INSTITUTO POLITÉCNICO DE BEJA

Escola Superior de Tecnologia e Gestão

Licenciatura em Engenharia Informática

EPItogether

André Reis nº21269

2024

INSTITUTO POLITÉCNICO DE BEJA

Escola Superior de Tecnologia e Gestão

Licenciatura em Engenharia Informática

EPItogether

Elaborado por:

André Reis nº21269

Orientado por:

Isabel Sofia Sousa Brito

Relatório de projeto da cadeira de Engenharia de Software na

Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Beja

2024

Resumo

O presente relatório fornece uma visão geral do projeto de desenvolvimento de software em andamento, detalhando o modelo de execução, os prazos previstos e as ferramentas a serem utilizadas. Com uma abordagem estruturada e colaborativa, buscamos alcançar nossos objetivos dentro do cronograma estabelecido, contando com a contribuição dos membros da equipe em suas áreas de especialização. Através do uso de ferramentas e plataformas CASE adequadas, visamos superar desafios e entregar um produto final de alta qualidade. Este relatório representa um marco importante em nosso progresso, demonstrando nosso compromisso com a excelência e a satisfação dos stakeholders.

Palavras-chave: desenvolvimento de software, projeto, modelo de execução, prazos, ferramentas CASE, colaboração, equipe, especialização, desafios, qualidade, stakeholders.

Abstract

This report provides an overview of the ongoing software development project, detailing the execution model, estimated deadlines, and tools to be used. With a structured and collaborative approach, we aim to achieve our objectives within the established timeline, leveraging the contributions of team members in their areas of expertise. Through the use of appropriate CASE tools and platforms, we seek to overcome challenges and deliver a high-quality end product. This report represents a significant milestone in our progress, demonstrating our commitment to excellence and stakeholder satisfaction.

Keywords: software development, project, execution model, deadlines, CASE tools, collaboration, team, expertise, challenges, quality, stakeholders.

Índice

Lista de Figuras

1. Introdução

Este relatório tem como objetivo apresentar a conceção e desenvolvimento de um sistema de gestão de crises e saúde destinado a adolescentes. O projeto foi realizado no âmbito da disciplina de Engenharia de Software do ano letivo 2023-2024.

A necessidade de um sistema deste tipo surge da crescente preocupação com a saúde mental e física dos jovens, especialmente na fase da adolescência, onde crises de saúde podem ter um impacto significativo na vida dos indivíduos. O sistema visa fornecer uma plataforma integrada onde adolescentes, pais, profissionais de saúde e educadores podem colaborar de forma eficiente para monitorizar, gerir e responder a crises de saúde.

Ao longo deste relatório, serão detalhadas as várias etapas do projeto, desde a análise de requisitos até à implementação e testes do sistema. O foco estará nos casos de uso mais relevantes, como o registo de sintomas e crises, a gestão de notas e o agendamento de consultas. Serão também apresentadas as principais decisões de design, incluindo os diagramas de casos de uso, sequências e classes que foram utilizados para modelar o sistema.

Esperamos que este relatório forneça uma visão clara e detalhada do processo de desenvolvimento do sistema, demonstrando a aplicação prática dos conceitos e técnicas aprendidos ao longo do curso. Acreditamos que o sistema desenvolvido tem o potencial de ser uma ferramenta valiosa para a gestão de crises de saúde entre adolescentes, contribuindo para um melhor acompanhamento e apoio a esta faixa etária.

2. Modelo de Desenvolvimento de Software:

Cascata.

O modelo em cascata é um dos modelos de ciclo de vida de desenvolvimento de software mais antigos e amplamente utilizados. Ele segue uma abordagem sequencial, onde o desenvolvimento do software é dividido em fases distintas (Análise, Desenho, Codificação, Teste, Implantação Manutenção), cada uma das quais deve ser concluída antes de avançar para a próxima.

O modelo em cascata é caracterizado pela sua abordagem sequencial e linear, onde cada fase depende do resultado da fase anterior. Isso significa que qualquer mudança nos requisitos ou no design do sistema pode ser difícil de incorporar após o início da implementação, tornando o modelo menos flexível em comparação com abordagens ágeis mais iterativas. No entanto, o modelo em cascata é eficaz para projetos onde os requisitos são bem compreendidos e estáveis desde o início.

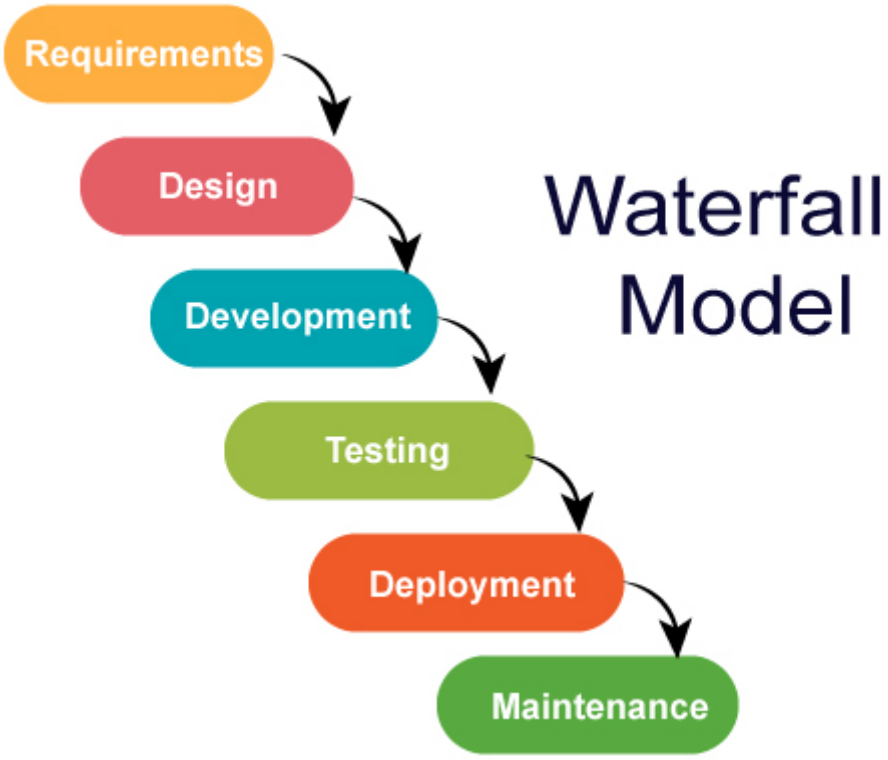

Figura 1- Modelo Cascata

Descrição do Modelo:

