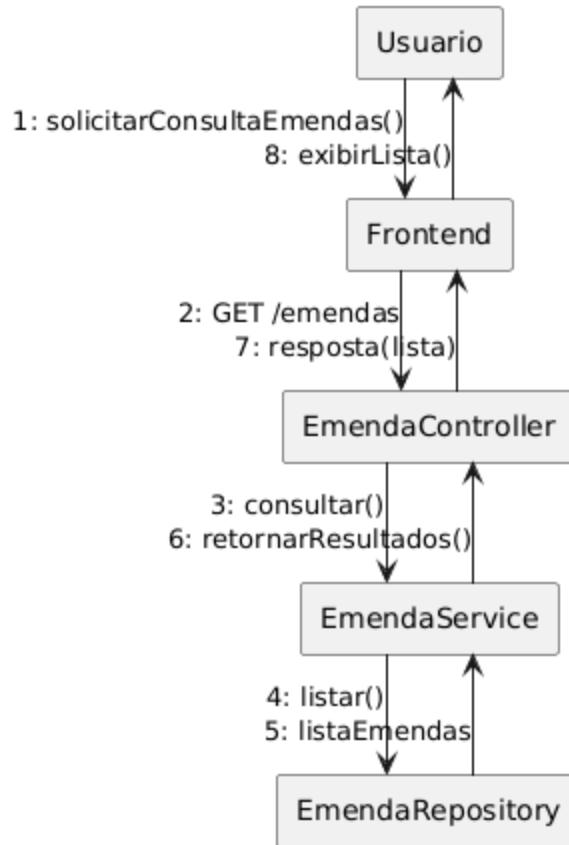


Figura 77 – Diagrama de Comunicação: Consultar Emendas

A Figura 78 representa o diagrama de comunicação da atualização de *status* de emenda, ainda no caso de uso UC05 – Gerenciar Emendas. O *Frontend* envia o identificador e o novo *status* ao *EmendaController*. O *EmendaService* localiza a emenda, verifica a transição de *status* e atualiza o registro no repositório *Emenda*, registrando a alteração no histórico antes de responder ao usuário.

Comunicação - UC05 Atualizar Status da Emenda

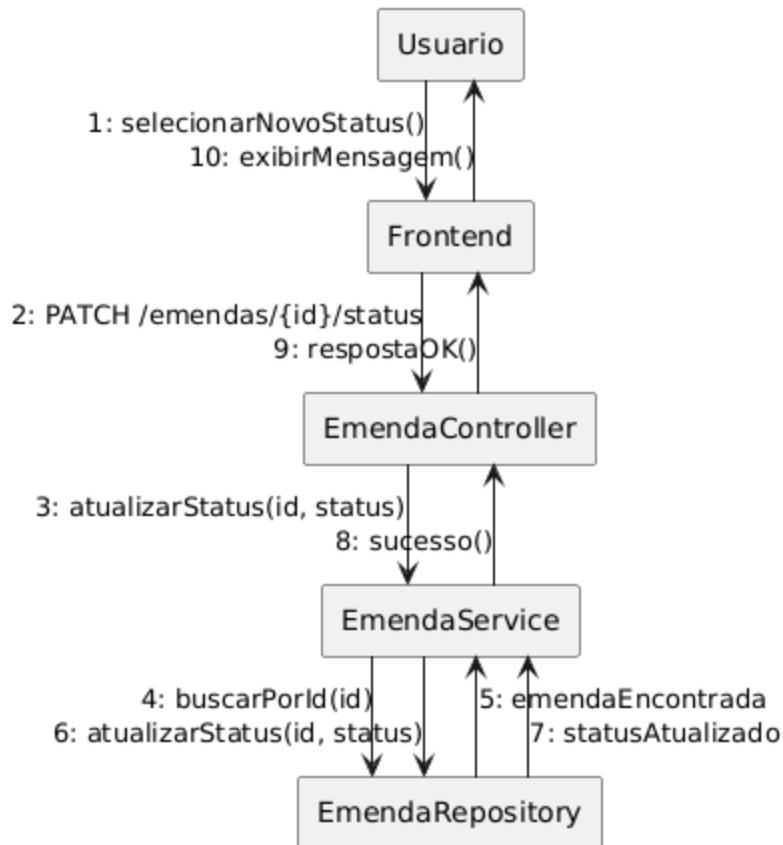


Figura 78 – Diagrama de Comunicação: Consultar Emendas

Como mostra a Figura 79, o diagrama de comunicação da operação de abertura de demanda, ligada ao caso de uso UC06 – Gerenciar Demandas, evidencia um fluxo mais rico: o *Frontend* envia os dados ao *DemandController*, que valida permissões e realiza a checagem de campos. O *DemandService* verifica a existência de cidade e instituição associadas, cria a demanda no repositório *Demand* e aciona o *HistoryService* para registrar a criação. A confirmação retorna ao usuário com o identificador da nova demanda.

Comunicação - UC06 Abrir Demanda

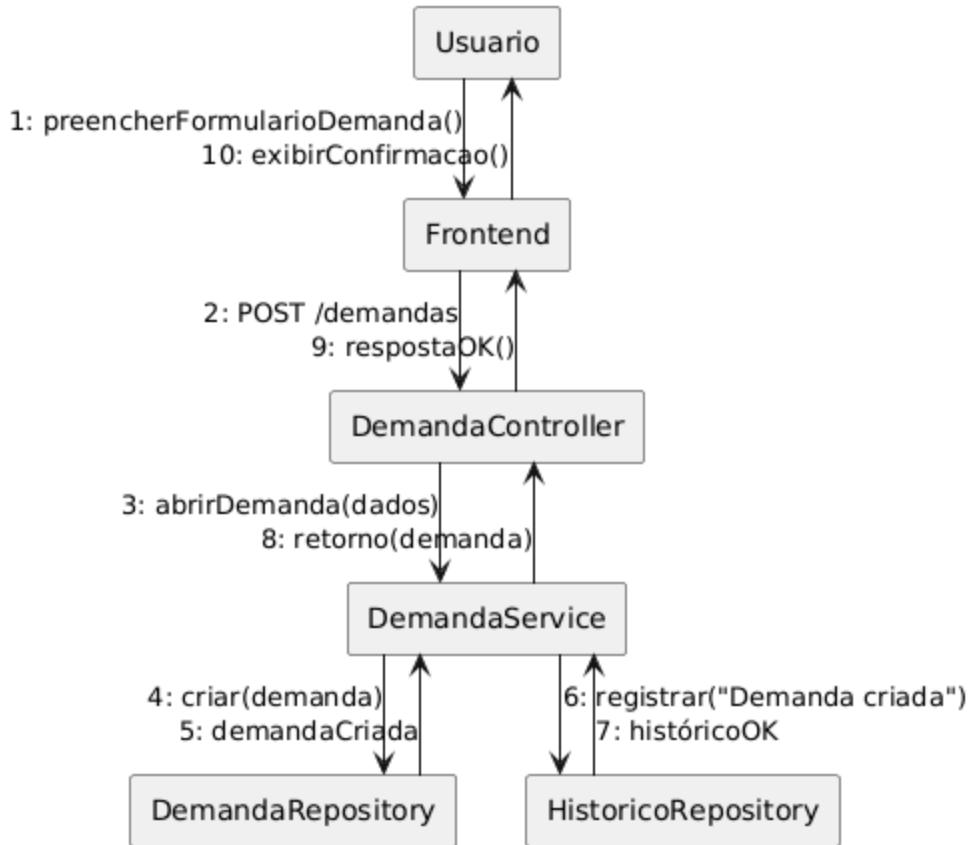


Figura 79 – Diagrama de Comunicação: Abrir Demanda

Na Figura 80, o diagrama de comunicação de consulta de demandas, também no contexto do caso de uso UC06 – Gerenciar Demandas, mostra o Frontend requisitando a lista ao DemandController, que valida permissões e passa os filtros ao DemandService. O serviço consulta o repositório Demanda e devolve a lista de registros ao usuário, permitindo filtros por *status*, prioridade, município, responsável e outros atributos.

Comunicação - UC06 Consultar Demandas

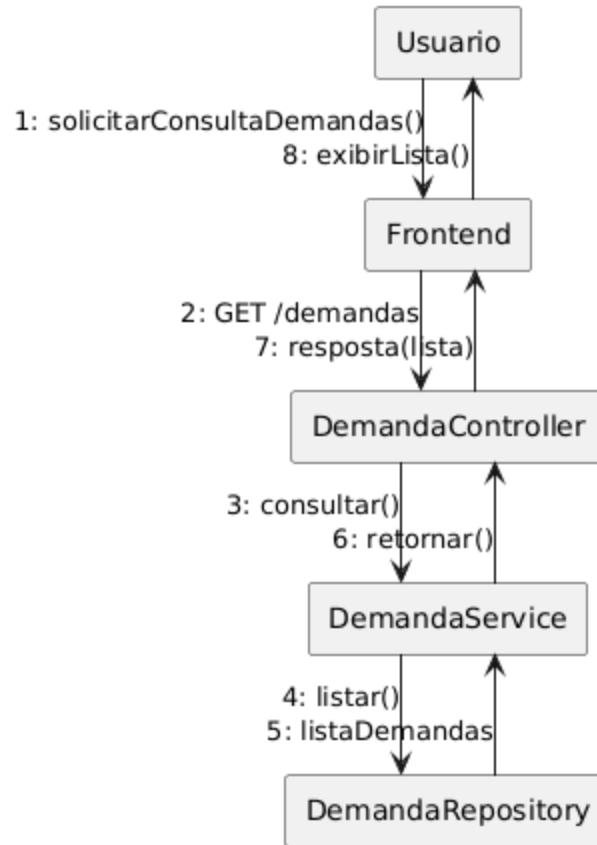


Figura 80 – Diagrama de Comunicação: Consultar Demandas

A Figura 81 apresenta o diagrama de comunicação da operação de atualização de *status* de demanda, ainda dentro do caso de uso UC06 – Gerenciar Demandas. O *Frontend* envia o ID da demanda e o novo status, o *DemandController* valida permissões e verifica se a transição é permitida. O *DemandService* atualiza o registro no repositório *Demand* e registra a mudança no *HistoryService*, retornando o resultado ao usuário.

Comunicação - UC06 Atualizar Status da Demanda

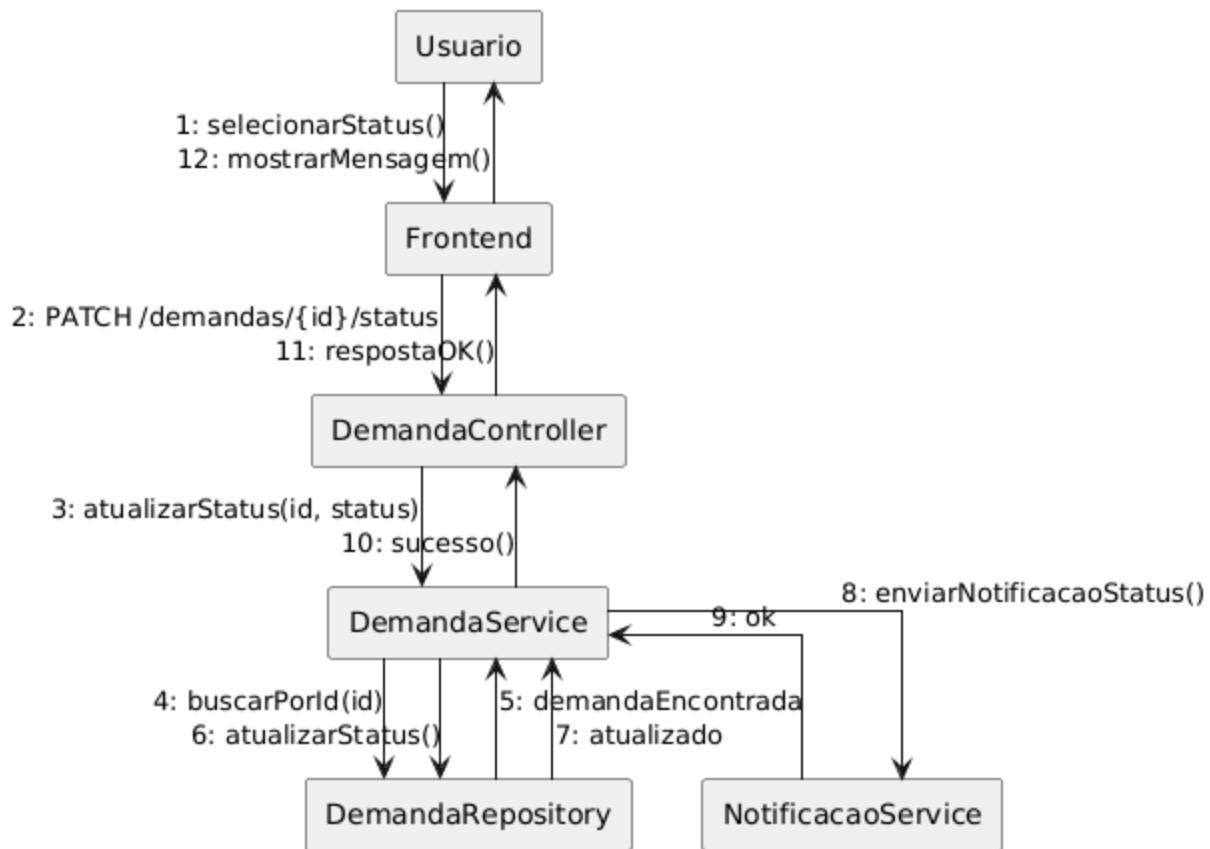


Figura 81 – Diagrama de Comunicação: Atualizar Status Demandada

Como mostra a Figura 82, o diagrama de comunicação da operação de atribuição de responsável, também pertencente ao caso de uso UC06 – Gerenciar Demandas, exibe o *Frontend* acionando o *DemandController* com o ID da demanda e o identificador do usuário responsável. Após validação de permissão e dados, o *DemandService* consulta o repositório de usuários para garantir que o responsável existe, atualiza o campo correspondente na demanda e registra a alteração no histórico.

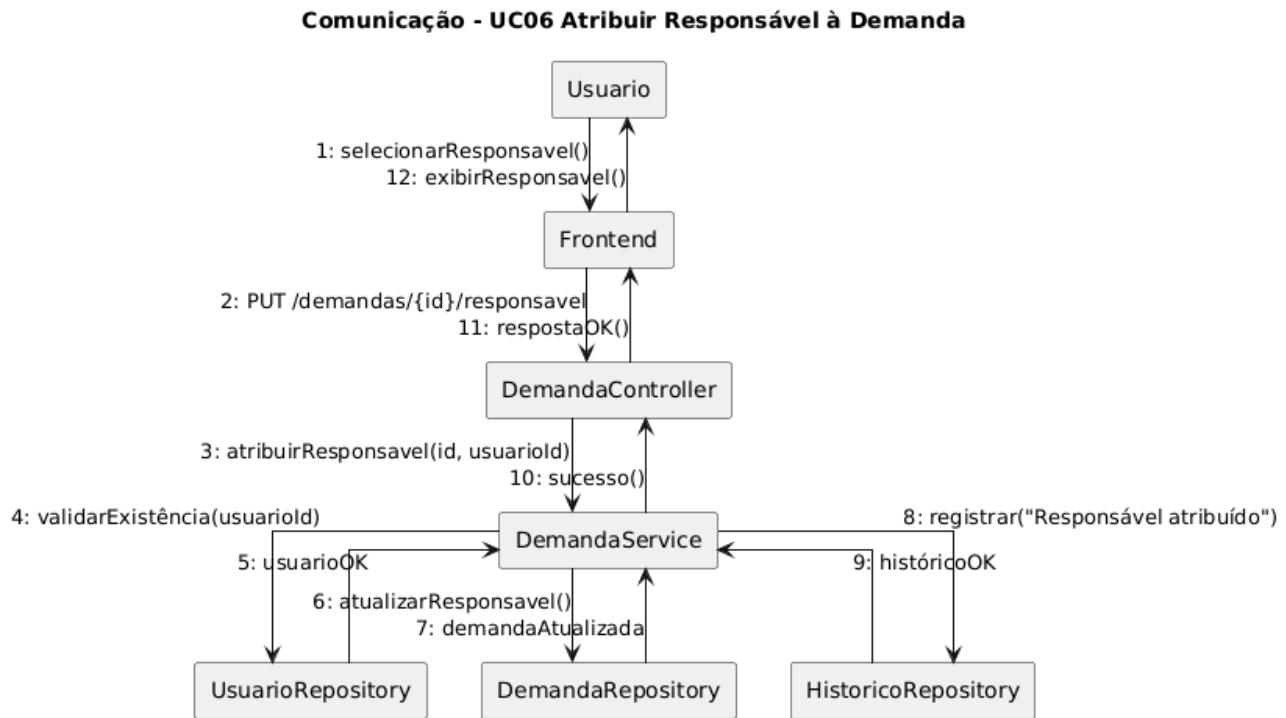


Figura 82 – Diagrama de Comunicação: Atribuir Responsável

Na Figura 83, o diagrama de comunicação de registro de pessoa atendida, ainda no caso de uso UC06 – Gerenciar Demandas, mostra o *Frontend* enviando os dados da pessoa ao *DemandController*. O *DemandService* valida as informações (como nome e contato), associa esses dados à demanda no repositório *Demand* e registra o evento no *HistoryService*, retornando a confirmação ao usuário.

