

Dossiê de Análise e Projeto de Sistemas

Projeto: [Nome do Aplicativo]

Versão: 1.0

Data: [Data de Entrega]

1. Identificação da Equipe

Nome Completo	Prontuário/Matrícula	Papel Principal na Equipe
Isabella Vitoria Astolfi	BV3055469	
Paola Cassiano Pires	BV3055418	
Talita Braz dos Santos	BV3044793	

2. Concepção do Projeto

2.1. O Problema

Estudantes frequentemente **esquecem datas importantes** de trabalhos escolares, provas e outros compromissos acadêmicos. Os principais problemas identificados pela equipe foram:

- Perda de prazos de entrega por falta de lembretes
- Dificuldade em priorizar múltiplas tarefas com prazos próximos
- Confusão entre compromissos pessoais e escolares
- Impossibilidade de acompanhar o progresso das tarefas
- Esquecimento frequente de revisar o calendário

A solução precisa ser rápida, simples e direta, focando especificamente em não deixar nenhuma data importante passar despercebida.

2.2. A Solução Proposta

Um aplicativo minimalista de **gerenciamento de prazos** que permite aos usuários:

1. **Cadastrar rapidamente** prazos de trabalhos e provas com título, tipo, data e hora
2. **Receber notificações personalizadas** 24 horas antes e no dia do evento
3. **Visualizar de forma clara** os próximos prazos em ordem cronológica
4. **Editar e excluir prazos** de forma intuitiva e rápida
5. **Gerenciar compromissos** sem funcionalidades complexas ou desnecessárias

O diferencial está na simplicidade e foco: interface minimalista, registro rápido, notificações oportunas e visualização clara.

2.3. Escopo do Projeto

- **O que o sistema FARÁ:**

1. Permitir ao usuário cadastrar prazos de trabalhos e provas de forma rápida.
2. Enviar notificações automáticas e personalizadas 24 horas antes e no dia do evento.
3. Exibir uma visualização clara e organizada dos próximos prazos do usuário.
4. Permitir ao usuário editar e excluir prazos cadastrados conforme necessário.
5. Oferecer uma interface minimalista e fácil de usar para facilitar o registro e consulta dos compromissos.

- **O que o sistema NÃO FARÁ:**

1. Não enviará notificações por outros canais além do próprio app (exemplo: SMS ou WhatsApp).
2. Não integrará outros calendários digitais (Google Calendar, Outlook, etc.) nesta versão.
3. Não fará análise automática de prioridades ou sugestão de horários de estudo.
4. Não incluirá controle de tarefas, checklist ou acompanhamento detalhado do progresso das atividades.
5. Não terá funcionalidades de compartilhamento de prazos com outros usuários nesta versão do sistema.

3. Engenharia de Requisitos

3.1. Atores do Sistema

Autor	Descrição
Usuário	Usuário que utiliza o app para cadastrar, editar, visualizar e excluir prazos de trabalhos e provas. Recebe notificações personalizadas sobre os compromissos.
Sistema	Sistema responsável pelo monitoramento dos prazos, envio das notificações 24h antes e no dia do evento, e gerenciamento dos dados dos usuários.

Serviço de Notificação	Componente do sistema que agenda e envia as notificações para o usuário nos horários programados.
-------------------------------	---

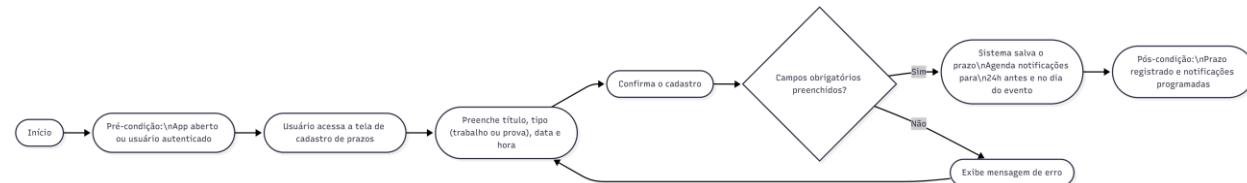
Banco de Dados	Armazena as informações dos usuários, prazos cadastrados e notificações agendadas.
-----------------------	--