1. **Análise (2 semanas):**
    - Recolha de informação: Realizar entrevistas com potenciais utilizadores, pesquisa online de sistemas semelhantes.
    - Identificação de requisitos: Diferenciar entre requisitos funcionais e não funcionais, especificar requisitos usando template.
    - Ferramentas CASE: Utilizar VisualParadigm para documentação e modelagem.
    - Equipa: Todos os membros da equipa envolvidos.
2. **Desenho (3 semanas):**
    - Elaboração de diagramas UML: Desenvolver diagramas de sequência, diagrama de classes e outros diagramas relevantes.
    - Ferramentas CASE: Continuar a utilizar VisualParadigm para elaboração de diagramas.
    - Equipa: Especialistas em modelagem e design.
3. **Codificação (4 semanas):**
    - Desenvolvimento da aplicação móvel e da página web: Programação das funcionalidades, criação de interfaces de usuário.
    - Plataformas: Utilizar frameworks como React Native para a aplicação móvel e React.js para a página web.
    - Ferramentas CASE: IDEs como Visual Studio Code para desenvolvimento.
    - Equipa: Programadores responsáveis pela implementação.
4. **Testes (2 semanas):**
    - Testes de unidade, integração e sistema: Verificar se o software atende aos requisitos e está livre de bugs.
    - Ferramentas CASE: Utilizar ferramentas de automação de testes como Jest e Selenium.
    - Equipa: Testadores dedicados.
5. **Implantação (1 semana):**
    - Preparação para lançamento: Preparar ambientes de produção, fazer testes finais.
    - Ferramentas CASE: Ferramentas de integração contínua como Jenkins.
    - Equipa: DevOps responsáveis pela implantação.
6. **Manutenção (contínua):**
    - Correção de bugs, atualizações de segurança e melhorias de desempenho.
    - Ferramentas CASE: Ferramentas de monitoramento como New Relic.
    - Equipa: DevOps e programadores responsáveis pela manutenção.

3. Análise

Recolha de informação

Foi feita uma breve pesquisa na internet de outros sistemas semelhantes.

Análise da Informação/Documentação e Identificação dos Diferentes Tipos de Requisitos

3.1.1. Requisitos Funcionais:

Registar Sinais e Sintomas:

- O sistema deve permitir que os utilizadores registem sinais e sintomas de crises de saúde.
- Deve ser possível visualizar informações sobre sinais e sintomas já registados.

Registar Crises:

- O sistema deve permitir que os utilizadores registem crises de saúde com todos os detalhes pertinentes.
- Deve permitir a visualização de registos anteriores de crises.

Filmar Crises:

- O sistema deve permitir a gravação de vídeos durante uma crise de saúde.
- Deve armazenar e associar esses vídeos aos registos de crises correspondentes.

Aceder a Vídeos:

- O sistema deve permitir que os utilizadores visualizem vídeos de crises registados anteriormente.
- Deve garantir que apenas utilizadores autorizados possam aceder aos vídeos.

Gerir Calendário:

- O sistema deve permitir a programação de datas e eventos no calendário.
- Deve incluir a funcionalidade de confirmar a ida a consultas, realização de tratamentos, e toma de medicamentos.
- Deve permitir o registo e a leitura de notas associadas aos eventos do calendário.

Login e Verificação de Palavra-Passe:

- O sistema deve permitir que os utilizadores façam login utilizando um nome de utilizador e uma palavra-passe.
- Deve verificar a palavra-passe fornecida durante o login.

Requisitar Terapeuta:

- O sistema deve permitir que os utilizadores requisitem a assistência de um terapeuta.

Enviar Prescrição:

- O sistema deve permitir que os profissionais de saúde enviem prescrições aos utilizadores.

Participar em Fórum e Grupos:

- O sistema deve permitir que os utilizadores participem em fóruns e grupos de apoio.
- Deve possibilitar o envio e recebimento de mensagens dentro dos grupos.

Inserir e Ler Informação sobre Gestão de Crises:

- O sistema deve permitir que os utilizadores insiram e leiam informações sobre a gestão de crises.

3.1.2. Requisitos Não Funcionais:

Desempenho:

- O sistema deve ser capaz de processar e responder às solicitações dos utilizadores em tempo real, especialmente durante o registo de crises.

Segurança:

- O sistema deve garantir a segurança dos dados dos utilizadores, implementando criptografia para proteger informações sensíveis.
- Deve garantir que apenas utilizadores autorizados possam aceder a determinadas funcionalidades, como visualização de vídeos de crises.

Usabilidade:

- O sistema deve ser intuitivo e fácil de usar, com uma interface amigável que permita uma navegação simples e eficiente.

Confiabilidade:

- O sistema deve ser confiável, garantindo a disponibilidade e a consistência dos dados registados.
- Deve incluir mecanismos de backup para prevenir a perda de dados.

Compatibilidade:

- O sistema deve ser compatível com diferentes dispositivos e sistemas operativos, incluindo computadores, tablets e smartphones.

Escalabilidade:

- O sistema deve ser escalável para suportar um aumento no número de utilizadores e na quantidade de dados registados sem degradação de desempenho.

Manutenibilidade:

- O sistema deve ser fácil de manter, com uma arquitetura modular que permita a atualização e a adição de novas funcionalidades sem impactar negativamente o desempenho.