Comunicação - UC06 Registrar Pessoa Atendida

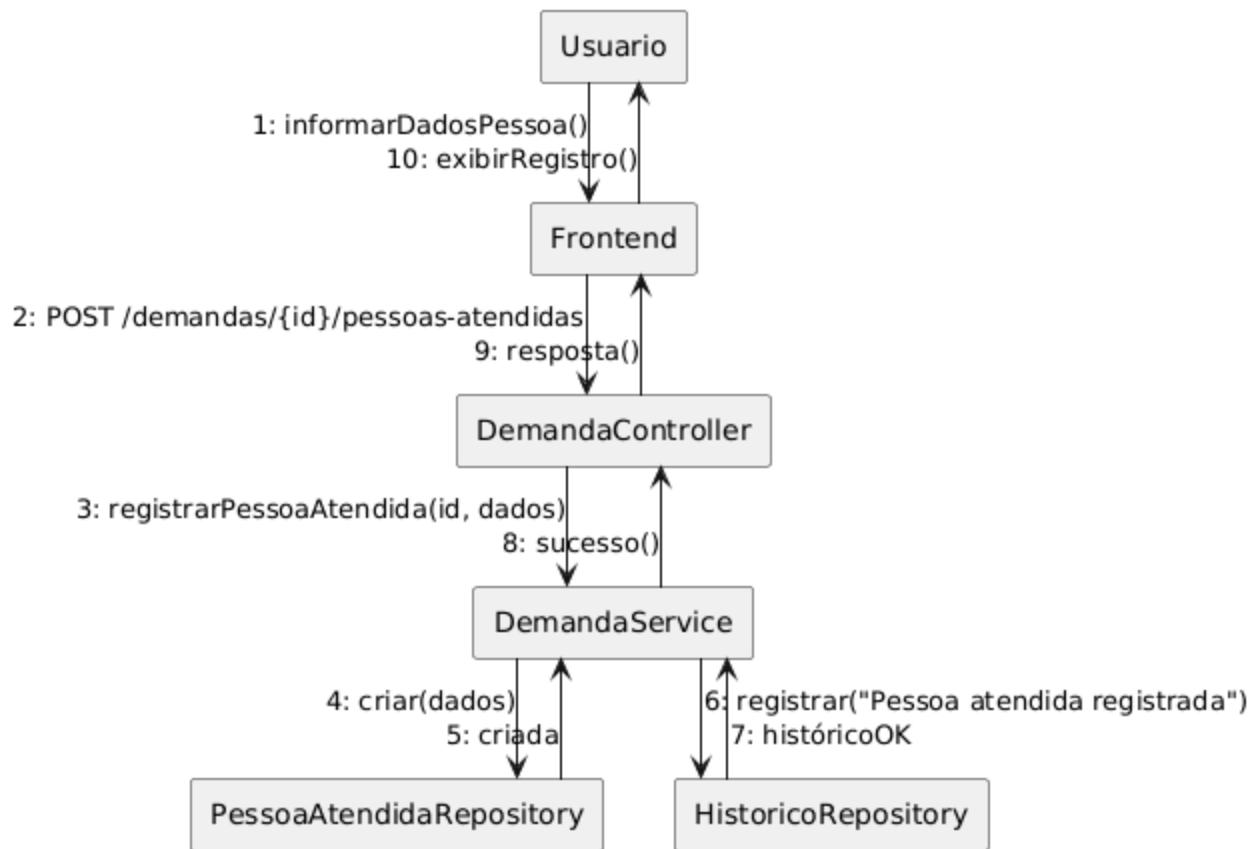


Figura 83 – Diagrama de Comunicação: Registrar Pessoa Atendida

A Figura 84 apresenta o diagrama de comunicação da operação de definição de prazo, vinculada ao caso de uso UC06 – Gerenciar Demandas. O usuário informa a nova data pelo *Frontend*; o *DemandController* valida permissão e aciona o *ValidationService* para verificar a coerência da data. O *DemandService* atualiza o prazo no repositório *Demand* e registra a mudança no histórico antes de devolver o resultado ao *Frontend*.

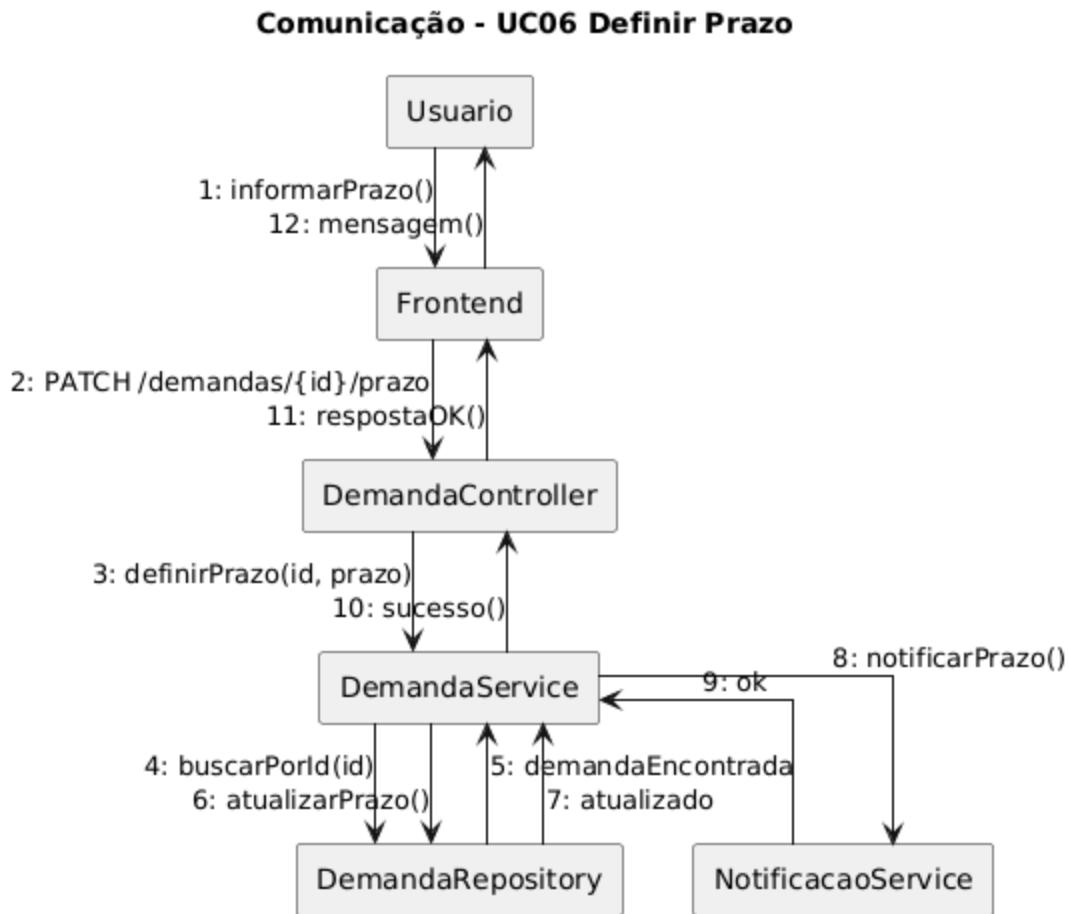


Figura 84 – Diagrama de Comunicação: Definir Prazo

Como mostra a Figura 85, o diagrama de comunicação da operação de anexar ofício, um dos fluxos mais complexos do caso de uso UC06 – Gerenciar Demandas, envolve, além do **DemandeController** e do **DemandeService**, o **ArquivoService**. O **Frontend** envia o arquivo e o identificador da demanda; o serviço de armazenamento grava o documento em S3 ou disco local e retorna o caminho ao **DemandeService**, que cria o registro de anexo no repositório apropriado e registra o evento no histórico. A confirmação é devolvida ao usuário.

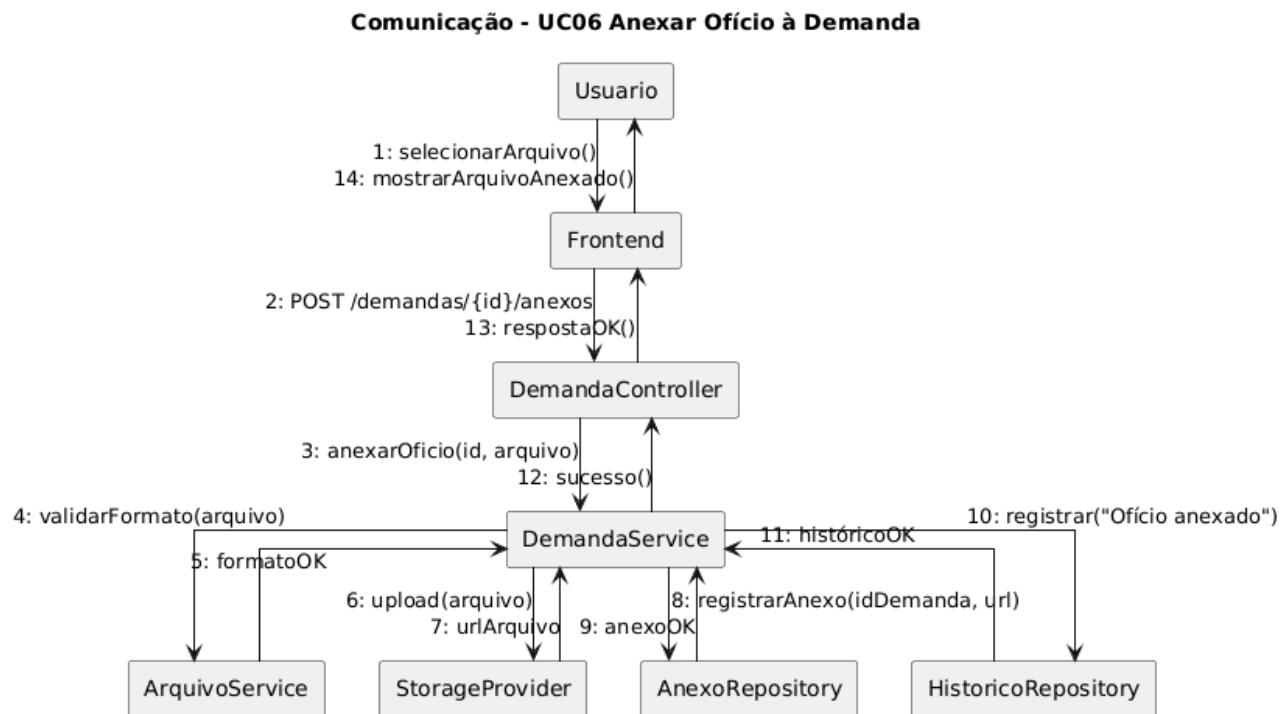


Figura 85 – Diagrama de Comunicação: Anexar Ofício

Na Figura 86, o diagrama de comunicação de cadastro de evento, ligado ao caso de uso UC07 – Gerenciar Agenda, ilustra o usuário preenchendo os dados no Frontend, o EventoController validando permissões e o EventoService persistindo o evento no repositório Evento. Quando aplicável, o serviço também verifica conflitos de agenda e registra o histórico. O resultado é retornado à interface.

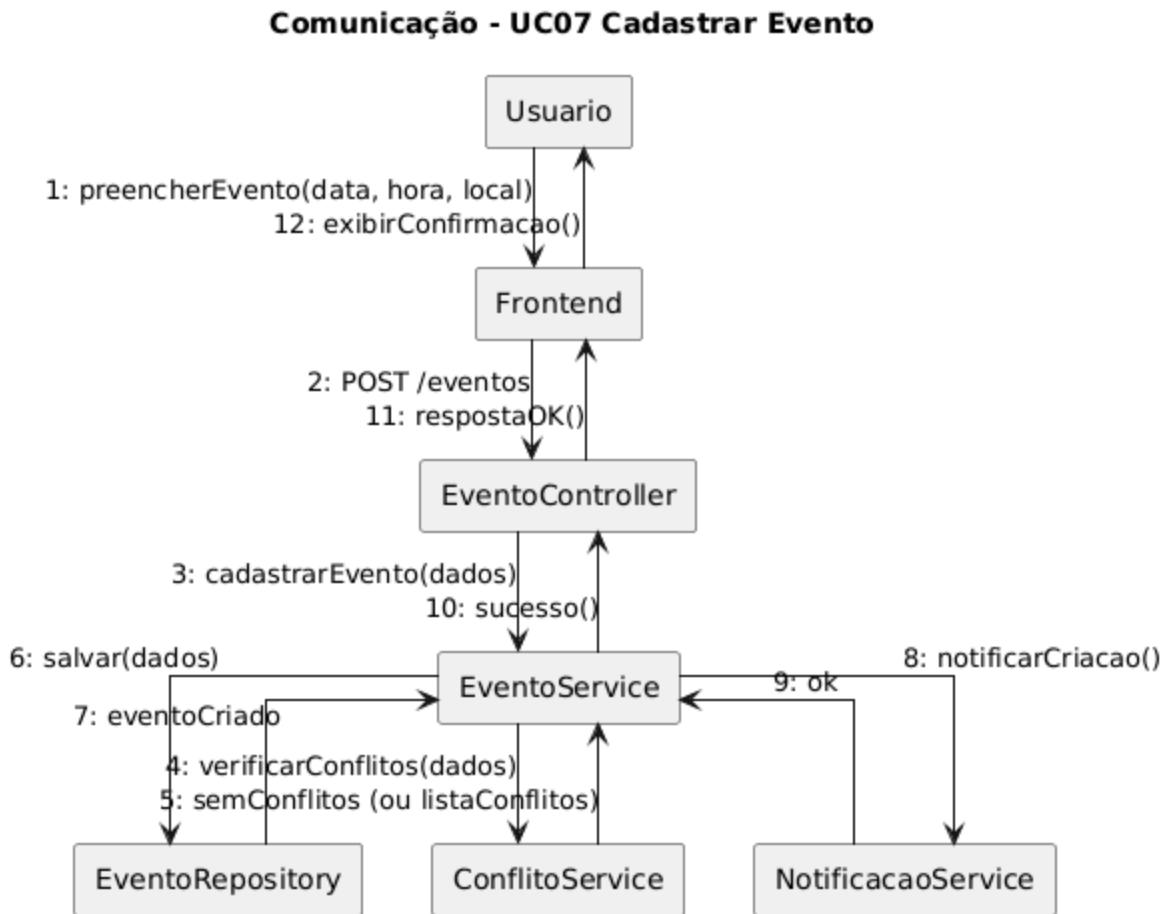


Figura 86 – Diagrama de Comunicação: Cadastrar Evento

A Figura 87 apresenta o diagrama de comunicação da consulta de agenda, também pertencente ao caso de uso UC07 – Gerenciar Agenda. O *Frontend* solicita os eventos de um período específico ao *AgendaController*, que valida permissões e encaminha a requisição ao *AgendaService*. O serviço consulta o repositório *Evento* com filtros de data e retorna a lista de compromissos, exibida em formato de calendário para o usuário.

Comunicação - UC07 Consultar Agenda / Calendário

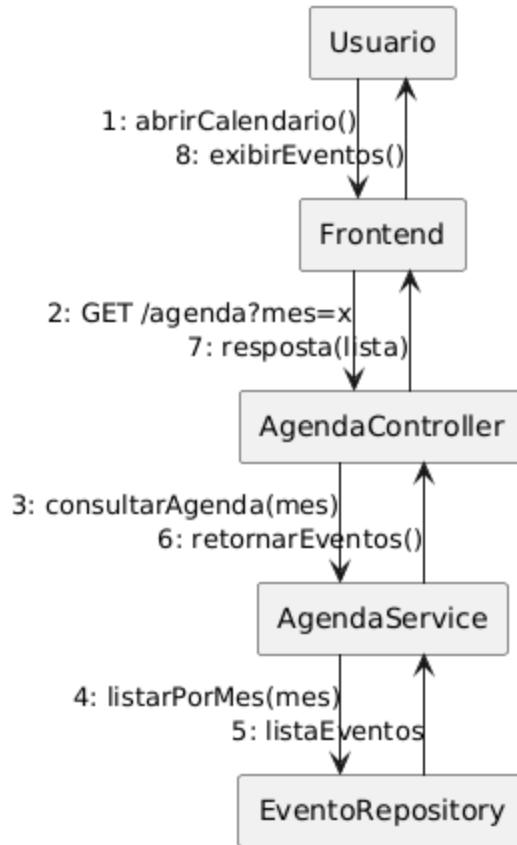


Figura 87 – Diagrama de Comunicação: Consultar Agenda

Como mostra a Figura 88, o diagrama de comunicação da operação de criação de alerta, ainda no caso de uso UC07 – Gerenciar Agenda, mostra o Frontend enviando os dados do alerta associado a um evento. O AlertaController valida a permissão, o AlertaService garante que o evento existe e registra o alerta, acionando o envio de notificações. O usuário recebe a confirmação ao final.