3.2. Histórias de Usuário (Backlog)

1. **Como usuário**, quero cadastrar meus prazos de trabalhos e provas de forma rápida para garantir que não vou esquecer nenhuma data importante.
2. **Como usuário**, quero receber notificações simples e personalizadas 24 horas antes e no dia do evento para lembrar dos prazos.
3. **Como usuário**, quero visualizar de forma clara e minimalista os próximos prazos para organizar meu tempo de forma eficiente.
4. **Como usuário**, quero poder editar e excluir prazos rapidamente para manter meu planejamento sempre atualizado.

4. Modelagem Funcional (Casos de Uso)

4.1. Diagrama de Casos de Uso



Legenda: Diagrama representando os atores e as principais funcionalidades do sistema.

4.2. Descrições de Casos de Uso

UC-01: Cadastrar Prazo

Ator Principal: Usuário

Pré-condições: Usuário autenticado no sistema com app aberto.

Fluxo Principal:

1. Usuário acessa a tela de cadastro de prazos.
2. Usuário preenche título, tipo (trabalho ou prova), data e hora.
3. Usuário confirma o cadastro.
4. Sistema valida se os campos obrigatórios foram preenchidos.
5. Sistema salva o novo prazo e agenda notificações (24h antes e no dia do evento).

6. Sistema exibe confirmação de cadastro ao usuário.

Fluxos Alternativos/Exceção:

A1. Se campos obrigatórios não forem preenchidos, sistema exibe mensagem de erro e solicita ajuste dos dados.

UC-02: Receber Notificação de Prazo

Ator Principal: Usuário

Pré-condições: Prazo cadastrado previamente, notificações ativas.

Fluxo Principal:

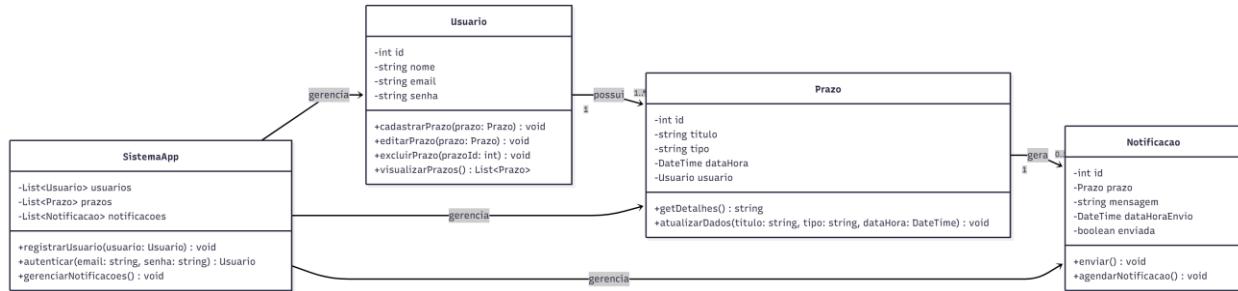
1. Sistema monitora os prazos cadastrados.
2. Quando restam 24 horas para o evento, sistema envia notificação personalizada ao usuário.
3. No dia do evento, sistema envia nova notificação.
4. Usuário visualiza a notificação e se prepara para o compromisso.

Fluxos Alternativos/Exceção:

A1. Caso o usuário desative notificações, nenhuma notificação será enviada.

5. Modelagem Estrutural (Classes)

5.1. Diagrama de Classes



Nota: O diagrama deve conter classes, atributos (privados), métodos (públicos), relacionamentos e multiplicidades.