Diagrama de casos de uso

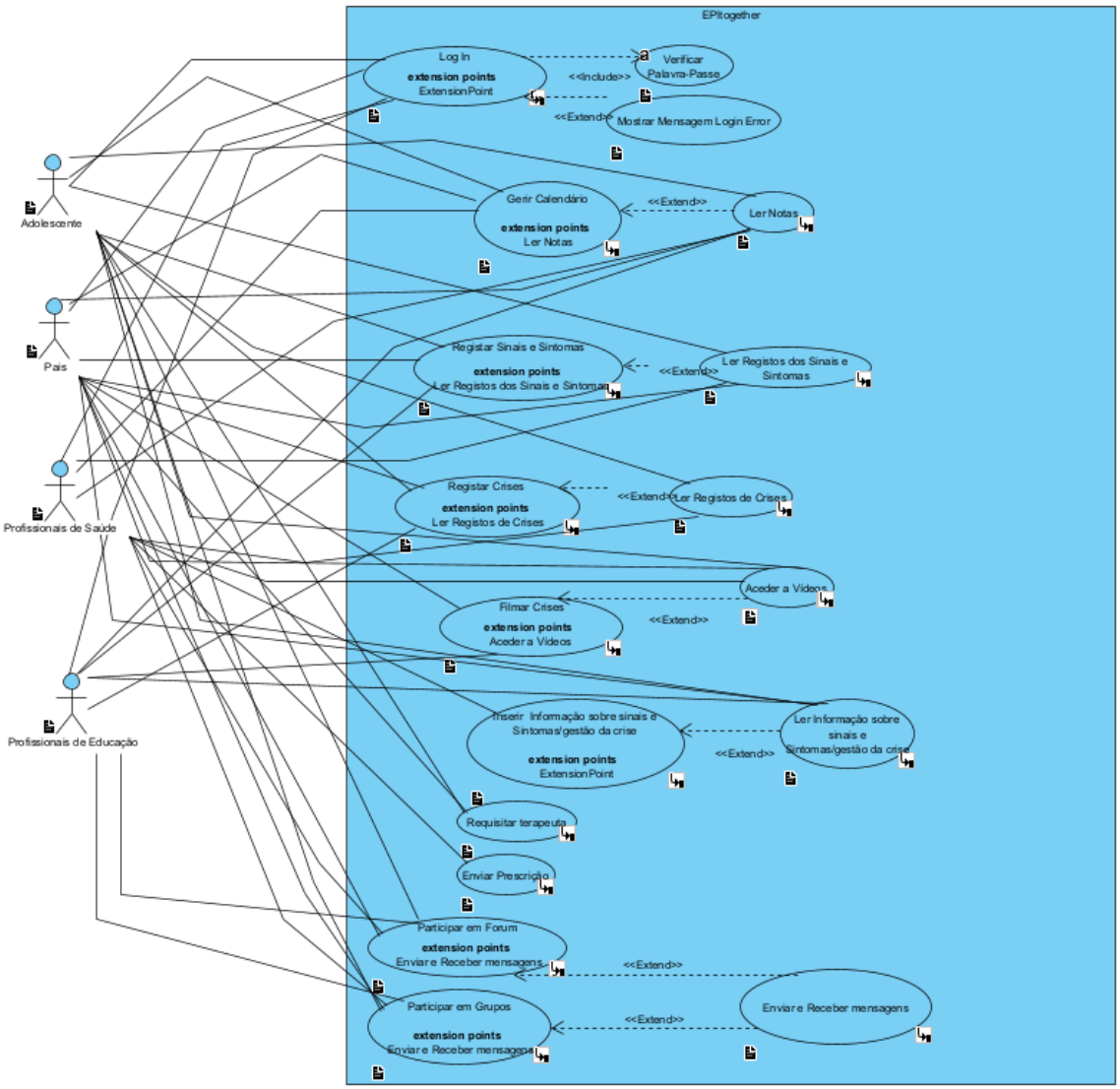

Figura 2

4. Desenho

Diagramas de sequência

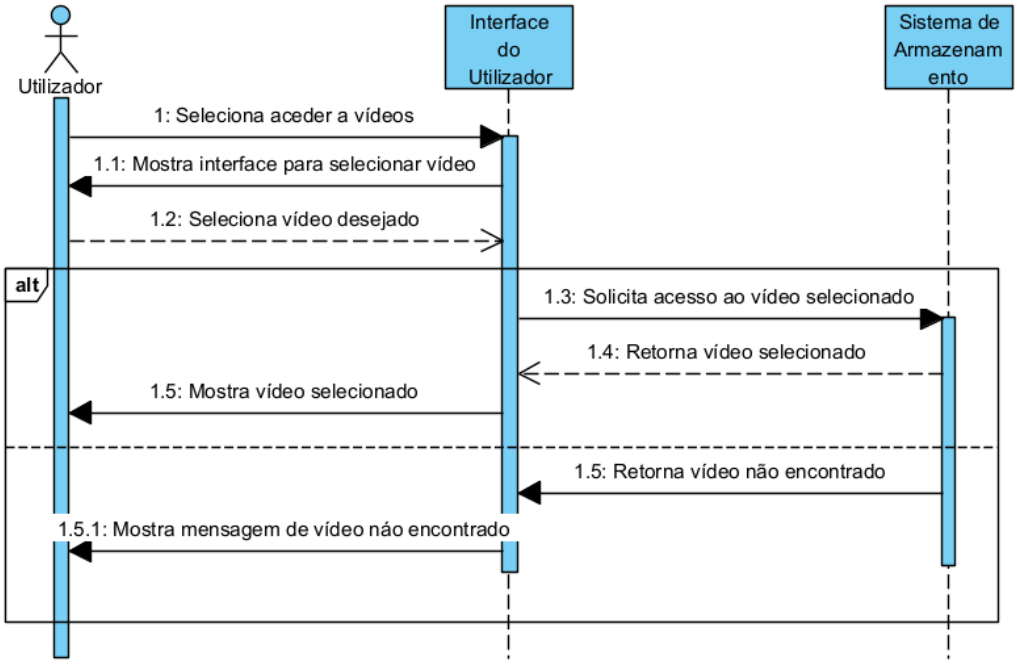

Figura 3-Aceder a Vídeos Sequence Diagram

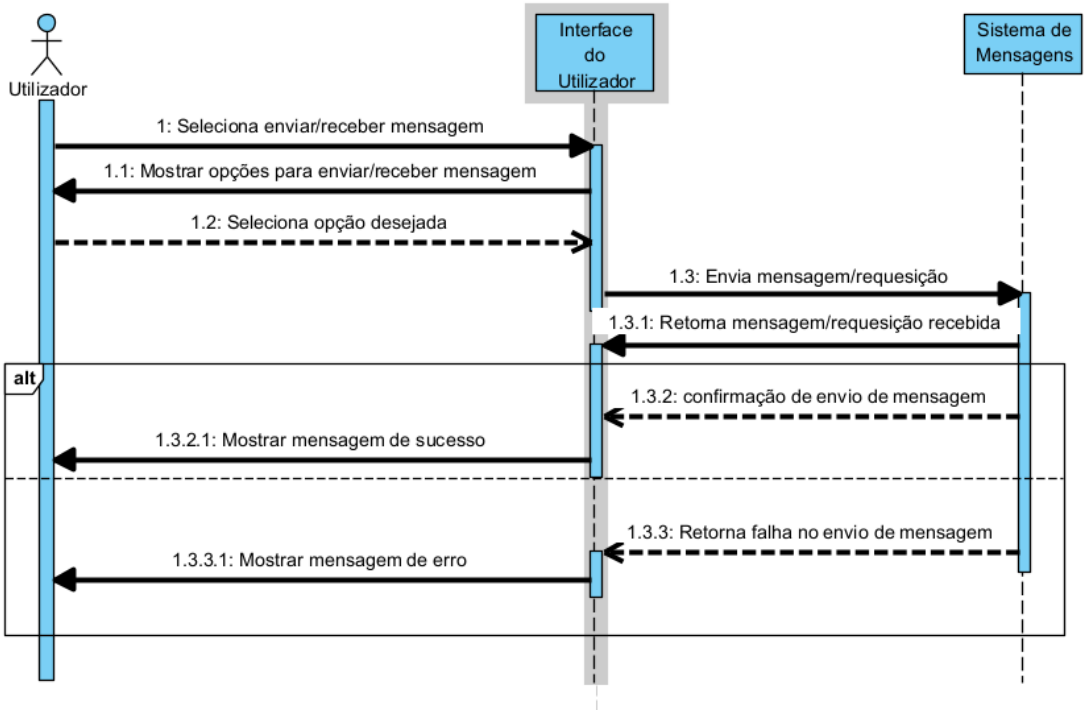

Figura 4-Enviar e Receber mensagens Sequence Diagram

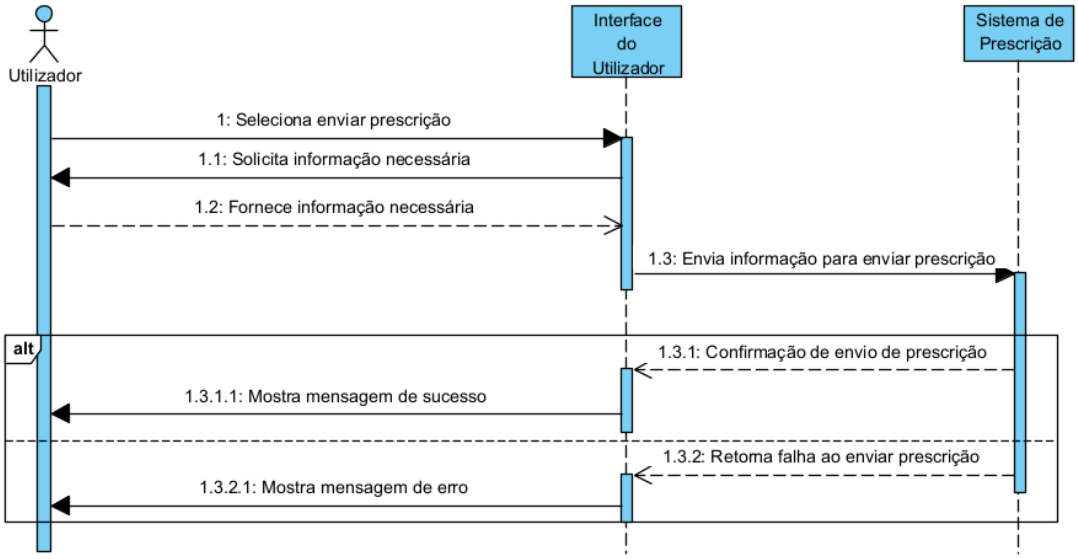