Comunicação - UC07 Criar Alerta/Lembrete

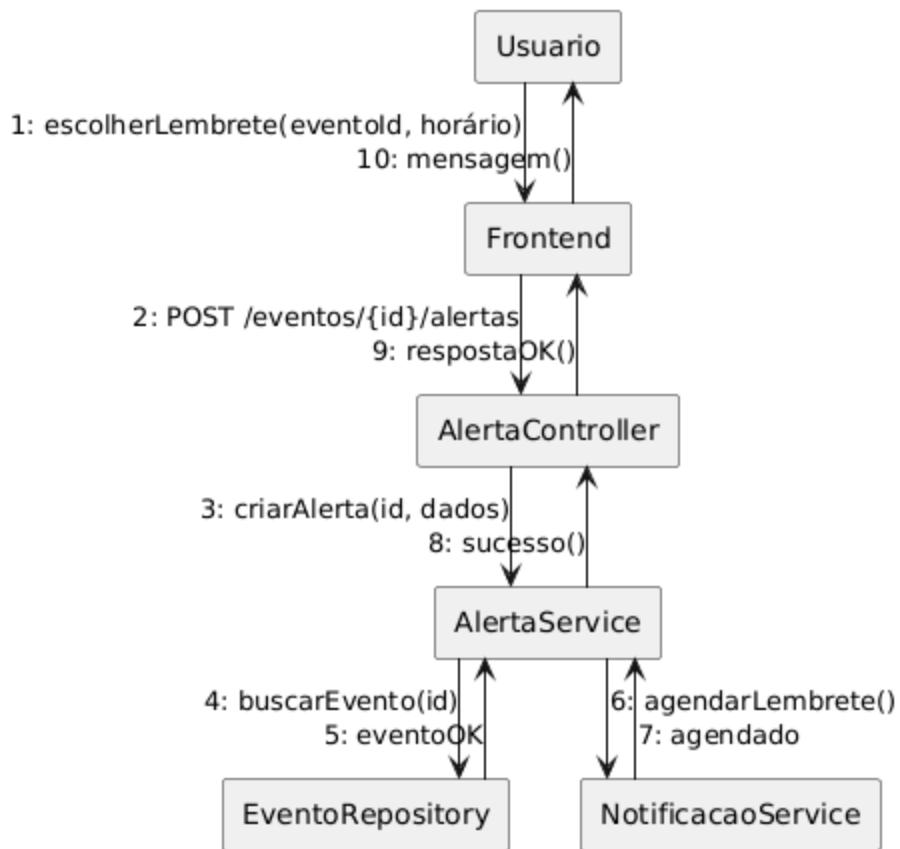


Figura 88 – Diagrama de Comunicação: Criar Alerta

Na Figura 89, o diagrama de comunicação da geração de relatório de Projetos de Lei, relacionado ao caso de uso UC08a – Relatório de Projetos de Lei, mostra o *Frontend* solicitando o relatório ao *RelatorioPLController*. O controlador verifica se o usuário possui permissão específica para esse tipo de relatório e aciona o *RelatorioPLService*, que consulta o repositório *ProjetoLei*, consolida os dados e gera o arquivo (por exemplo, PDF ou planilha). O arquivo é então retornado para download pelo *Frontend*.

Comunicação - UC08a Relatório de Projetos de Lei

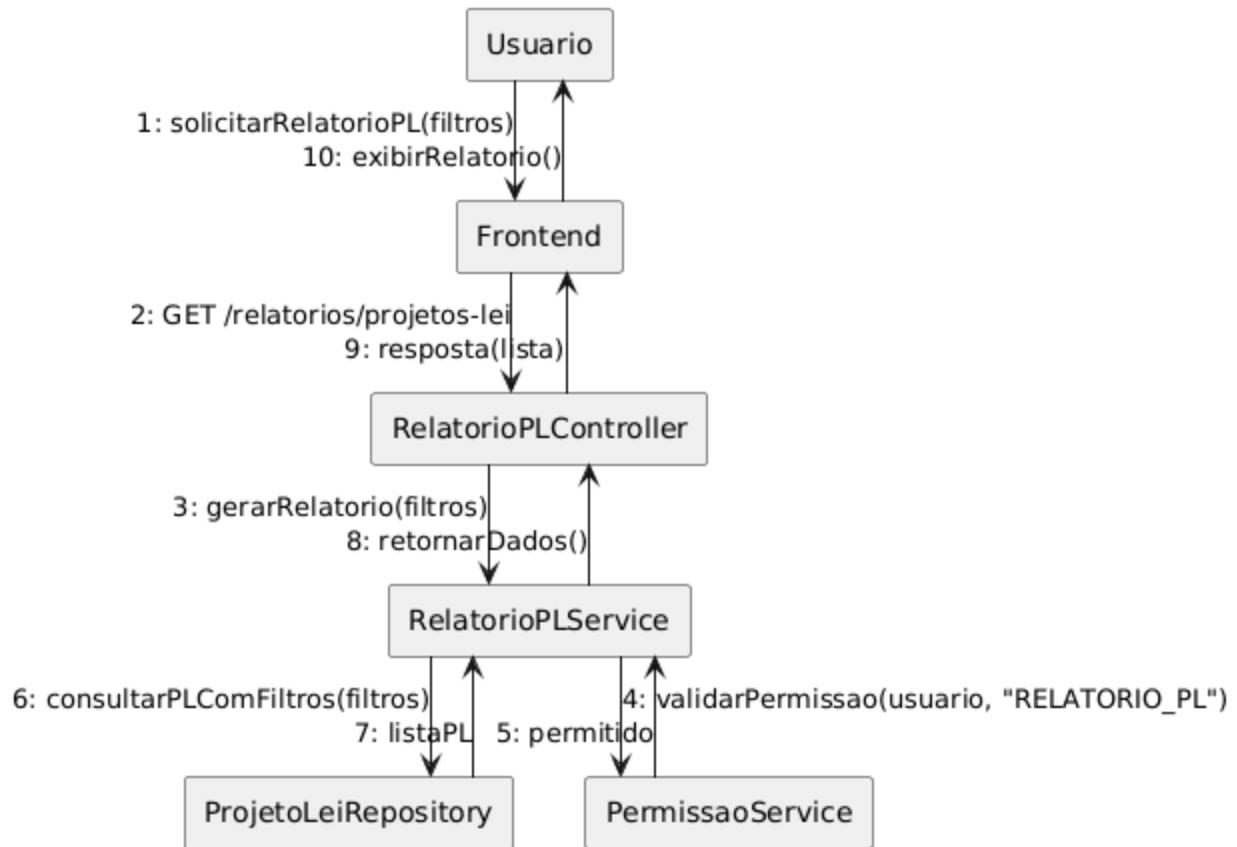


Figura 89 – Diagrama de Comunicação: Gerar Relatório Projetos de Lei

A Figura 90 apresenta o diagrama de comunicação da geração de relatório de emendas, associada ao caso de uso UC08b – Relatório de Emendas. O fluxo é semelhante ao anterior: o *Frontend* faz a requisição ao *RelatorioController*, que valida a permissão e delega ao *RelatorioService*. O serviço consulta o repositório *Emenda*, processa os dados e produz o relatório consolidado, devolvido ao usuário.

Comunicação - UC08b Relatório de Emendas

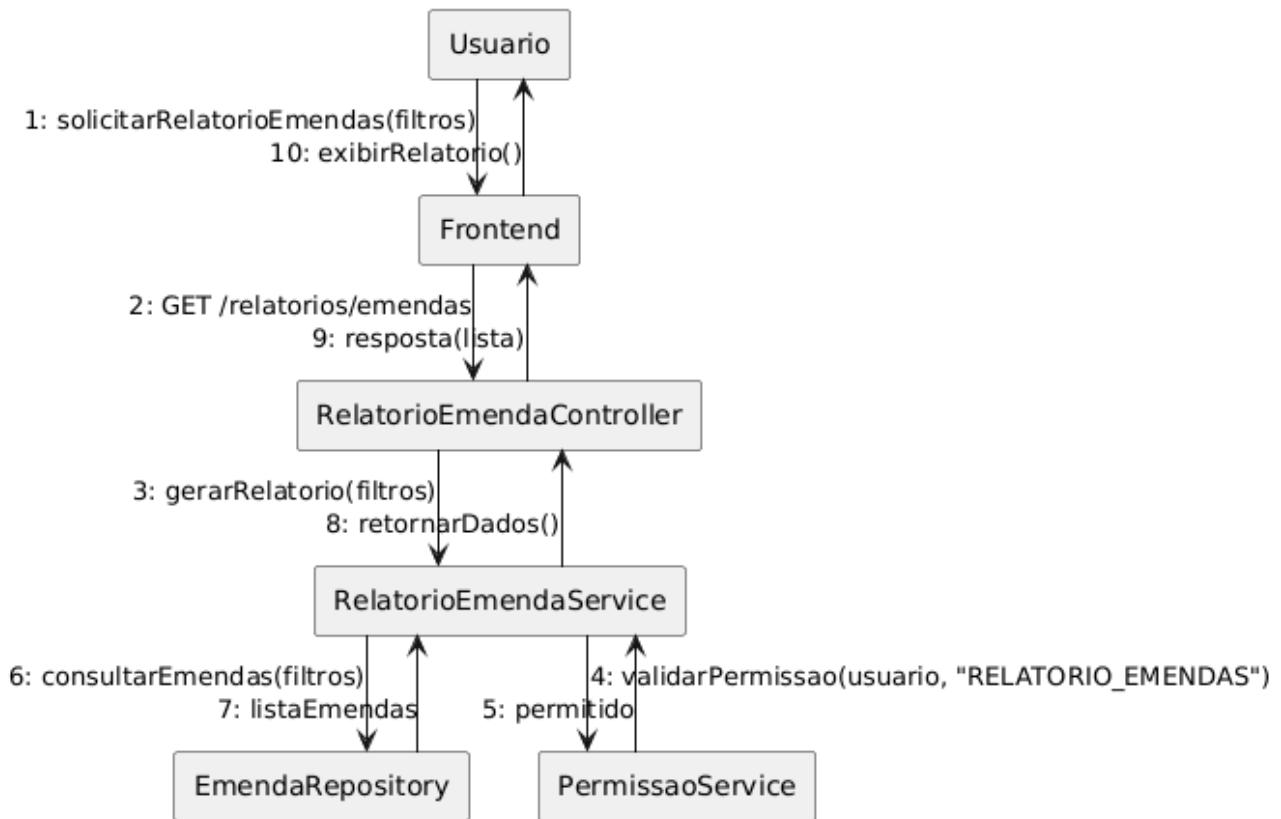


Figura 90 – Diagrama de Comunicação: Gerar Relatório de Emendas

Como mostra a Figura 91, o diagrama de comunicação da geração de relatório de demandas, vinculado ao caso de uso UC08c – Relatório de Demandas, evidencia o *Frontend* solicitando o documento ao *RelatorioController*, que valida permissões específicas para esse tipo de visualização. O *RelatorioService* consulta o repositório *Demandas*, aplica filtros (*status*, *cidade*, *período*) e gera o relatório, disponibilizado para *download* ao usuário autorizado.

Comunicação - UC08c Relatório de Demandas

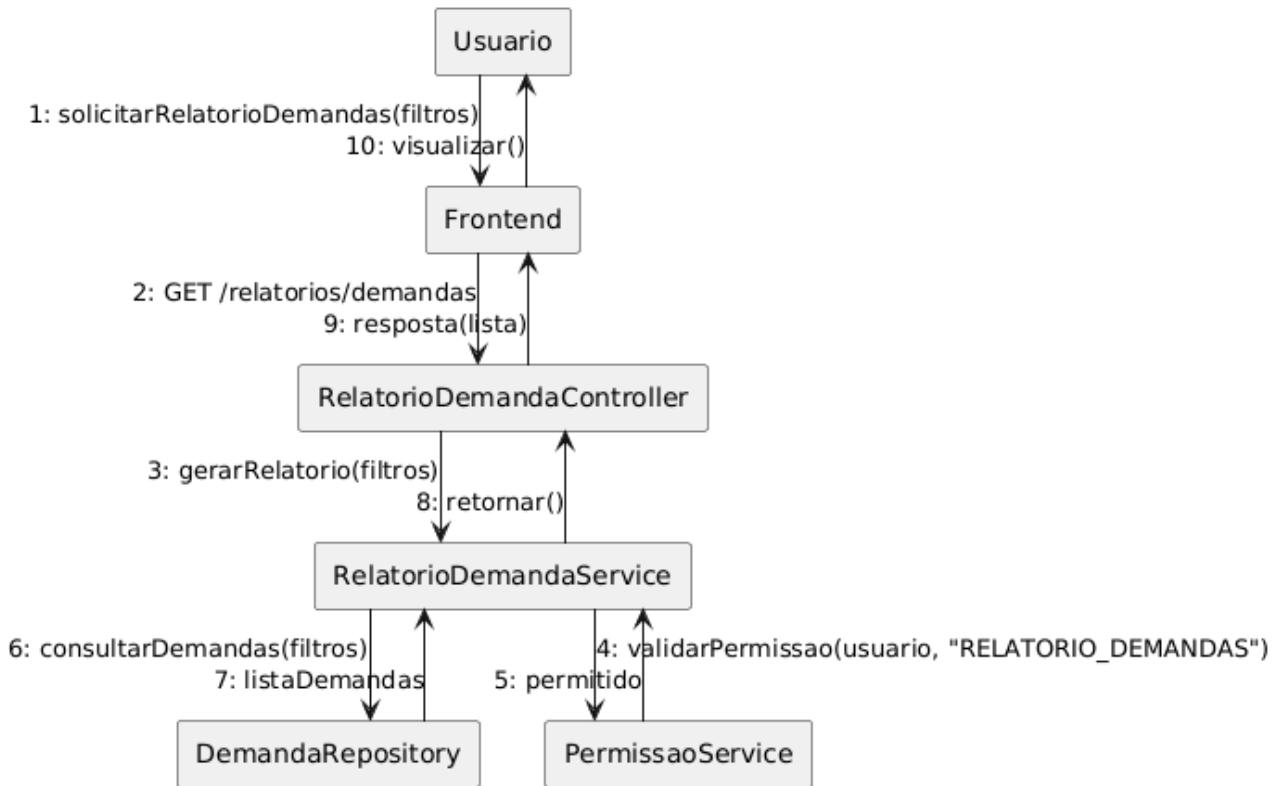


Figura 91 – Diagrama de Comunicação: Gerar Relatório de Demandas

Por fim, a Figura 92 apresenta o diagrama de comunicação da geração do relatório consolidado geral, correspondente ao caso de uso UC08d – Relatório Consolidado Geral. Nesse fluxo, apenas a Deputada têm permissão para acionar o RelatorioController, que delega ao RelatorioService a tarefa de combinar dados de Projetos de Lei, Emendas e Demandas em um único documento estratégico. O serviço consulta múltiplos repositórios, consolida os indicadores e devolve o relatório final ao Frontend, fechando o ciclo de cooperação entre componentes para suporte à decisão política.

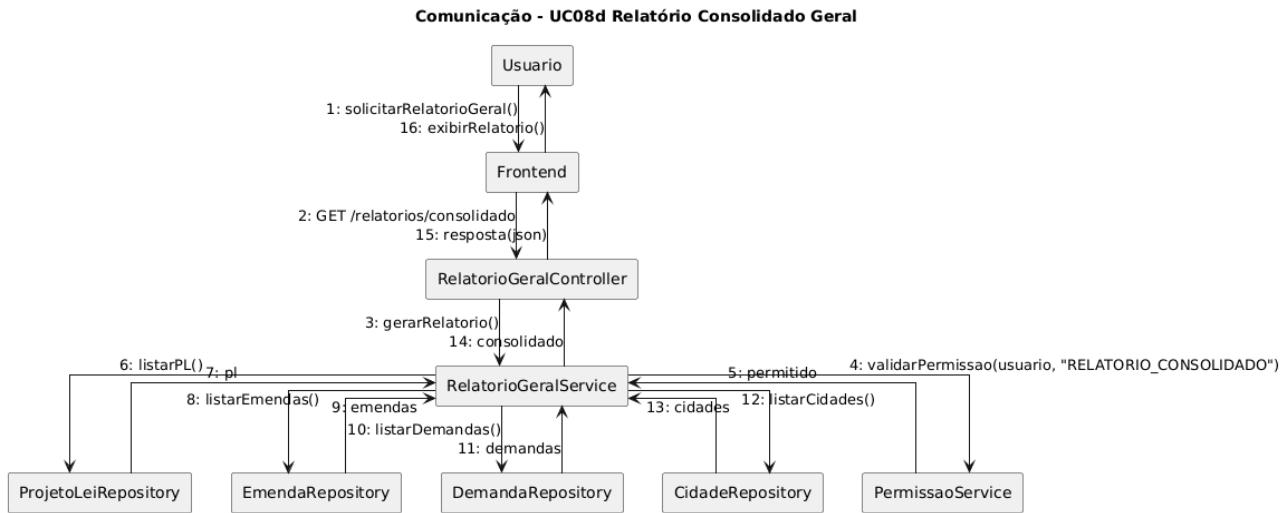


Figura 92 – Diagrama de Comunicação: Gerar Relatório Consolidado

3.4 Arquitetura

A arquitetura lógica do sistema Gabinete Inteligente foi modelada utilizando o C4 Model, uma abordagem moderna para representar sistemas de software em diferentes níveis de abstração. Neste trabalho, foram utilizados dois níveis:

- Nível 1 – Diagrama de Contexto: visão mais ampla do sistema e seus atores externos.
- Nível 2 – Diagrama de Contêineres: visão dos principais blocos de execução internos (frontend, backend, banco de dados etc.).