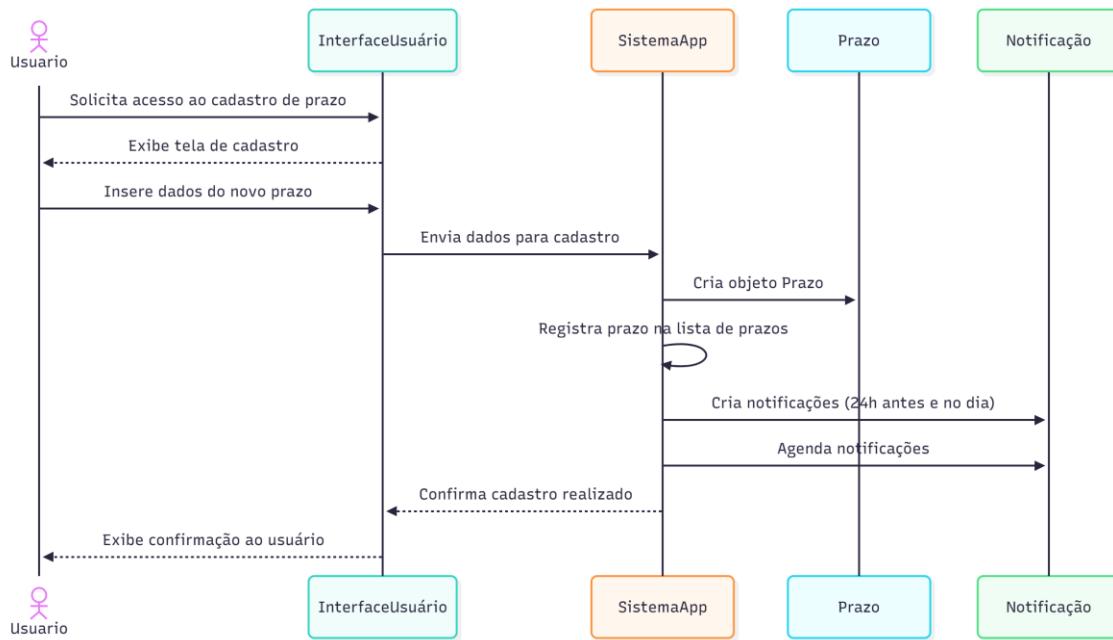
5.2. Dicionário de Classes (Opcional)

- **SistemaApp:** Responsável por gerenciar usuários, prazos e notificações. Executa funções principais do sistema, como autenticação e controle das notificações.
- **Usuario:** Representa o usuário do sistema. Permite cadastrar, editar, excluir e visualizar prazos associados ao seu perfil.

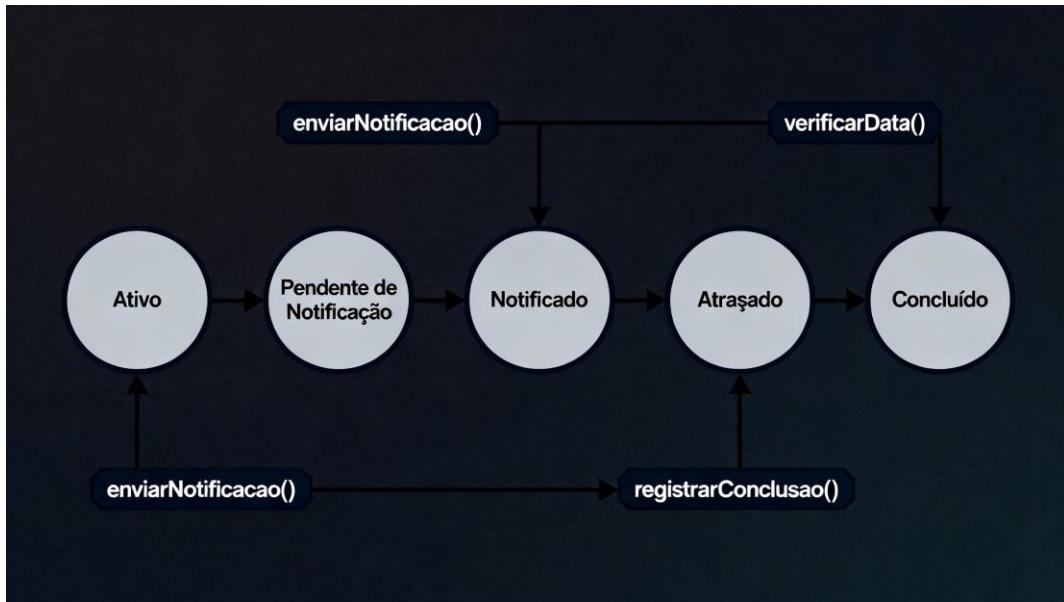
- **Prazo:** Refere-se a um prazo de trabalho ou prova cadastrado pelo usuário. Possui informações como título, tipo, data e hora, além de métodos para retornar detalhes e atualizar dados.
- **Notificação:** Responsável por armazenar informações das notificações geradas para os prazos. Atua agendando e enviando alertas ao usuário nas datas programadas.