Figura 5-Enviar Prescrição Sequence Diagram

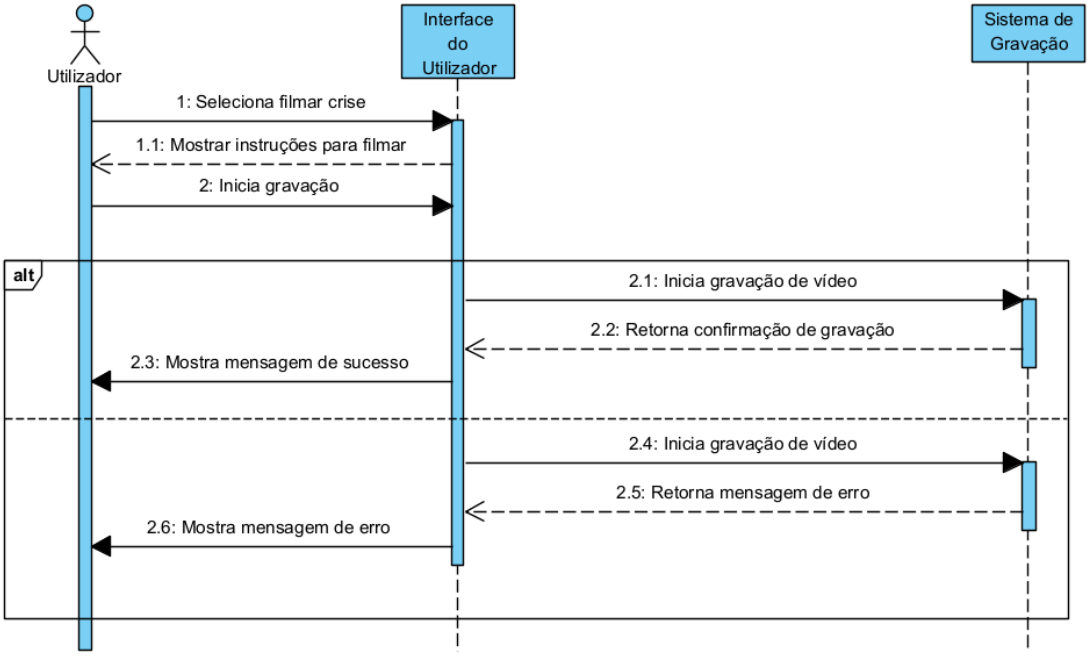

Figura 6-Filmar Crises Sequence Diagram

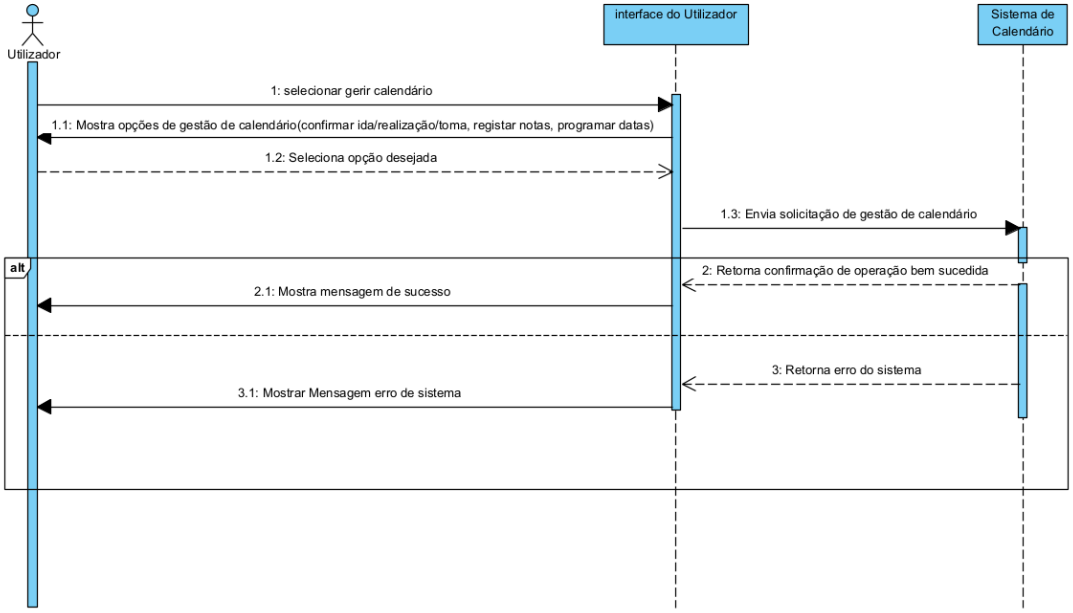

Figura 7-Gerir Calendário Sequence Diagram

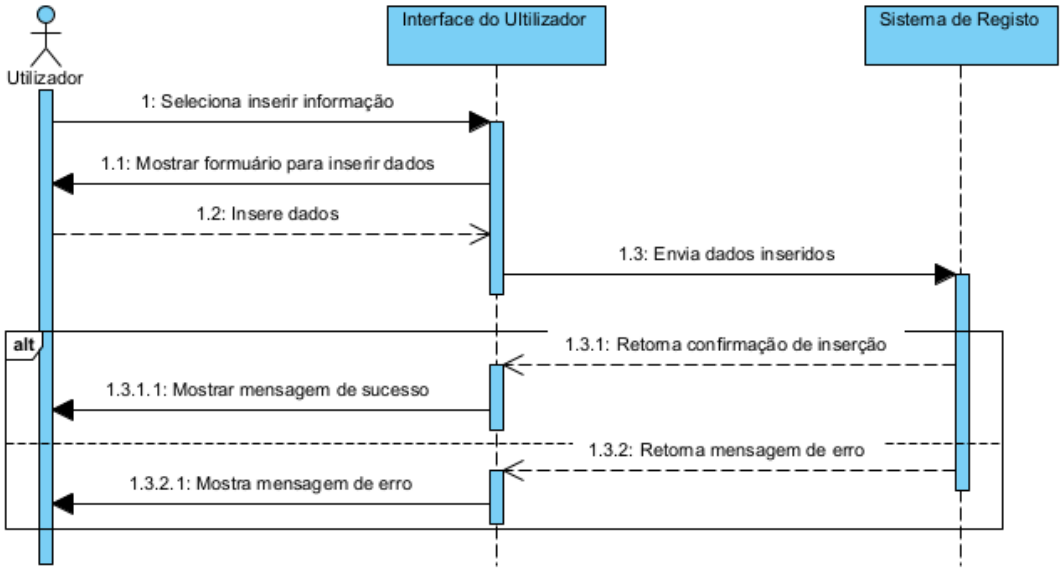

Figura 8-Inserir Informação sobre sinais e Sintomas/gestão da crise Sequence Diagram