Cada nível possui rótulos próprios, diagramas independentes e é discutido em subseções específicas, de acordo com a recomendação da banca.

O Diagrama de Contexto (Nível 1), representado na Figura 93, descreve o ambiente geral em que o sistema opera, detalhando quem interage com o Gabinete Inteligente e como essas interações ocorrem. Nesta camada, o sistema é tratado como uma única caixa preta, destacando seus limites e as relações com usuários e serviços externos.

Como mostra o diagrama, os principais atores são:

Deputada: consome informações estratégicas (emendas, agenda, projetos de lei).

Chefe de Gabinete: coordena a agenda e acompanha o andamento das demandas.

Gestora de Demandas: atribui responsáveis, controla prazos e gera relatórios.

Equipe de Atendimento: registra demandas, atualiza informações e interage com lideranças.

Cidadão (Consulta Pública): acessa apenas conteúdos institucionais.

O sistema em si é composto por dois contêineres gerais:

Frontend Web (React): interface que os usuários utilizam no navegador.

Backend API (Laravel): responsável por regras de negócio, persistência de dados e integrações.

Além disso, o sistema se integra com três serviços externos: Banco de Dados (MySQL), Armazenamento de Arquivos (S3/Local) e Serviço de E-mail/Notificações (SMTP/API).

O Nível 1 estabelece o panorama macroscópico, destacando que o fluxo de comunicação ocorre por meio de requisições REST/JSON via HTTPS, sempre mediadas pelo frontend.

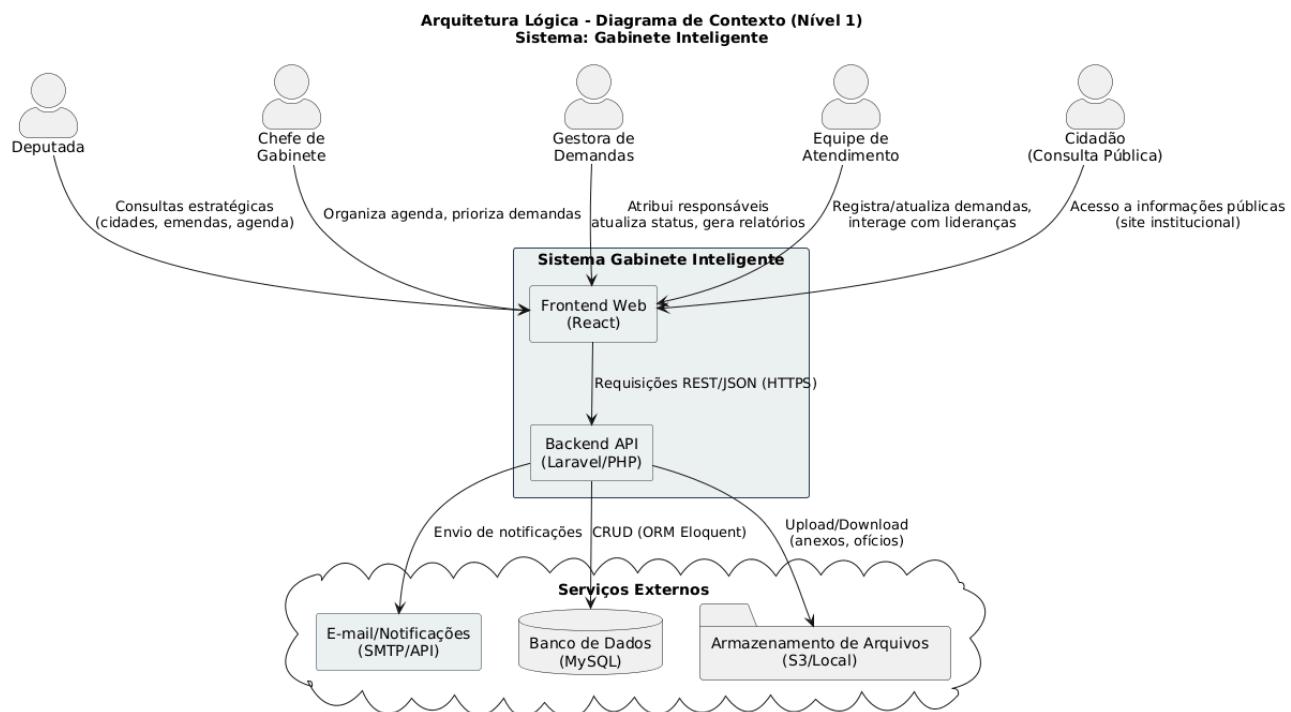


Figura 93 – Diagrama de Contexto

A Figura 94 apresenta o Diagrama de Contêineres (Nível 2), que aprofunda a visão estrutural do sistema, detalhando os principais blocos internos e como eles interagem. Neste nível, a arquitetura é decomposta em contêineres executáveis, cada um com responsabilidades claramente definidas.

Os contêineres principais são:

- Frontend Web (React): Executado no navegador. Responsável por interface, navegação, validações iniciais e envio de requisições para a API. Comunica-se exclusivamente via HTTPS com a API.
- Backend API (Laravel/PHP):

- Este contêiner é subdividido em módulos: Módulo de Autenticação (JWT/Guards): controla login, geração de token e permissões. Módulo de Demandas: abertura, atualização, anexos e notificações. Módulo de Emendas: cadastro, status e consultas. Módulo de Cidades e Instituições: gestão de cadastros auxiliares. Módulo de Agenda: eventos, alertas e verificação de conflitos. Módulo de Relatórios: geração de documentos e exportações.
- A API centraliza toda a lógica de negócio e se comunica com: MySQL: utilizando o ORM Eloquent para operações CRUD. S3/Local Storage: para upload de anexos, ofícios e documentos gerados. SMTP/API: para envio de e-mails e notificações automáticas.
- Banco de Dados (MySQL): Armazena entidades como demandas, emendas, eventos, usuários etc. Cada módulo do backend acessa tabelas específicas, garantindo separação lógica.
- Armazenamento de Arquivos (S3/Local): Responsável por anexos de demandas, relatórios e mídias do site.
- Serviço de E-mail: Utilizado para notificações de prazos, atribuição de responsáveis e alertas da agenda.

O nível 2 demonstra como o sistema é organizado internamente, apresentando uma arquitetura modular, escalável e compatível com boas práticas do Laravel.

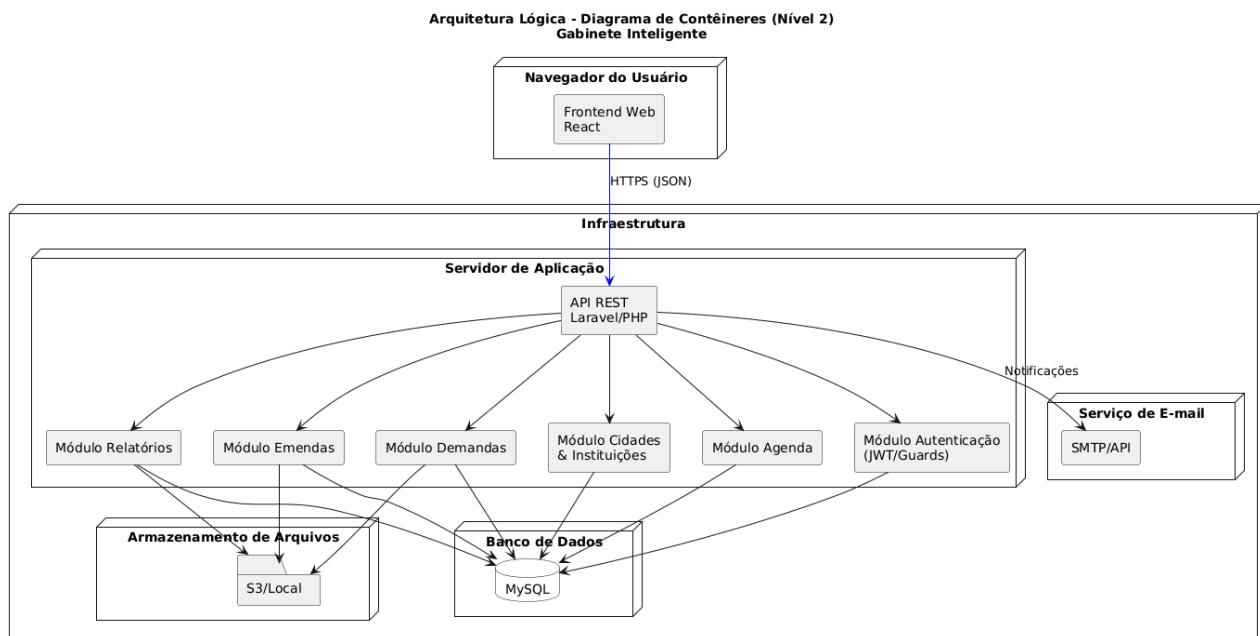


Figura 94 – Diagrama de Contêineres

3.5 Diagramas de Atividade

A presente subseção descreve o Diagrama de Atividades elaborado para o sistema Gabinete Virtual. O objetivo deste diagrama é fornecer uma visão comportamental das etapas que compõem o fluxo de Gerenciamento de Demandas, considerado o processo mais importante e abrangente do sistema. O Diagrama de Atividades apresentado na Figura 95 representa o fluxo operacional do caso de uso principal do sistema, relacionado ao processo de gerenciamento de demandas. O diagrama inicia-se com o acesso do usuário ao sistema, passando pela verificação de autenticação e permissão, até alcançar as etapas que envolvem o registro e acompanhamento das demandas. A partir desse ponto, o fluxo demonstra como o usuário pode registrar uma nova demanda, atualizar informações, realizar atribuições, definir prazos, anexar documentos e alterar o status conforme a evolução do atendimento. Além disso, o diagrama evidencia as decisões condicionais que orientam o comportamento do sistema, como validação de dados obrigatórios, bloqueio por falta de permissão ou impossibilidade de transição de status. O encerramento do fluxo ocorre quando a demanda é finalizada e o processo segue para arquivamento ou geração de relatórios. Dessa forma, a Figura 95 sintetiza, de forma sequencial, as ações possíveis e as regras que regulam o ciclo de vida de uma demanda no Gabinete Virtual.

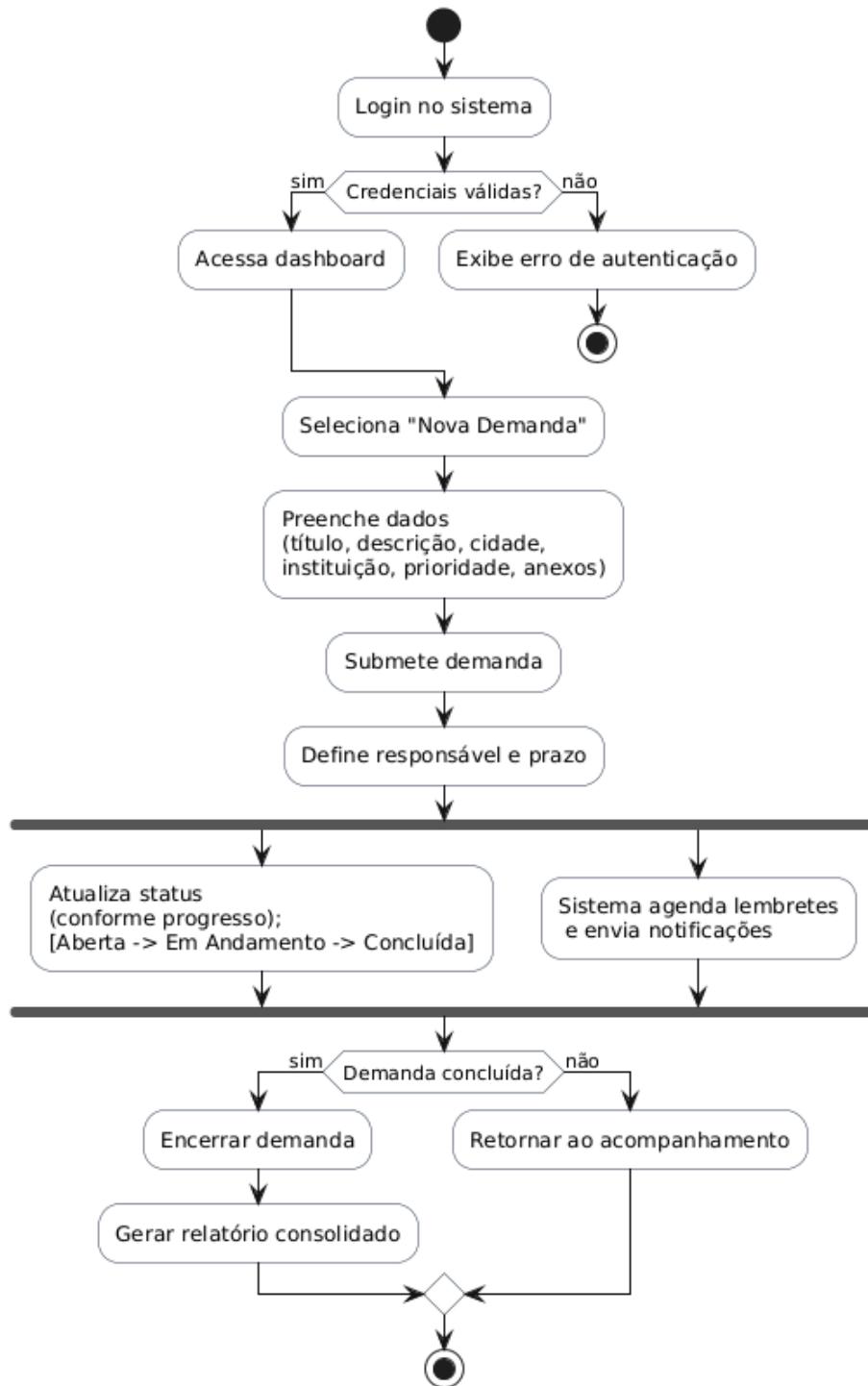
Diagrama de Atividades - Caso Principal: Gerenciar Demanda

Figura 95 – Diagrama de Atividades

3.6 Diagrama de Componentes e Implantação.

Esta seção apresenta a descrição dos Diagramas de Componentes e de Implantação desenvolvidos para o sistema Gabinete Virtual. Tais diagramas representam a visão física e estrutural da aplicação, evidenciando como os módulos internos se organizam e como o sistema é distribuído em ambiente de execução.

O Diagrama de Implantação da Figura 96 representa a estrutura física e tecnológica que suporta o sistema. Ele é composto pelos seguintes nós:

- Servidor de Frontend (Navegador Web)

Onde o React é renderizado e realiza as requisições HTTPS para a API.

- Servidor Backend (Laravel/PHP)

Hospeda os controladores, serviços, repositórios e toda a lógica de negócio do sistema. Comunica-se diretamente com o banco de dados, serviço de armazenamento e servidor de e-mails.

- Servidor de Banco de Dados (MySQL)

Responsável por persistir todas as entidades principais: usuários, cidades, instituições, lideranças, demandas, eventos, PLs, emendas, textos institucionais e históricos.

- Servidor de Armazenamento (S3/Local Storage)

Utilizado para guardar anexos de demandas, imagens do site e arquivos gerados por relatórios.

- Servidor de E-mail/Notificações

Responsável pelo envio de alertas automáticos, avisos de prazos, lembretes e validações de segurança.

O diagrama reforça que toda a comunicação ocorre via protocolos seguros (HTTPS) e que o backend atua como intermediário entre o frontend e todos os demais serviços, garantindo centralização das regras de negócio e consistência das operações.