6. Modelagem Comportamental (Sequência e Estados)

6.1. Diagrama de Sequência

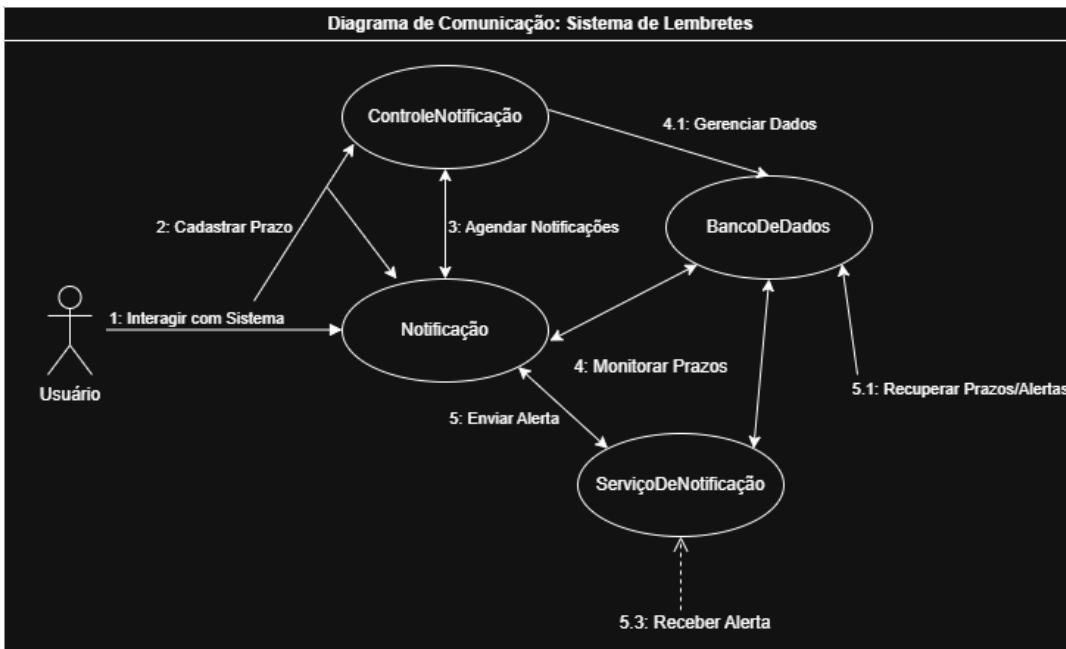


6.2. Diagrama de Estados



7. Arquitetura e Implantação (Opcional)

7.1. Diagrama de Componentes/Implantação



8. Conclusão

Durante a criação do projeto, tivemos algumas dificuldades para entender quais requisitos realmente eram necessários e como transformar as ideias em diagramas claros. Aprendemos bastante sobre como cada etapa da modelagem ajuda a visualizar melhor o funcionamento do sistema antes mesmo de começar a programar. O trabalho em equipe também foi essencial, pois cada um contribuiu com pontos de vista diferentes, o que ajudou bastante a encontrar soluções e ajustar o que fosse preciso para o sistema ficar mais simples e prático para o usuário.