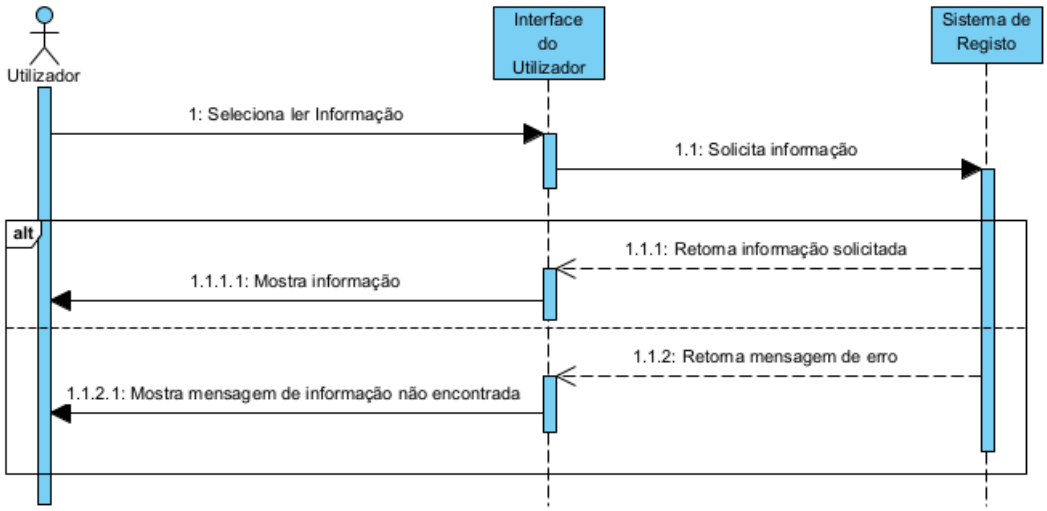

Figura 9-Ler Informação sobre sinais e Sintomas/gestão da crise Sequence Diagram

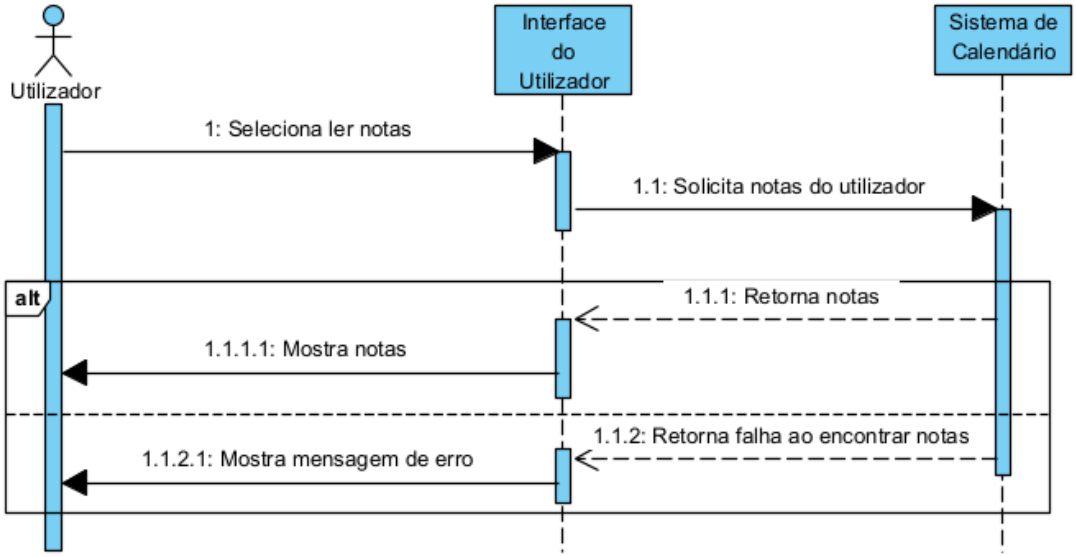

Figura 10-Ler Notas Sequence Diagram

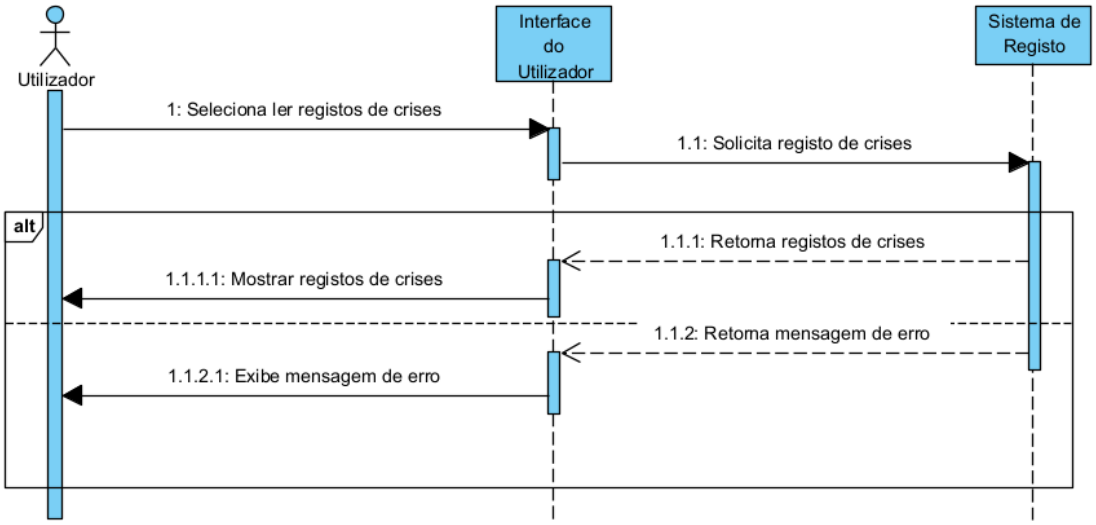

Figura 11-Ler Registos de Crises Sequence Diagram

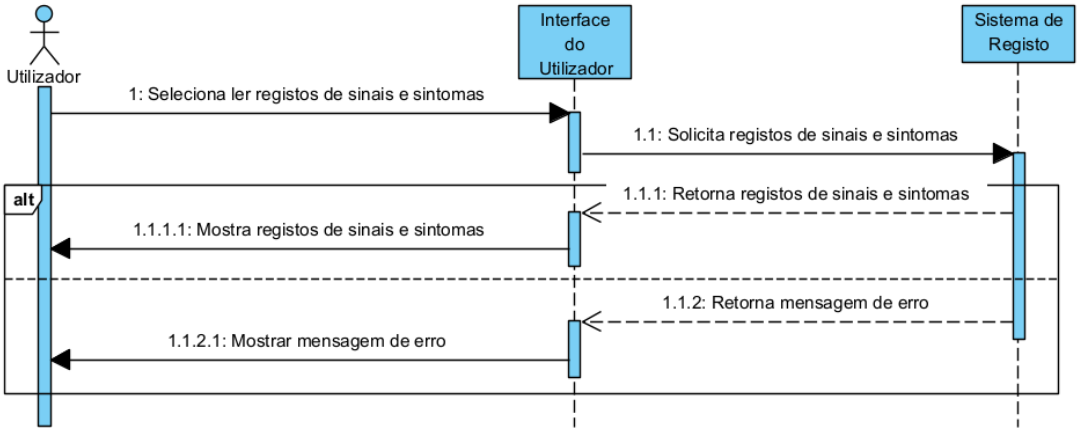

Figura 12-Ler Registos dos Sinais e Sintomas Sequence Diagram

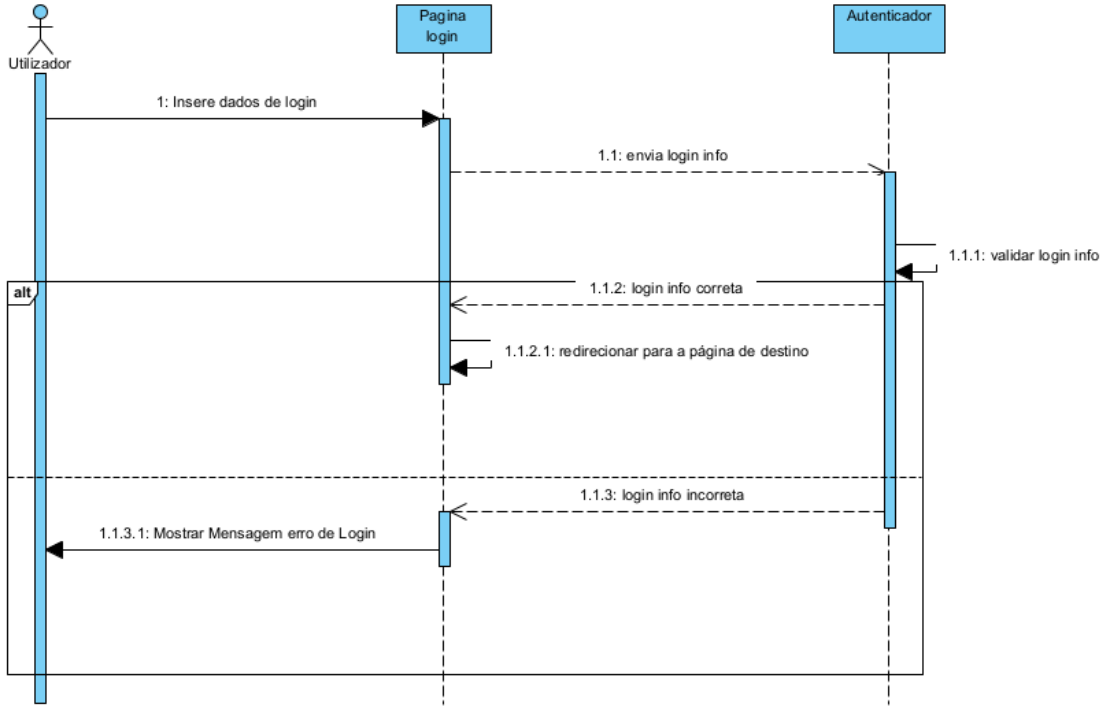

Figura 13-Log In Sequence Diagram

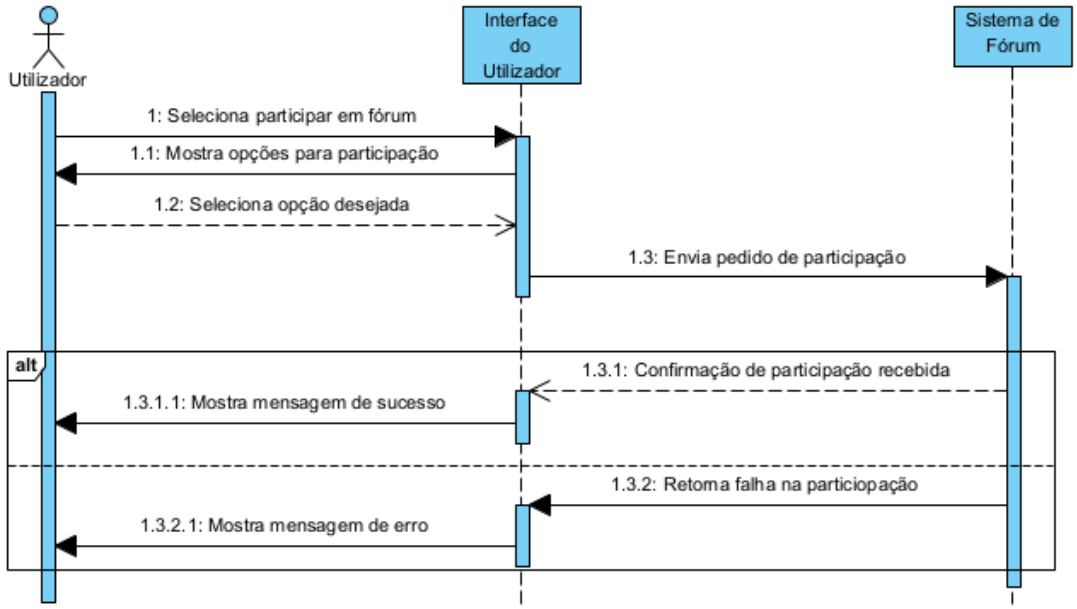

Figura 14- Participar em Forum Sequence Diagram

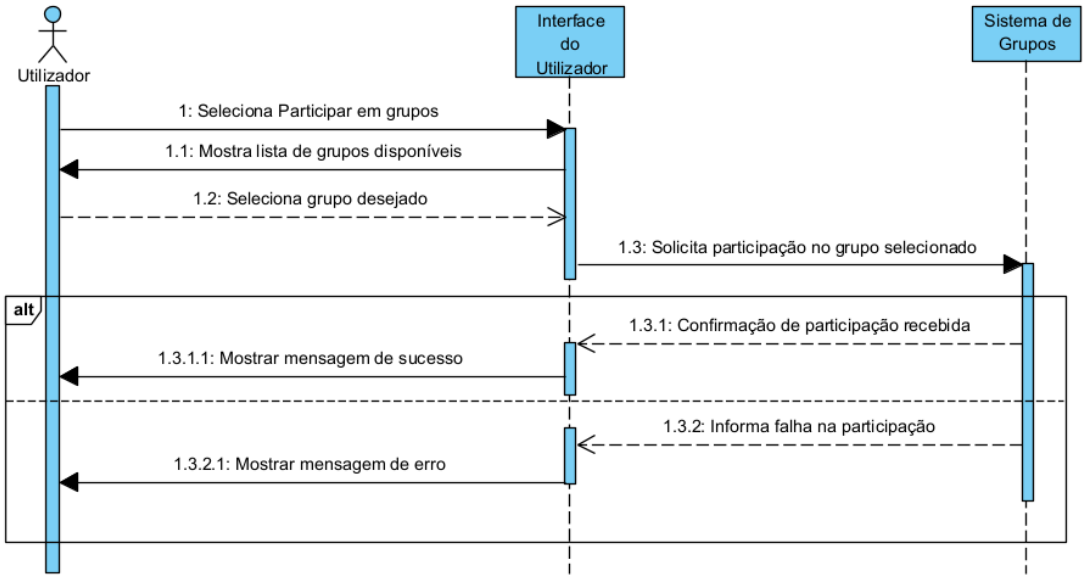

Figura 15- Participar em Grupos Sequence Diagram

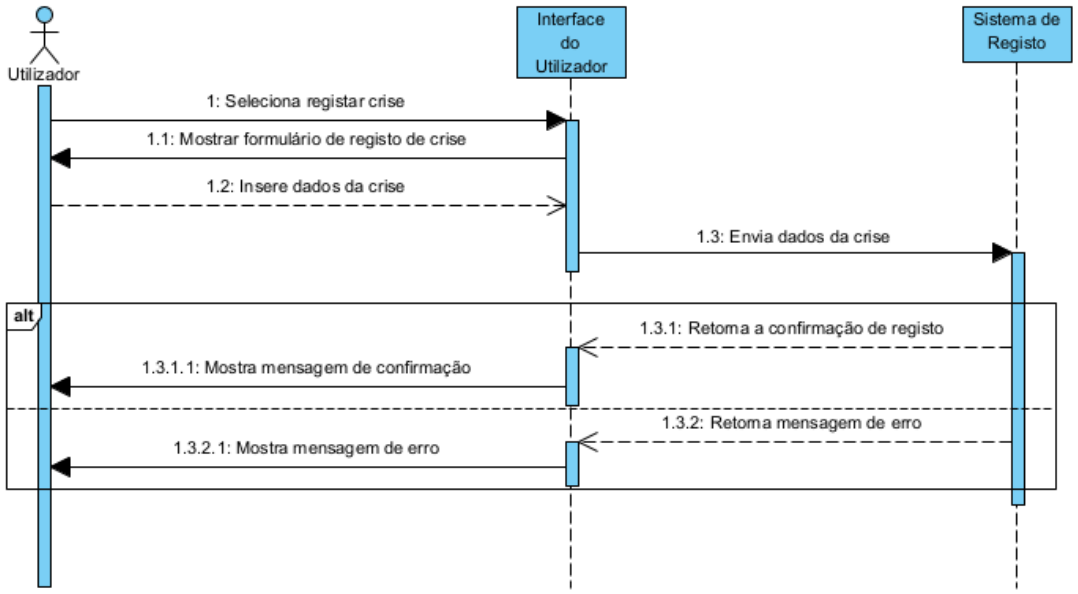