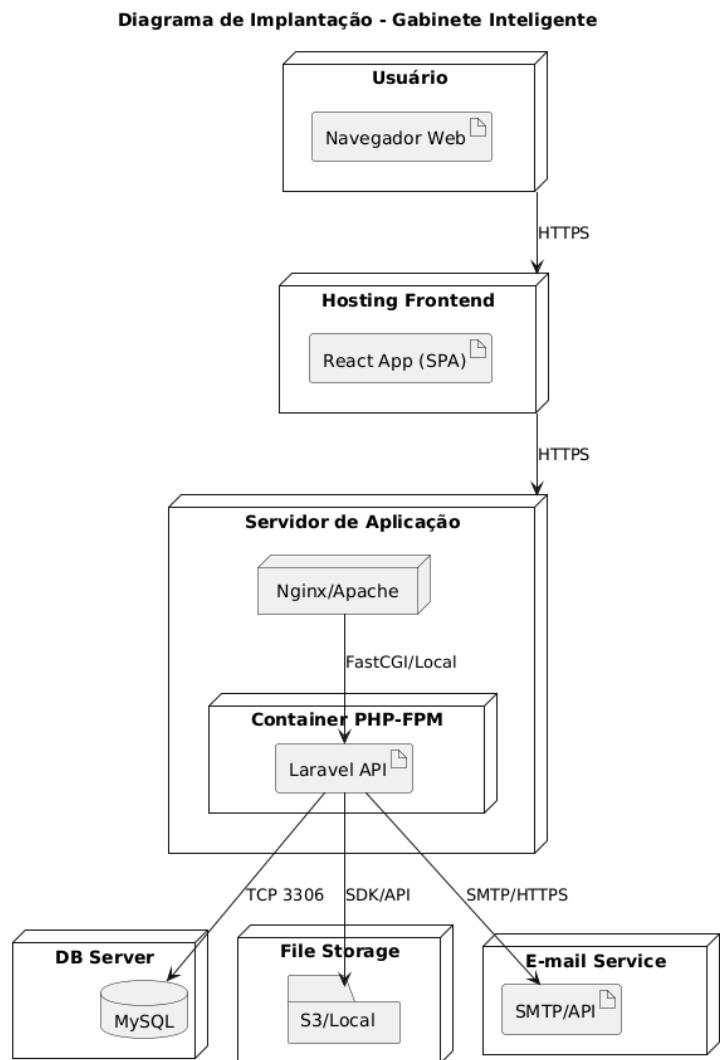


Figura 96 – Diagrama de Implantação

O Diagrama de Componentes, representado na Figura 97, demonstra a divisão modular da aplicação no backend Laravel. Os principais componentes identificados são:

- Compartimento de Autenticação:
Responsável pelo login, geração e validação de tokens JWT, verificações de perfil e autorização de rotas. É utilizado transversalmente por todos os demais módulos.
- Compartimento de Cadastros:
Inclui classes e serviços para gerenciar Instituições, Lideranças, Municípios e Igrejas. Este componente fornece operações CRUD e validações específicas.
- Compartimento de Demandas:

Abrange abertura, atualização, atribuição de responsáveis, anexos, prazos e históricos. É o módulo mais complexo e integra serviços auxiliares como armazenamento de arquivos e notificações.

- Componente de Emendas e Projetos de Lei

Responsável pela gestão de emendas parlamentares e PLs, incluindo status, vínculos com municípios e geração de relatórios legislativos.

- Componente de Agenda

Gerencia eventos, conflitos, alertas e integração com o serviço de e-mail para envio de lembretes automáticos.

- Componente de CMS (Conteúdo do Site)

Trata das notícias e seções institucionais (Missão, Trajetória, Projetos do Site).

- Componente de Relatórios

Centraliza a geração de documentos consolidados (demandas, emendas, PLs e relatórios gerais).

O diagrama evidencia a independência lógica entre módulos, mas também a comunicação entre eles por meio de serviços compartilhados (Histórico, Autorização, Armazenamento, Validação).

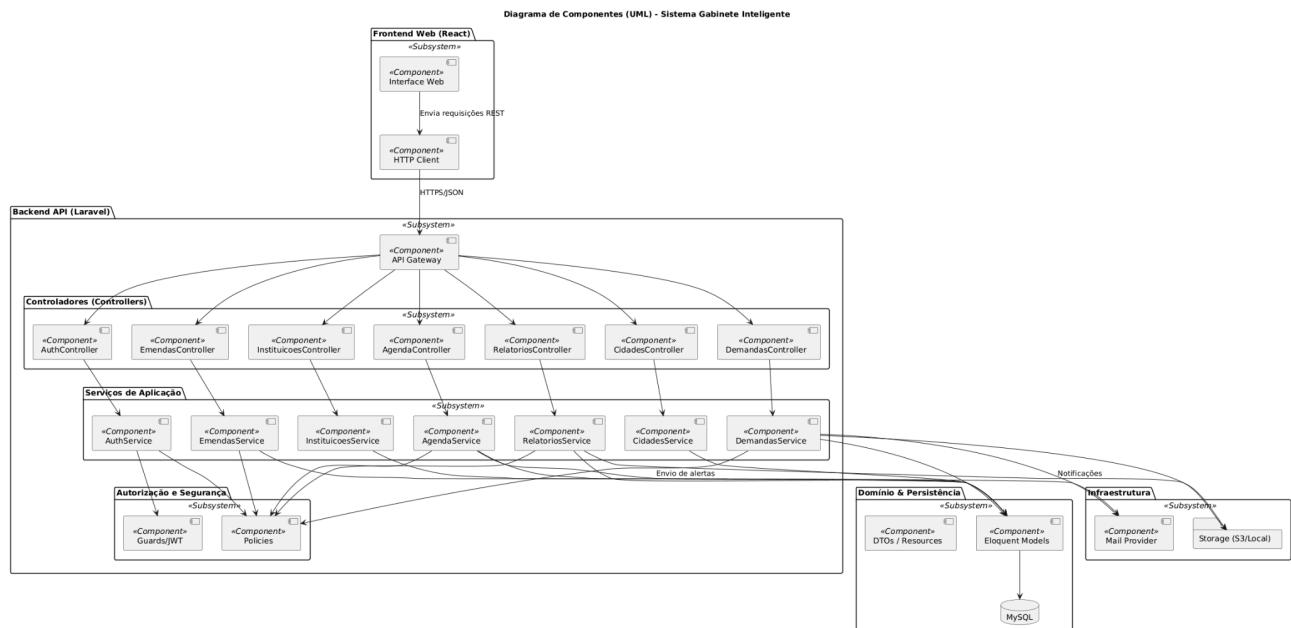


Figura 97 – Diagrama de Componentes

4. Projeto de Interface com Usuário

O projeto de interface define a experiência do usuário e o *design* das telas que compõem o sistema Gabinete Virtual. Nesta seção, são apresentados *wireframes* das principais telas do sistema, que servem como protótipos visuais para orientar a implementação do *frontend* React.

Os *wireframes* foram elaborados considerando os requisitos de usabilidade levantados nas personas, priorizando simplicidade, clareza e eficiência no uso diário do sistema.

4.1 Esboço das Interfaces

As imagens abaixo representam um esboço das principais telas do sistema. Foram feitos *wireframes* para permitir uma visualização inicial do conceito e usabilidade da aplicação. Portanto, ainda que bem desenhados, podem não representar a interface final do sistema.

A Figura 98 representa a tela de autenticação do sistema, primeira interface de contato de todos os usuários internos com a plataforma. Esta tela apresenta um layout centralizado e minimalista, contendo campos para e-mail institucional e senha, além de um botão primário para efetuar o login. O design prioriza simplicidade e clareza, reduzindo barreiras de entrada ao sistema.

Perfis com acesso: Todos os perfis internos (Deputada, Chefe de Gabinete, Gestora de Demandas, Equipe de Atendimento e Funcionários).

Elementos principais: Campos de e-mail e senha, botão "Entrar", credenciais de demonstração para ambiente de testes.

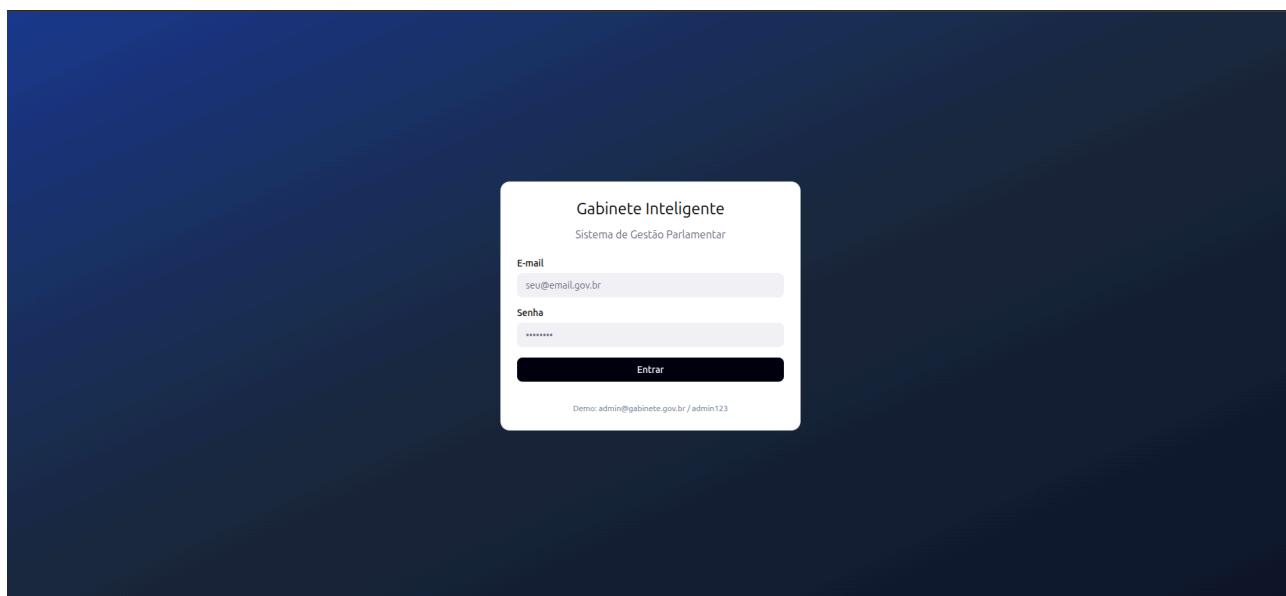


Figura 98 – Tela de Login

Como mostra a Figura 99, o Dashboard constitui a tela inicial após autenticação bem-sucedida, oferecendo uma visão consolidada das principais métricas do gabinete. A interface exibe cards informativos com totalizadores de demandas ativas, projetos de lei, emendas, eventos programados, instituições cadastradas e taxa de resolução. Complementam a tela painéis de atividades recentes e atalhos rápidos para as funcionalidades mais utilizadas, permitindo navegação eficiente.

Perfis com acesso: Deputada, Chefe de Gabinete, Gestora de Demandas.

Elementos principais: Indicadores quantitativos, painel de atividades recentes, menu lateral de navegação, atalhos para ações frequentes.

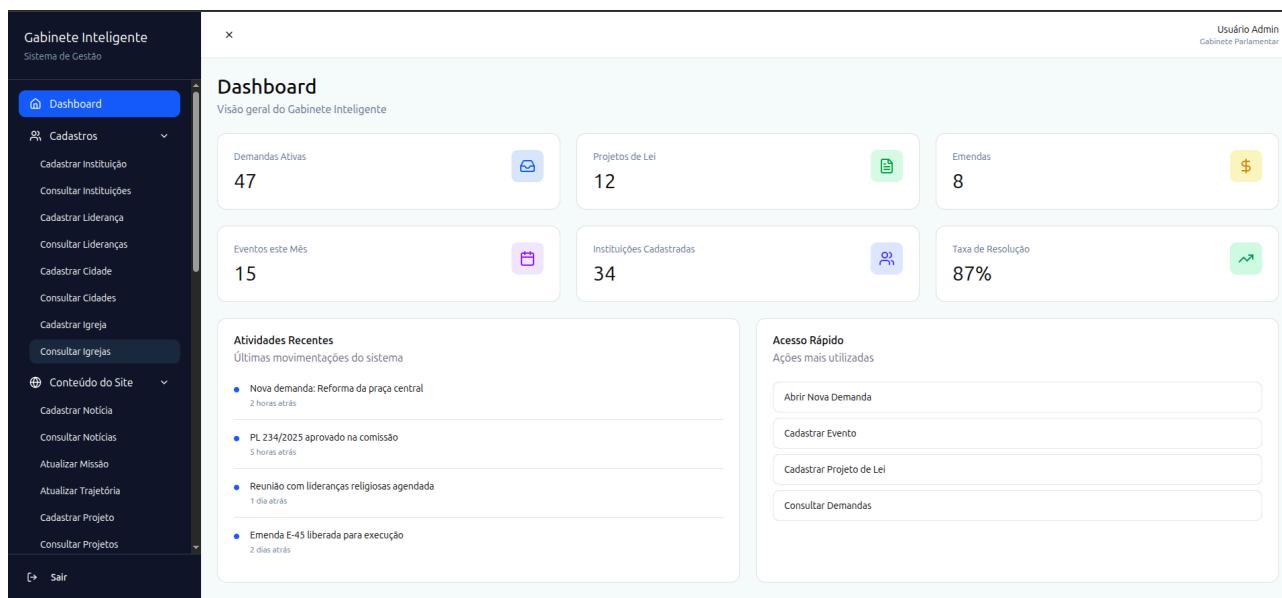


Figura 99 – Tela Inicial - Dashboard

A Figura 100 apresenta o formulário de cadastro de instituições, contendo campos para nome da instituição, tipo (Prefeitura, Câmara Municipal, Órgão Estadual, ONG/Associação, Sindicato) e município vinculado. A interface inclui botões de ação primária "Cadastrar Instituição" e secundária "Ver Todas", além de validações em tempo real.

Perfis com acesso: Chefe de Gabinete, Gestora de Demandas, Funcionários.

Elementos principais: Campos de texto, seletores de tipo e município, botões de ação.

Figura 100 – Tela de Cadastro de Instituição

Na Figura 101, observa-se a interface de consulta que permite filtrar instituições por nome, tipo e município. Os resultados são exibidos em tabela com colunas de nome, tipo, município e ações (editar/excluir). Um botão "Nova Instituição" possibilita acesso rápido ao cadastro.

Perfis com acesso: Todos os perfis internos.

Elementos principais: Filtros de busca, tabela de resultados, botões de ação por linha, botão de novo cadastro.

The screenshot shows the 'Consultar Instituições' (Search Institutions) page. The left sidebar has a 'Consultar Instituições' button highlighted. The main area has a search bar and dropdowns for 'Nome', 'Tipo', and 'Município'. Below is a table with columns 'Nome', 'Tipo', 'Município', and 'Ações' (Actions). The table lists five entries:

Nome	Tipo	Município	Ações
Prefeitura Municipal de Campinas	Prefeitura	Campinas	<input checked="" type="checkbox"/> <input type="checkbox"/>
Câmara Municipal de Santos	Câmara Municipal	Santos	<input checked="" type="checkbox"/> <input type="checkbox"/>
Secretaria Estadual de Saúde	Órgão Estadual	São Paulo	<input checked="" type="checkbox"/> <input type="checkbox"/>
Associação de Moradores do Bairro Alto	ONG / Associação	Sorocaba	<input checked="" type="checkbox"/> <input type="checkbox"/>
Sindicato dos Professores	Sindicato	Ribeirão Preto	<input checked="" type="checkbox"/> <input type="checkbox"/>

Figura 101 – Tela de Consulta de Instituições

A Figura 102 demonstra o formulário para registro de lideranças políticas e comunitárias, com campos para nome completo, cargo/função, telefone e instituição vinculada. A vinculação com instituição pré-cadastrada garante integridade referencial.

Perfis com acesso: Chefe de Gabinete, Gestora de Demandas, Funcionários.

Elementos principais: Campos de identificação, seletor de instituição, campo de telefone com máscara.

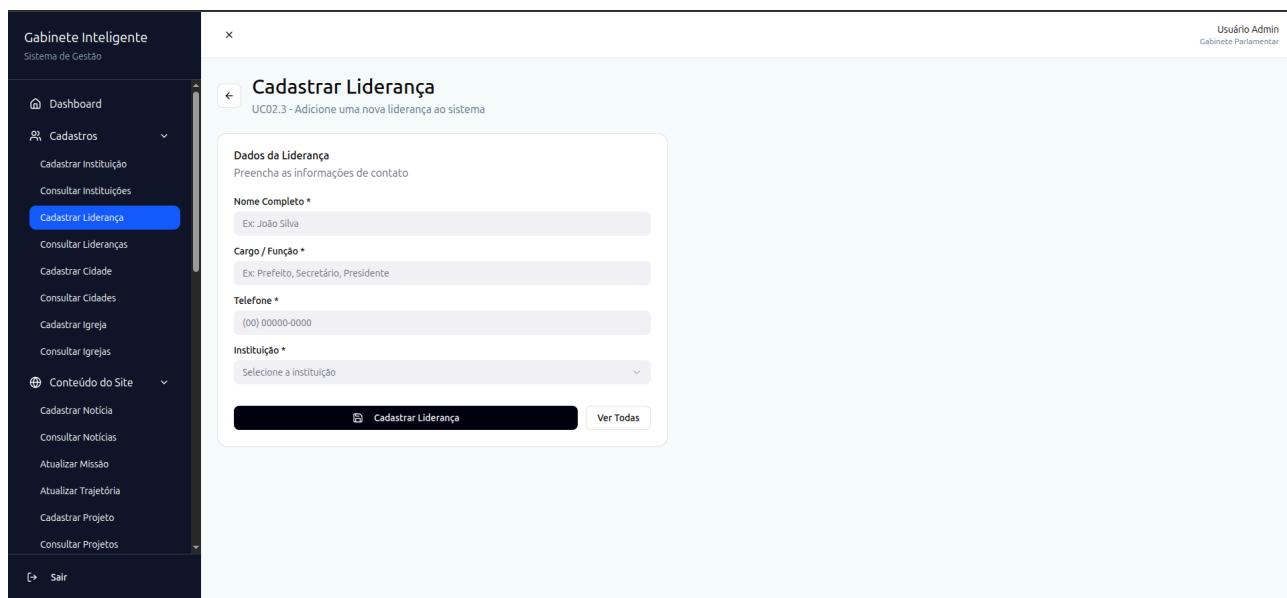


Figura 102 – Tela de Cadastro de Lideranças

Como mostra a Figura 103, a consulta de lideranças oferece filtros por nome e cargo, apresentando resultados tabelados com informações de nome, cargo, instituição, telefone e ações disponíveis.

Perfis com acesso: Todos os perfis internos.

Elementos principais: Filtros de busca múltiplos, tabela com dados de contato, ícones de ação.

Gabinete Inteligente		Consultar Lideranças																															
		UC02.4 - Busque e gerencie lideranças cadastradas																															
		Filtros de Busca <div style="float: right; border: 1px solid #ccc; padding: 2px;">+ Nova Liderança</div>																															
		Nome <input type="text" value="Buscar por nome..."/> Cargo <input type="text" value="Buscar por cargo..."/>	<input type="button" value="Buscar"/>																														
		Resultados (5) <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Nome</th> <th>Cargo</th> <th>Instituição</th> <th>Telefone</th> <th>Ações</th> </tr> </thead> <tbody> <tr> <td>João Silva</td> <td>Prefeito</td> <td>Prefeitura Municipal de Campinas</td> <td>(19) 99876-5432</td> <td><input checked="" type="checkbox"/> <input type="checkbox"/></td> </tr> <tr> <td>Maria Santos</td> <td>Vereadora</td> <td>Câmara Municipal de Santos</td> <td>(13) 98765-4321</td> <td><input checked="" type="checkbox"/> <input type="checkbox"/></td> </tr> <tr> <td>Pedro Oliveira</td> <td>Secretário de Saúde</td> <td>Secretaria Estadual de Saúde</td> <td>(11) 97654-3210</td> <td><input checked="" type="checkbox"/> <input type="checkbox"/></td> </tr> <tr> <td>Ana Costa</td> <td>Presidente</td> <td>Associação de Moradores do Bairro Alto</td> <td>(15) 96543-2109</td> <td><input checked="" type="checkbox"/> <input type="checkbox"/></td> </tr> <tr> <td>Carlos Ferreira</td> <td>Diretor Sindical</td> <td>Sindicato dos Professores</td> <td>(16) 95432-1098</td> <td><input checked="" type="checkbox"/> <input type="checkbox"/></td> </tr> </tbody> </table>		Nome	Cargo	Instituição	Telefone	Ações	João Silva	Prefeito	Prefeitura Municipal de Campinas	(19) 99876-5432	<input checked="" type="checkbox"/> <input type="checkbox"/>	Maria Santos	Vereadora	Câmara Municipal de Santos	(13) 98765-4321	<input checked="" type="checkbox"/> <input type="checkbox"/>	Pedro Oliveira	Secretário de Saúde	Secretaria Estadual de Saúde	(11) 97654-3210	<input checked="" type="checkbox"/> <input type="checkbox"/>	Ana Costa	Presidente	Associação de Moradores do Bairro Alto	(15) 96543-2109	<input checked="" type="checkbox"/> <input type="checkbox"/>	Carlos Ferreira	Diretor Sindical	Sindicato dos Professores	(16) 95432-1098	<input checked="" type="checkbox"/> <input type="checkbox"/>
Nome	Cargo	Instituição	Telefone	Ações																													
João Silva	Prefeito	Prefeitura Municipal de Campinas	(19) 99876-5432	<input checked="" type="checkbox"/> <input type="checkbox"/>																													
Maria Santos	Vereadora	Câmara Municipal de Santos	(13) 98765-4321	<input checked="" type="checkbox"/> <input type="checkbox"/>																													
Pedro Oliveira	Secretário de Saúde	Secretaria Estadual de Saúde	(11) 97654-3210	<input checked="" type="checkbox"/> <input type="checkbox"/>																													
Ana Costa	Presidente	Associação de Moradores do Bairro Alto	(15) 96543-2109	<input checked="" type="checkbox"/> <input type="checkbox"/>																													
Carlos Ferreira	Diretor Sindical	Sindicato dos Professores	(16) 95432-1098	<input checked="" type="checkbox"/> <input type="checkbox"/>																													

Figura 103 – Tela de Consulta de Lideranças

A Figura 104 apresenta o formulário simplificado para cadastro de municípios, contendo campos para nome da cidade, região administrativa e população estimada. Este cadastro é fundamental para vincular demandas, emendas e instituições.

Perfis com acesso: Chefe de Gabinete, Funcionários.

Elementos principais: Campos de texto, seletor de região, campo numérico de população.

The screenshot shows the 'Gabinete Inteligente' interface. On the left, there's a sidebar with various administrative options like 'Dashboard', 'Cadastros' (selected), 'Consultar', 'Conteúdo do Site', and 'Sair'. The main window is titled 'Cadastrar Cidade' and has a sub-instruction 'UC02.5 - Adicione uma nova cidade ao sistema'. It contains three input fields: 'Nome da Cidade' (with placeholder 'Ex: Campinas'), 'Região' (with placeholder 'Selecionar a região'), and 'População (estimada)' (with placeholder 'Ex: 12000000'). At the bottom are two buttons: a large black 'Cadastrar Cidade' button and a smaller 'Ver Todas' button.

Figura 104 – Tela de Cadastro de Cidades

Na Figura 105, a interface de consulta permite buscar cidades por nome, exibindo em tabela as informações de nome, região, população e ações disponíveis.

Perfis com acesso: Todos os perfis internos.

Elementos principais: Campo de busca textual, tabela informativa, indicadores populacionais.

The screenshot shows the 'Consultar Cidades' (Search Cities) page. The left sidebar has a 'Consultar Cidades' button highlighted in blue. The main area has a search bar with placeholder 'Buscar por nome...' and a 'Buscar' button. Below is a table titled 'Resultados (5)' with columns: Nome, Região, População, and Ações. The table data is:

Nome	Região	População	Ações
Campinas	Região Metropolitana	1.213.792 habitantes	<input checked="" type="checkbox"/> <input type="checkbox"/>
Santos	Litoral	433.311 habitantes	<input checked="" type="checkbox"/> <input type="checkbox"/>
São José dos Campos	Vale do Paraíba	729.737 habitantes	<input checked="" type="checkbox"/> <input type="checkbox"/>
Sorocaba	Interior	679.378 habitantes	<input checked="" type="checkbox"/> <input type="checkbox"/>
Ribeirão Preto	Interior	711.825 habitantes	<input checked="" type="checkbox"/> <input type="checkbox"/>

Figura 105 – Tela de Consulta de Cidades

A Figura 106 demonstra o formulário para registro de instituições religiosas, com campos para nome da igreja, tipo/denominação (Católica, Batista, Assembleia de Deus, Presbiteriana, Adventista, etc.) e município.

Perfis com acesso: Chefe de Gabinete, Funcionários.

Elementos principais: Campos de identificação, seletor de denominação, vínculo com município.

The screenshot shows the 'Cadastrar Igreja' (Register Church) page. The left sidebar has a 'Cadastrar Igreja' button highlighted in blue. The main area has a form with three fields: 'Nome da Igreja *' (placeholder: Ex: Igreja Assembleia de Deus Central), 'Tipo / Denominação *' (placeholder: Selecione o tipo), and 'Município *' (placeholder: Selecione o município). At the bottom are two buttons: 'Cadastrar Igreja' and 'Ver Todas'.

Figura 106 – Tela de Cadastro de Igrejas

Como mostra a Figura 107, a consulta de igrejas permite filtrar por nome, exibindo resultados com nome, tipo, município e ações de edição/exclusão.

Perfis com acesso: Todos os perfis internos.

Elementos principais: Campo de busca, tabela de resultados, ações rápidas.

Nome	Tipo	Município	Ações
Igreja Assembleia de Deus Central	Assembleia de Deus	Campinas	
Igreja Batista Memorial	Batista	Santos	
Catedral Católica de São João	Católica	São José dos Campos	
Igreja Presbiteriana do Brasil	Presbiteriana	Sorocaba	
Igreja Adventista do Sétimo Dia	Adventista	Ribeirão Preto	

Figura 107 – Tela de Consulta de Igrejas

A Figura 108 apresenta o editor para publicação de notícias no portal institucional, contendo campos para título, conteúdo (com formatação rica), data de publicação e autor. O sistema registra automaticamente o usuário responsável pela publicação.

Perfis com acesso: Chefe de Gabinete, Funcionários com permissão editorial.

Elementos principais: Editor de texto rico, campos de metadados, botão de publicação.

Figura 108 – Tela de Cadastro de Notícias

Na Figura 109, a interface permite buscar notícias por título, exibindo resultados com título, autor, data e ações de edição/exclusão.

Perfis com acesso: Todos os perfis internos.

Elementos principais: Filtro por título, listagem cronológica, botão de nova notícia.

Título	Autor	Data	Ações
Deputada visita obras da nova escola municipal	Assessoria de Comunicação	14/11/2025	
Emenda de R\$ 500 mil aprovada para reforma do hospital	Gabinete Parlamentar	09/11/2025	
Reunião com lideranças religiosas discute projetos sociais	Assessoria de Comunicação	04/11/2025	
Projeto de Lei para apoio a pequenos produtores é aprovado	Gabinete Parlamentar	27/10/2025	
Audiência pública debate segurança nas escolas	Assessoria de Comunicação	19/10/2025	

Figura 109 – Tela de Consulta de Notícias

A Figura 110 demonstra o editor para atualização da seção institucional "Missão" do site. A interface apresenta um campo de texto rico com preview em tempo real, permitindo visualizar como o conteúdo será exibido publicamente.

Perfis com acesso: Deputada, Chefe de Gabinete.

Elementos principais: Editor de texto, área de preview, botão de salvar alterações.

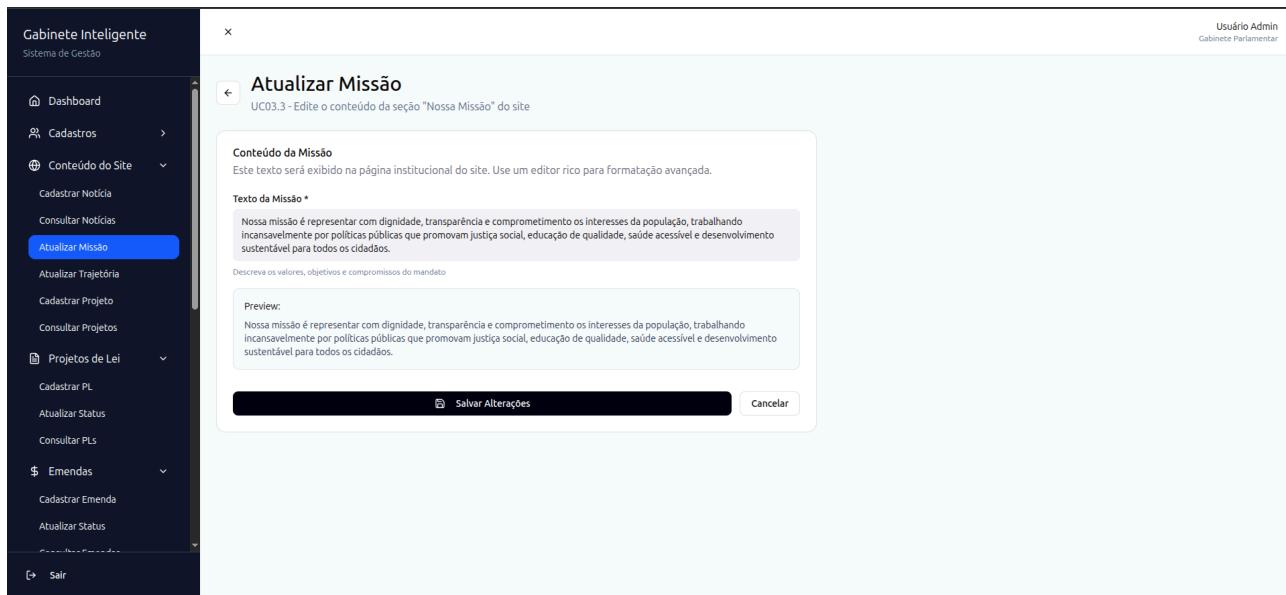


Figura 110 – Tela de Atualização de Missão

Como mostra a Figura 111, similar à tela anterior, esta interface permite editar a seção "Trajetória" do portal, com editor rico e preview.

Perfis com acesso: Deputada, Chefe de Gabinete.

Elementos principais: Editor de texto longo, preview contextual, botão de confirmação.

Figura 111 – Tela de Atualização de Trajetória

A Figura 112 apresenta o formulário para cadastro de projetos institucionais a serem exibidos no portal público, com campos para nome do projeto, descrição, município beneficiado e status.

Perfis com acesso: Chefe de Gabinete, Funcionários.

Elementos principais: Campos descritivos, seletor de município e status, botão de cadastro.

Figura 112 – Tela de Cadastro de Projetos do Site

Na Figura 113, a consulta permite filtrar projetos por nome, exibindo tabela com nome do projeto, município, status e ações disponíveis.

Perfis com acesso: Todos os perfis internos.

Elementos principais: Filtro de busca, tabela de projetos, indicadores de status coloridos.

Nome do Projeto	Município	Status	Ações
Reforma da Praça Central	Campinas	Em Execução	
Construção de Centro Comunitário	Santos	Em Planejamento	
Revitalização da Escola Municipal	São José dos Campos	Concluído	
Ampliação do Posto de Saúde	Sorocaba	Em Execução	
Iluminação LED nas ruas	Ribeirão Preto	Concluído	

Figura 113 – Tela de Consulta de Projetos do Site

A Figura 114 demonstra o formulário de registro de PLs, contendo campos para número do projeto, descrição/ementa, status inicial e data de protocolo.

Perfis com acesso: Deputada, Chefe de Gabinete, Funcionários.

Elementos principais: Campo de número, editor de ementa, seletor de status, campo de data.

Figura 114 – Tela de Cadastro de Projetos de Lei

Como mostra a Figura 115, esta interface permite atualizar o andamento de projetos de lei, selecionando o PL desejado, definindo novo status (Em Comissão, Em Votação, Aprovado, Sancionado, Rejeitado) e adicionando observações sobre a mudança.

Perfis com acesso: Deputada, Chefe de Gabinete, Funcionários.

Elementos principais: Seletor de PL, seletor de status, campo de observações, botão de atualização.

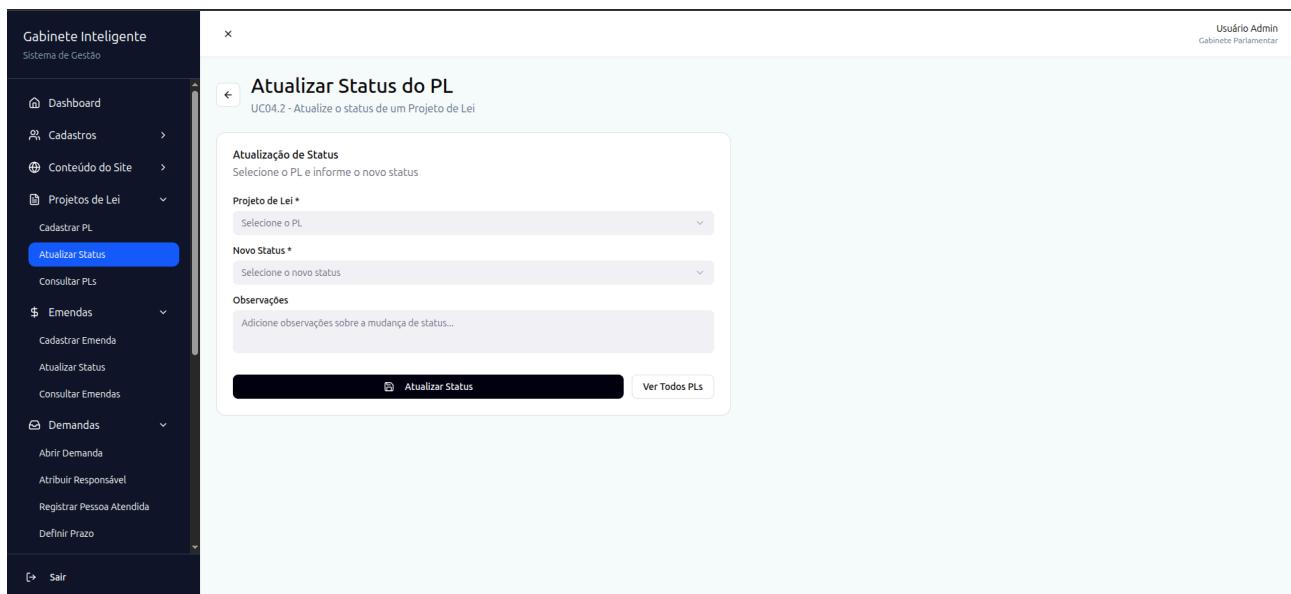


Figura 115 – Tela de Atualização de Status do Projeto de Lei

A Figura 116 apresenta a interface de consulta com filtro por número do PL, exibindo tabela com número, descrição, status e data, além de ações de visualização e edição.