Figura 16- Registar Crises Sequence Diagram

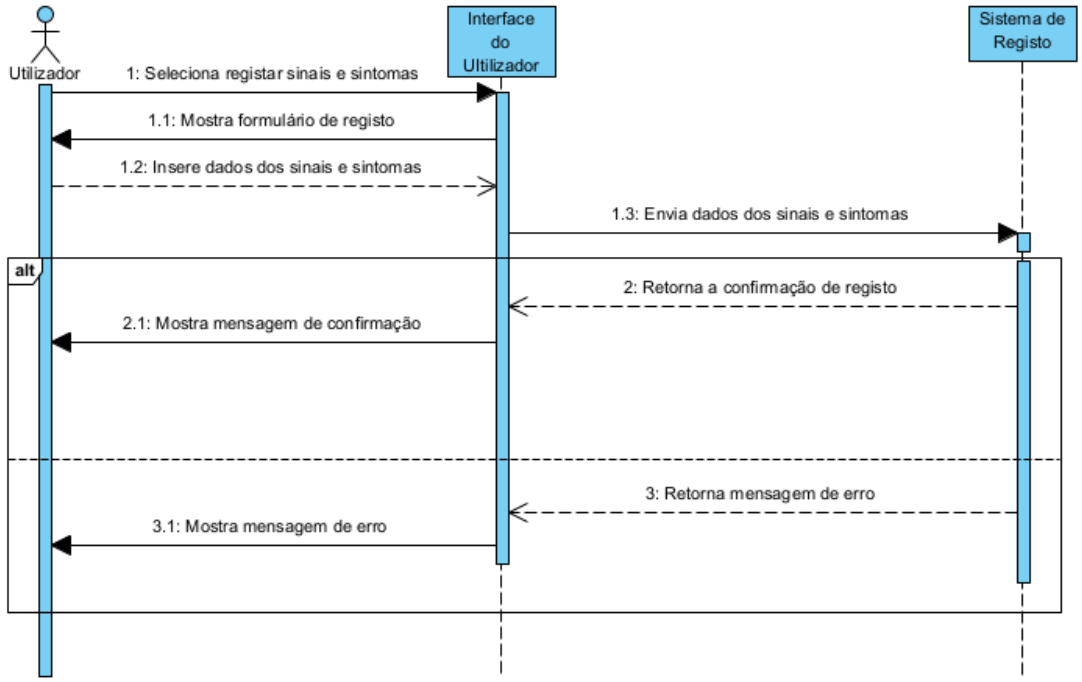

Figura 17- Registar Sinais e Sintomas Sequence Diagram

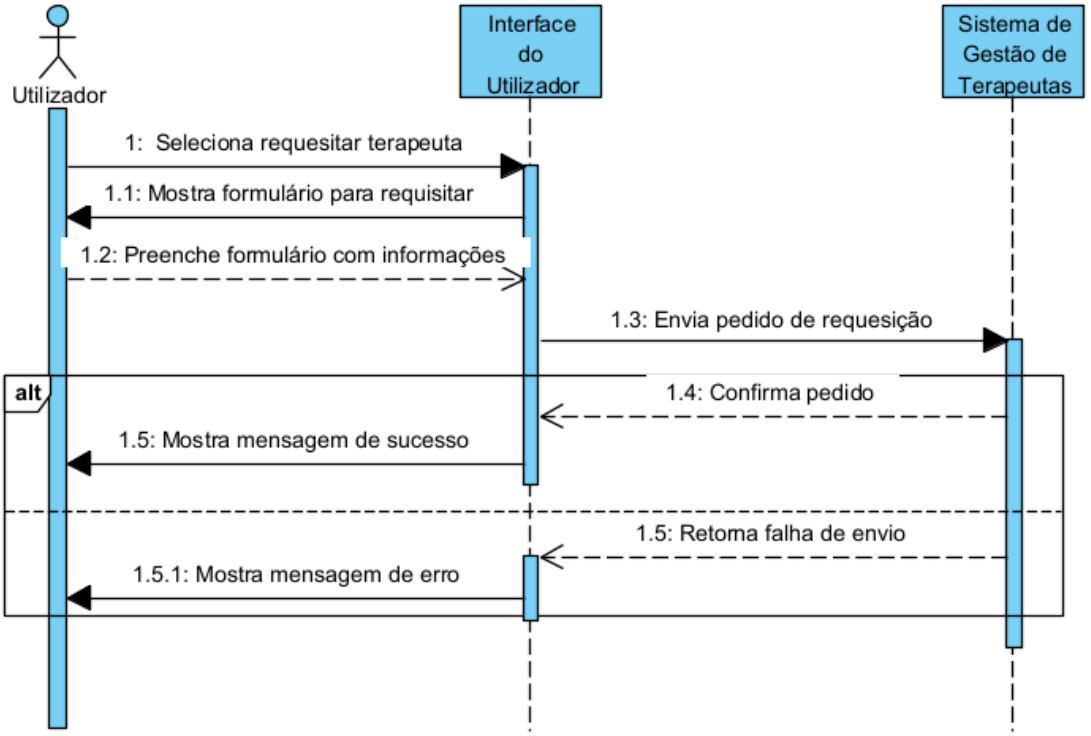

Figura 18- Requisitar terapeuta Sequence Diagram

Diagrama de classes

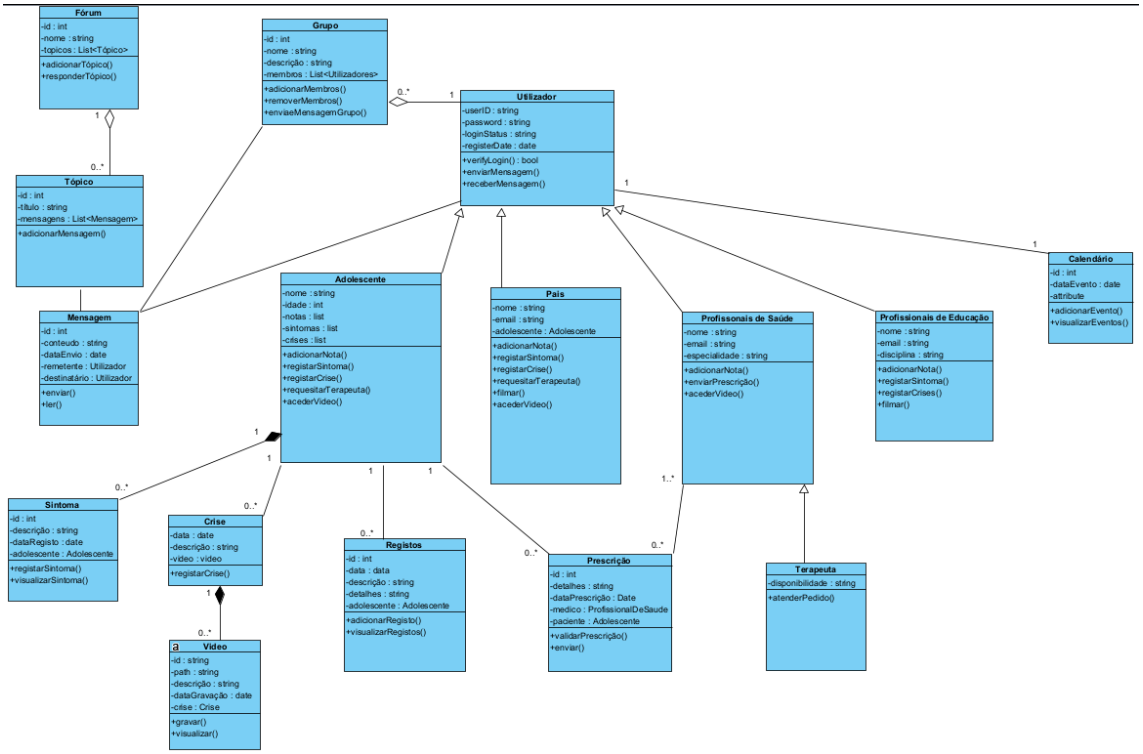

Figura 19-Diagrama de Classes

5. Gestão

Controlo de versões

Github.

O GitHub foi escolhido como a ferramenta de controle de versões para este projeto de devido à sua ampla adoção na indústria de tecnologia e às suas características robustas e amigáveis. Como uma plataforma baseada em nuvem, o GitHub permite que a equipe de desenvolvimento colabore de forma eficiente, mantendo um histórico detalhado de