Perfis com acesso: Todos os perfis internos.

Elementos principais: Filtro numérico, tabela de PLs, badges de status, ações por linha.

Figura 116 – Tela de Consulta de Projetos de Lei

Na Figura 117, o formulário permite registrar emendas parlamentares com número, valor em reais, finalidade, município beneficiado e status inicial.

Perfis com acesso: Deputada, Chefe de Gabinete, Funcionários.

Elementos principais: Campos numéricos, campo monetário, área de texto, seletores.

Figura 117 – Tela de Cadastro de Emenda

A Figura 118 demonstra a interface para mudança de status de emendas, com seletor da emenda, novo status (Liberada, Em Análise, Em Execução, Finalizada, Rejeitada) e campo de observações.

Perfis com acesso: Deputada, Chefe de Gabinete, Funcionários.

Elementos principais: Seletor de emenda, seletor de status, campo de observações.

Figura 118 – Tela de Atualização de Status de Emenda

Como mostra a Figura 119, a consulta permite buscar por número da emenda, exibindo tabela com número, finalidade, valor, município, status e ações.

Perfis com acesso: Todos os perfis internos.

Elementos principais: Filtro de busca, tabela com valores monetários, badges coloridos de status.

Número	Finalidade	Valor	Município	Status	Ações
E-45/2025	Reforma do Hospital Municipal	R\$ 500.000,00	Campinas	Liberada	
E-32/2025	Equipamentos para escolas públicas	R\$ 300.000,00	Santos	Em Execução	
E-28/2024	Pavimentação de vias urbanas	R\$ 750.000,00	Sorocaba	Finalizada	
E-15/2024	Construção de centro esportivo	R\$ 1.000.000,00	Ribeirão Preto	Finalizada	

Figura 119 – Tela de Consulta de Emendas

A Figura 120 apresenta o formulário completo para registro de novas demandas, com campos para título, descrição detalhada, município, instituição solicitante e prioridade (Alta, Média, Baixa).

Perfis com acesso: Chefe de Gabinete, Gestora de Demandas, Equipe de Atendimento.

Elementos principais: Campos de texto, seletores múltiplos, indicador de prioridade, botão de abertura.

Figura 120 – Tela de Abertura de Demanda

Na Figura 121, a interface permite selecionar uma demanda e atribuir um usuário responsável, com notificação automática por e-mail ao designado.

Perfis com acesso: Chefe de Gabinete, Gestora de Demandas.

Elementos principais: Seletor de demanda, seletor de usuário responsável, botão de atribuição, aviso de notificação.

Figura 121 – Tela de Atribuição de Responsável

A Figura 122 demonstra o formulário para vincular dados do cidadão atendido à demanda, com campos para demanda relacionada, nome completo, CPF e telefone.

Perfis com acesso: Gestora de Demandas, Equipe de Atendimento.

Elementos principais: Seletor de demanda, campos de identificação pessoal, botão de registro.

Figura 122 – Tela de Registro de Atendimento

Como mostra a Figura 123, esta tela permite estabelecer data limite para conclusão da demanda, com seletor de demanda, campo de data e aviso sobre lembretes automáticos próximos ao vencimento.

Perfis com acesso: Chefe de Gabinete, Gestora de Demandas.

Elementos principais: Seletor de demanda, campo de data, informativo de notificações, botão de definição.

The screenshot shows the 'Gabinete Inteligente' interface. On the left, there's a sidebar with various menu items like 'Dashboard', 'Cadastros', 'Conteúdo do Site', 'Projetos de Lei', 'Emendas', 'Demandas', 'Agenda', and 'Saír'. The 'Demandas' item has a dropdown with 'Definir Prazo' selected. The main content area is titled 'Definir Prazo' with the subtitle 'UC06.4 - Estabeleça um prazo limite para a demanda'. It contains a 'Definição de Prazo' section with a note 'Estabeleça a data limite para conclusão da demanda', a 'Demandas' dropdown, a 'Data Limite' input field ('dd/mm/aaaa'), and a 'Lembretes automáticos' section with a note 'O sistema enviará notificações próximas ao vencimento do prazo'. At the bottom are 'Definir Prazo' and 'Ver Demandas' buttons. The top right corner shows 'Usuário Admin' and 'Gabinete Parlamentar'.

Figura 123 – Tela de Definição de Prazo

A Figura 124 apresenta o formulário para anexar documentos oficiais às demandas, com campos para identificação da demanda, tipo de documento, número do ofício, data de emissão, órgão de destino, assunto, observações e upload do arquivo digital.

Perfis com acesso: Gestora de Demandas, Equipe de Atendimento.

Elementos principais: Seletor de demanda, campos descritivos do documento, área de upload de arquivo.

The screenshot shows the 'Anexar Ofício' (Attach Document) screen. The left sidebar has 'Anexar Ofício' selected. The main form has the following fields:

- Identificação da Demanda**
 - Demandá *
 - Tipo de Documento *
 - Data de Emissão *
 - Número do Ofício *
 - Órgão de Destino *
- Conteúdo do Documento**
 - Assunto *
 - Observações
- Upload do Arquivo**

Figura 124 – Tela de Anexo de Ofício

Na Figura 125, a interface permite alterar o status de andamento das demandas, com seletor da demanda, novo status (Em Andamento, Aguardando Resposta, Concluída, Arquivada) e campo de observação.

Perfis com acesso: Gestora de Demandas, Equipe de Atendimento.

Elementos principais: Seletor de demanda, seletor de status, campo de observação, botão de atualização.

The screenshot shows the 'Consultar Demandas' (Search Demands) screen. The left sidebar has 'Consultar Demandas' selected. The main form has the following sections:

- Filtros de Busca** (Search Filters)

Busca Geral	Status	Prioridade
Buscar por título, código ou origem...	Todos	Todas
- Resultados (5)** (Results (5))

Código	Título	Origem	Status	Prioridade	Responsável	Abertura	Prazo	Ações
DEM-001	Solicitação de iluminação pública	Associação de Moradores Vila São José	Em Andamento	Alta	João Silva	15/10/2024	15/12/2024	
DEM-002	Recabeamento de rua	Morador - Rua das Flores	Aguardando Resposta	Média	Maria Santos	20/10/2024	20/01/2025	
DEM-003	Ampliação de posto de saúde	Conselho de Saúde Municipal	Concluída	Alta	Carlos Oliveira	05/09/2024	05/11/2024	
DEM-004	Construção de creche municipal	Secretaria de Educação	Em Andamento	Alta	Ana Costa	01/10/2024	01/03/2025	
DEM-005	Melhorias no transporte público	Sindicato dos Trabalhadores	Arquivada	Baixa	Pedro Martins	10/08/2024	10/10/2024	

Figura 125 – Tela de Consulta de Demandas

A Figura 126 demonstra a interface de consulta com filtros múltiplos por busca geral, status e prioridade, exibindo tabela completa com código, título, origem, status, prioridade, responsável, data de abertura, prazo e ações (visualizar/anexar).

Perfis com acesso: Todos os perfis internos.

Elementos principais: Filtros combinados, tabela detalhada, badges de status e prioridade, ícones de ação.

The screenshot shows the 'Cadastrar Evento' (Create Event) page. The left sidebar has a dark theme with white text and icons. It includes sections for Demanda (Demandas, Abrir Demanda, Atribuir Responsável, Registrar Pessoa Atendida, Definir Prazo, Atualizar Status, Anexar Ofício, Consultar Demandas), Agenda (Agenda, Cadastrar Evento, Visualizar Calendário, Criar Alerta), and Relatórios (Projetos de Lei, Emendas, Demandas, Consolidado Geral). The 'Cadastrar Evento' option is highlighted with a blue box. The main content area has a light gray background. It starts with a header 'Cadastrar Evento' and a sub-header 'UC07.1 - Adicione um novo evento à sua agenda'. Below this is a section titled 'Informações do Evento' with fields for 'Título do Evento' (Title), 'Tipo de Evento' (Event Type), 'Data de Início' (Start Date), 'Data de Término' (End Date), 'Local' (Location), 'Hora de Início' (Start Time), and 'Hora de Término' (End Time). There is also a 'Descrição' (Description) field with placeholder text 'Descreva os detalhes e objetivos do evento...' and a 'Participantes' (Participants) field with placeholder text 'Liste os participantes esperados...'. At the bottom is a 'Cor no Calendário' (Color in Calendar) section with a color picker and the instruction 'Escolha uma cor para identificar o evento no calendário'.

Figura 126 – Tela de Cadastro de Evento

A Figura 127 apresenta o formulário para registro de compromissos na agenda, com campos para título do evento, tipo (Reunião, Audiência, Visita, Sessão), local, data de início, hora de início, data de término, hora de término, descrição, participantes esperados e seletor de cor no calendário.

Perfis com acesso: Deputada, Chefe de Gabinete.

Elementos principais: Campos de identificação do evento, campos de data/hora duplos, editor de descrição, lista de participantes, seletor de cor.

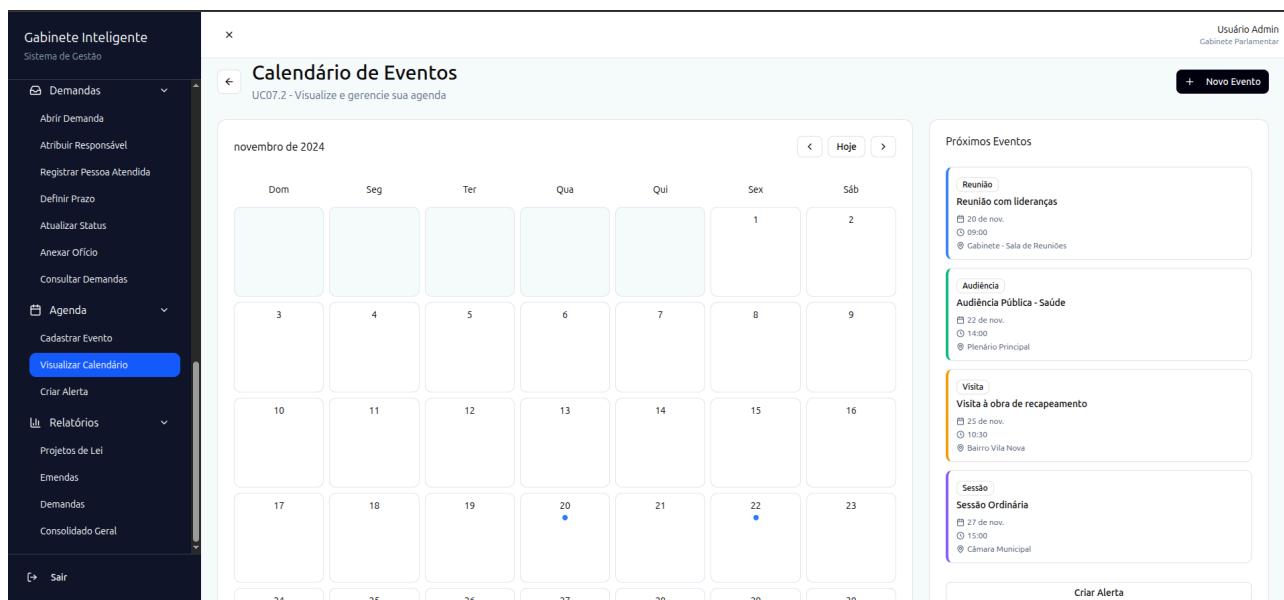


Figura 127 – Tela de Visualização do Calendário

Como mostra a Figura 128, o calendário mensal exibe todos os eventos agendados com indicadores visuais por dia. À direita, um painel lista os próximos eventos com detalhes de tipo, título, data, horário e local. A navegação permite alternar entre meses e um botão "Novo Evento" facilita cadastros rápidos.

Perfis com acesso: Todos os perfis internos.

Elementos principais: Calendário mensal interativo, painel de próximos eventos, navegação temporal, botão de novo evento.

Criar Alerta

UC07.3 - Configure lembretes para seus eventos

Informações do Alerta

Título do Alerta *
Ex: Lembrete - Reunião com lideranças

Evento Relacionado
Selecione um evento (opcional)

Data do Alerta *
dd/mm/aaaa

Hora do Alerta *

Antecedência
Selecione a antecedência

Tipo de Notificação *
Selecione o tipo

Mensagem Personalizada
Digite a mensagem que deseja receber no alerta...

Repetir alerta periodicamente

Dica sobre alertas
Configure alertas com antecedência adequada para garantir que você não perca compromissos importantes. Você pode criar múltiplos alertas para o mesmo evento.

Cancelar **Criar Alerta**

Figura 128 – Tela de Criação de Alerta

A Figura 129 demonstra o formulário para configurar lembretes automáticos, com campos para título do alerta, evento relacionado (opcional), data do alerta, hora do alerta, antecedência (1 hora antes, 1 dia antes, 1 semana antes, etc.), tipo de notificação (E-mail, Notificação no Sistema) e mensagem personalizada. Um checkbox permite configurar alertas periódicos.

Perfis com acesso: Deputada, Chefe de Gabinete.

Elementos principais: Campos de configuração temporal, seletor de tipo de notificação, editor de mensagem, opção de recorrência.

Relatório de Projetos de Lei

UC08.1 - Análise completa dos projetos de lei

Total de Projetos	Em Tramitação	Aprovados	Rejeitados
5	2	2	1

Filtros do Relatório

Período: Ano Atual; Status: Todos; Formato de Exportação: PDF

Detalhamento dos Projetos

Número	Título	Autor	Status	Data
PL 001/2024	Ampliação da rede de saúde	João Silva	Em Tramitação	15/01/2024
PL 002/2024	Programa de educação digital	Maria Santos	Aprovado	20/02/2024
PL 003/2024	Incentivo ao transporte sustentável	Carlos Oliveira	Em Tramitação	10/03/2024
PL 004/2024	Reforma da praça central	Ana Costa	Rejeitado	05/04/2024
PL 005/2024	Ampliação de bibliotecas públicas	Pedro Martins	Sancionado	22/05/2024

Figura 129 – Tela de Relatório de Projetos de Lei

Na Figura 130, o relatório de emendas apresenta total de emendas (5), valor total (R\$ 2.500.000,00), distribuição por status (2 Aprovadas, 1 Executada) e filtros por período, tipo de emenda e status. A tabela detalha número, tipo, valor, município e status de cada emenda.

Perfis com acesso: Deputada, Chefe de Gabinete.

Elementos principais: Indicadores financeiros, filtros combinados, tabela com valores monetários, opções de exportação.

Número	Tipo	Valor	Município	Status	Data
EMD-001/2024	Orçamentária	R\$ 500.000	Campinas	Aprovada	10/03/2024
EMD-002/2024	Parlamentar	R\$ 250.000	Santos	Em Análise	15/03/2024
EMD-003/2024	Orçamentária	R\$ 800.000	São Paulo	Aprovada	20/04/2024
EMD-004/2024	Impositiva	R\$ 350.000	Sorocaba	Rejeitada	05/05/2024
EMD-005/2024	Parlamentar	R\$ 600.000	Ribeirão Preto	Executada	12/06/2024

Figura 130 – Tela de Relatório de Emendas

A Figura 131 demonstra o relatório operacional com total de demandas (5), distribuição por status (2 Em Andamento, 1 Concluída, 1 Aguardando) e filtros por período, prioridade, status e formato. A tabela lista código, título, origem, prioridade, status e data de cada demanda.

Perfis com acesso: Deputada, Gestora de Demandas, Equipe de Atendimento.

Elementos principais: Cards quantitativos, filtros múltiplos, tabela operacional, botões de geração.

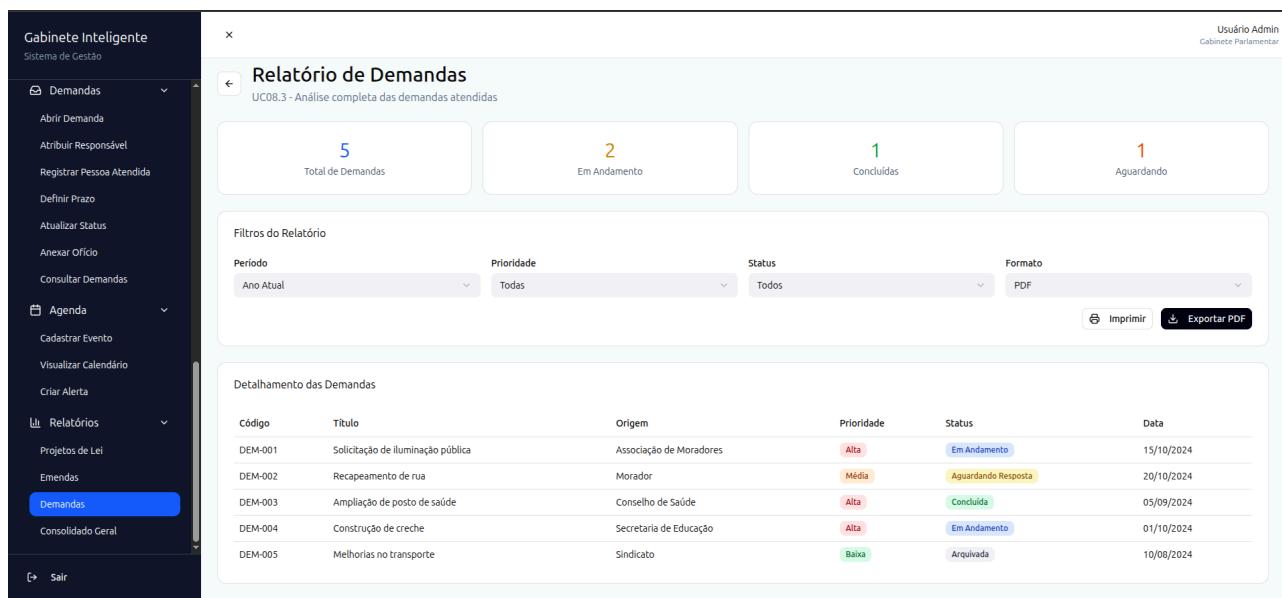


Figura 131 – Tela de Relatório de Demandas

Como mostra a Figura 132, o relatório estratégico exclusivo da Deputada consolida todas as atividades do gabinete em uma única visualização. A seção "Configurações do Relatório" permite selecionar período e formato de exportação (PDF Completo). O relatório é dividido em quatro blocos principais: Projetos de Lei (12 totais, com distribuição: 5 Aprovados, 6 Em Tramitação, 1 Rejeitado, taxa de aprovação 42%), Emendas Parlamentares (18 totais, R\$ 4.500.000,00 destinados, 10 Aprovadas, 5 Executadas), Demandas Atendidas (35 totais, com distribuição não detalhada na imagem) e Agenda e Eventos (42 totais).

Perfis com acesso: Deputada (exclusivo).

Elementos principais: Configurações de período, múltiplos painéis de indicadores, dados legislativos e financeiros consolidados, botões de impressão e exportação.

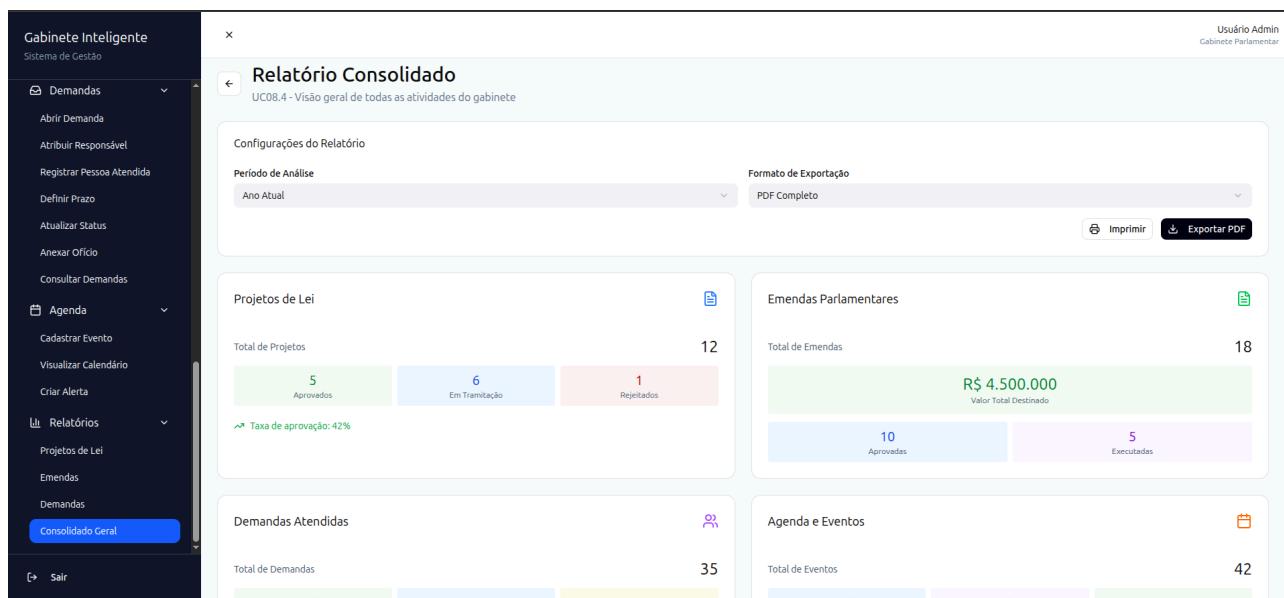


Figura 132 – Tela de Relatório Consolidado

5. Glossário e Modelos de Dados

O modelo de dados foi desenvolvido com base nas principais entidades identificadas nos casos de uso e histórias de usuário. Ele reflete o relacionamento entre demandas, emendas, cidades, instituições e outros objetos do domínio do sistema. A Figura 133 representa o Diagrama de Entidade e Relacionamento para o sistema a ser desenvolvido.

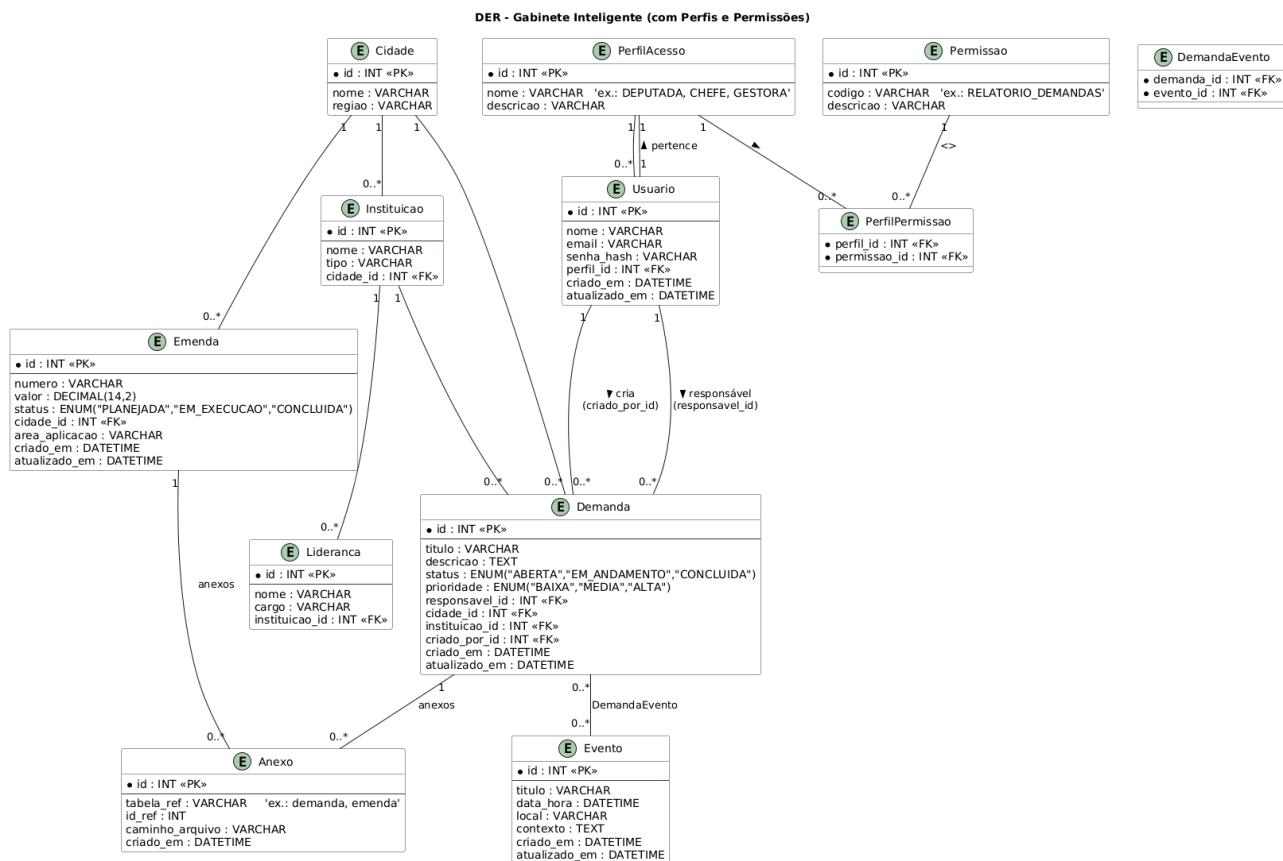


Figura 133 – Diagrama de Entidade e Relacionamento

Os glossários a seguir, representados pelas Tabelas de 29 a 40, apresentam os principais atributos reconhecidos pelos usuários nas interfaces do sistema. Eles foram separados por entidade, por esse motivo, são exibidas uma tabela para cada uma. Cada atributo possui um formato e uma descrição resumida, facilitando o entendimento e a padronização dos termos utilizados no sistema.

Cidade		
Atributo	Tipo	Descrição
id	INT (PK)	Identificador único da cidade.
nome	VARCHAR	Nome do município.
regiao	VARCHAR	Região administrativa.

Tabela 29 - Glossário para Cidade

PerfilAcesso		
Atributo	Tipo	Descrição
id	INT (PK)	Identificador do perfil.

PerfilAcesso		
nome	VARCHAR	Nome do perfil.
descricao	VARCHAR	Descrição do perfil.

Tabela 30 - Glossário para PerfilAcesso

Permissao		
Atributo	Tipo	Descrição
id	INT (PK)	Identificador da permissão.
codigo	VARCHAR	Código da permissão.
descricao	VARCHAR	Descrição textual.

Tabela 31 - Glossário para Permissao

PerfilPermissao		
Atributo	Tipo	Descrição
perfil_id	INT (FK)	Perfil associado.
permissao_id	INT (FK)	Permissão atribuída.

Tabela 32 - Glossário para PerfilPermissao

Usuario		
Atributo	Tipo	Descrição
id	INT (PK)	Identificador do usuário.
nome	VARCHAR	Nome completo.
email	VARCHAR	E-mail único.
senha_hash	VARCHAR	Senha criptografada.
perfil_id	INT (FK)	Perfil associado.
criado_em	DATETIME	Data de criação.
atualizado_em	DATETIME	Última atualização.

Tabela 33 - Glossário para Usuario

Instituicao		
Atributo	Tipo	Descrição
id	INT (PK)	Identificador da instituição.
nome	VARCHAR	Nome da instituição.
tipo	VARCHAR	Tipo da instituição.
cidade_id	INT (FK)	Cidade associada.

Tabela 34 - Glossário para Instituicao

Lideranca		
Atributo	Tipo	Descrição
id	INT (PK)	Identificador.
nome	VARCHAR	Nome da liderança.
cargo	VARCHAR	Cargo exercido.
instituicao_id	INT (FK)	Instituição vinculada.

Tabela 35 - Glossário para Lideranca

Emenda		
Atributo	Tipo	Descrição
id	INT (PK)	Identificador.
numero	VARCHAR	Número oficial.
valor	DECIMAL(14,2)	Valor financeiro.
status	ENUM	Status da emenda.
cidade_id	INT (FK)	Cidade beneficiada.
area_aplicacao	VARCHAR	Área temática.
criado_em	DATETIME	Data de criação.
atualizado_em	DATETIME	Última atualização.

Tabela 36 - Glossário para Emenda

Demanda		
Atributo	Tipo	Descrição
id	INT (PK)	Identificador.
titulo	VARCHAR	Título da demanda.
descricao	TEXT	Descrição detalhada.
status	ENUM	Status atual.
prioridade	ENUM	Prioridade.
responsavel_id	INT (FK)	Responsável.
cidade_id	INT (FK)	Cidade associada.
instituicao_id	INT (FK)	Instituição associada.
criado_por_id	INT (FK)	Usuário criador.
criado_em	DATETIME	Data de criação.
atualizado_em	DATETIME	Última atualização.

Tabela 37 - Glossário para Demanda

Anexo		
Atributo	Tipo	Descrição
id	INT (PK)	Identificador.
tabela_ref	VARCHAR	Tabela de referência.
id_ref	INT	ID do registro vinculado.
caminho_arquivo	VARCHAR	Local do arquivo.
criado_em	DATETIME	Data de upload.

Tabela 38 - Glossário para Anexo

Evento		
Atributo	Tipo	Descrição
id	INT (PK)	Identificador.
titulo	VARCHAR	Título do evento.
data_hora	DATETIME	Data e hora.
local	VARCHAR	Local de realização.
contexto	TEXT	Descrição contextual.
criado_em	DATETIME	Data de criação.

Evento		
atualizado_em	DATETIME	Última atualização.

Tabela 39 - Glossário para Evento

DemandaEvento		
Atributo	Tipo	Descrição
demand_id	INT (FK)	Demanda associada.
evento_id	INT (FK)	Evento associado.

Tabela 40 - Glossário para DemandaEvento

6. Casos de Teste

7. Cronograma e Processo de Implementação

O cronograma do TCC II foi estruturado em quinzenas, cobrindo o período entre a primeira quinzena de fevereiro de 2026 e a segunda quinzena de junho de 2026. Esse planejamento distribui de forma progressiva e organizada todas as etapas necessárias para a implementação, testes, documentação e preparação da defesa do Sistema Gabinete Virtual.

O cronograma, apresentado na Tabela 41, inicia com atividades de alinhamento, revisão dos artefatos produzidos durante o TCC I e preparação do ambiente técnico de desenvolvimento. Em seguida, avançam-se para as fases de implementação dos principais módulos do sistema, começando pelos componentes centrais — autenticação, perfis de acesso, cadastros e gestão de demandas — e depois evoluindo para os módulos legislativos, gestão de agenda, CMS e geração de relatórios.

A partir do mês de maio, o foco se desloca para testes integrados, refinamento das funcionalidades, otimização das rotinas e ajustes de usabilidade, garantindo que o sistema esteja estável, coerente com os requisitos e alinhado às especificações técnicas. Por fim, no mês de junho, concentram-se as atividades de finalização do frontend, revisão detalhada da documentação, construção dos slides e preparação para a apresentação perante a banca avaliadora. A última quinzena abriga a entrega final e a defesa do trabalho.

Esse planejamento escalonado permite que o desenvolvimento avance de forma contínua, reduzindo riscos, garantindo tempo para revisões e possibilitando que cada módulo seja implementado e validado com qualidade antes do lançamento final.

Período	Atividades Previstas
1ª Quinzena – Fevereiro/2026	Início oficial do TCC II; Revisão final dos artefatos; Alinhamento com orientador; Ajustes na documentação; Definição do escopo da implementação.
2ª Quinzena – Fevereiro/2026	Preparação do ambiente; Estruturação inicial do backend; Definição das entidades; Configuração do banco de dados.
1ª Quinzena – Março/2026	Implementação da autenticação e perfis; Cadastros básicos; Criação das primeiras rotas; Testes iniciais.
2ª Quinzena – Março/2026	Implementação da gestão de demandas; Histórico; Integração inicial com frontend; Testes integrados.
1ª Quinzena – Abril/2026	Implementação da gestão de emendas e PLs; Validação de status; Criação de serviços auxiliares.
2ª Quinzena – Abril/2026	Implementação da agenda; CMS; Integração com frontend; Testes dos módulos.
1ª Quinzena – Maio/2026	Desenvolvimento dos relatórios; Criação de componentes visuais; Ajustes de usabilidade.
2ª Quinzena – Maio/2026	Testes completos; Correções; otimização de consultas; Revisão dos módulos.
1ª Quinzena – Junho/2026	Conclusão do frontend; Revisão da documentação; Preparação da apresentação.
2ª Quinzena – Junho/2026	Entrega final; Defesa; Consolidação dos artefatos no GitHub Classroom.

Tabela 41 - Cronograma do Sistema