todas as alterações no código-fonte. Além disso, oferece recursos poderosos, como ramificações (branches), solicitações de pull (pull requests) e integração contínua, que facilitam a gestão do código e a implementação de boas práticas de desenvolvimento. A interface intuitiva do GitHub torna a navegação e a revisão do código simples para todos os membros da equipe, promovendo uma comunicação transparente e eficaz durante todo o ciclo de desenvolvimento do software.

6. Prazos previstos para execução das tarefas/atividades

- Análise: 2 semanas
- Desenho: 3 semanas
- Codificação: 4 semanas
- Testes: 2 semanas
- Implantação: 1 semana

7. Conclusões

Ao longo deste projeto, delineámos um plano abrangente para o desenvolvimento de software, estabelecendo um modelo claro de execução e definindo prazos específicos para cada fase. A adoção de uma abordagem estruturada e colaborativa permitiu-nos alcançar nossos objetivos dentro do cronograma estabelecido. Utilizando ferramentas e plataformas CASE adequadas, juntamente com a contribuição de todos os membros da equipe em suas respetivas áreas de especialização, estamos confiantes de que conseguiremos superar os desafios e entregar um produto final de alta qualidade. Este relatório serve como um marco importante no progresso do projeto, demonstrando nosso compromisso com a excelência e com a entrega de valor aos nossos clientes e usuários finais.