

Abstração geral do projeto fórum de monitorias

Henrique Z. Cochak¹, Jaasiel Abner Alves Sousa¹, Lucas T. de Oliveira¹, Mateus N. Ladeira¹

¹Departamento de Ciência da Computação (DCC)
Universidade do Estado de Santa Catarina (UDESC)

henrique_z_c@hotmail.com, mateusnladeira@gmail.com,

jabner101@gmail.com, lucas.thomas99@hotmail.com

Abstract. *The project aims to create a system that allows easy interaction between UDESC CCT students and monitors during monitoring hours. After attending, the student should evaluate the monitor, allowing teachers to follow more easily how the monitors are performing in the function of it's job.*

Resumo. *O projeto visa a criação de um sistema que permita a fácil interação entre alunos e monitores da UDESC CCT durante o horário de monitoria. Após o atendimento, o aluno deverá avaliar o monitor, permitindo que os professores acompanhem mais facilmente como os monitores estão se saindo na função.*

1. Introdução sobre o sistema

Um dos grandes problemas que afligem o meio universitário é a sobrecarga de trabalho sofrida pelos professores. Estes, via de regra, possuem uma rotina de intenso estudo e ensino, que, por vezes, lhes custa grande parte do tempo. Com isso, o acompanhamento dos monitores escolhidos se torna inviável, e os orientadores não conseguem verificar o quanto satisfeitos estão os alunos com o atendimento recebido nas monitorias. Por isso, faz-se necessária a criação de uma ferramenta capaz de fornecer a aproximação entre professores, alunos e monitores.

Nesse contexto, o projeto visa a criação de um fórum, onde os alunosão de publicar as dúvidas que surgirem em determinada disciplina. Quando a monitoria começar, o responsável responderá as dúvidas por ordem de envio. Ao fim do atendimento, o aluno deverá avaliar o monitor que o atendeu, deixando sua avaliação e um comentário.

Com isso, o professor poderá facilmente abrir o fórum e conferir o grau de satisfação dos seus alunos com os monitores. Se as avaliações forem predominantemente positivas, o orientador saberá que o monitor mantém um bom rendimento, tirando todas as dúvidas e tratando respeitosamente os alunos atendidos. Em contrapartida, se as avaliações forem ruins, ele saberá que deve tomar providências.

2. Arquitetura do sistema

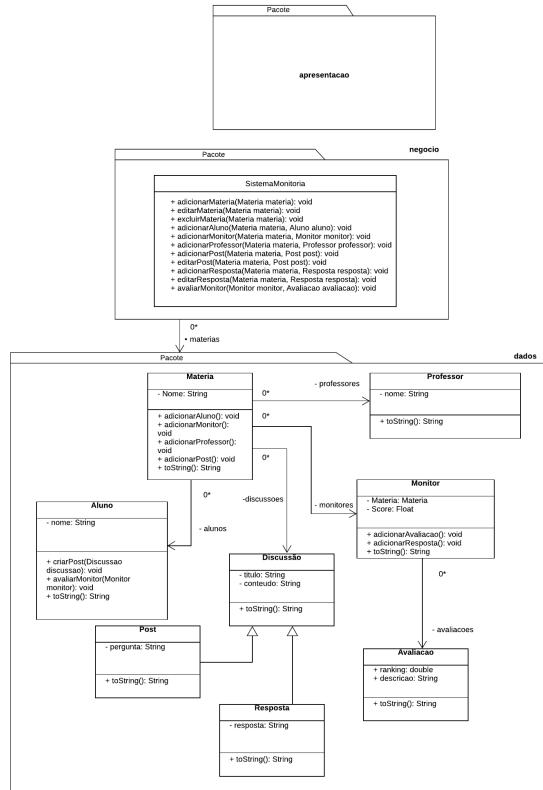


Figura 1. Projeto de arquitetura do sistema

3. Funcionalidades e abstração

Idealmente, este sistema utiliza de um grupo maior de funcionalidades do que existe até o momento mas para ser considerado viável para a matéria de Projeto de Programas, será descrito um subgrupo essencial considerando somente o que é viável para a implementação deste projeto no tempo limite.

A tabela que descreve os métodos com a funcionalidades (Tabela 1) descreve as funções, seu parâmetros, tipos de retorno e uma descrição geral.

Tabela 1. Métodos e funcionalidades do sistema

+ adicionarMateria(Materia materia); void	Permite que o administrador adicione uma nova matéria na lista de matérias do sistema de monitorias
+ excluirMateria(Materia materia); void	Permite que o administrador exclua uma matéria já existente da lista de matérias do sistema de monitorias
+ editarMateria(Materia materia); void	Permite que o administrador altere o nome da matéria passada como parâmetro
+ adicionarAluno(Materia materia, Aluno aluno); void	Permite que o administrador cadastre um novo aluno em uma lista de alunos da matéria
+ adicionarMonitor(Materia materia, Monitor monitor); void	Permite que o administrador adicione um novo monitor na lista de monitores da matéria
+ adicionarProfessor(Materia materia, Professor professor); void	Permite que o administrador adicione um novo professor na lista de professores da matéria
+ adicionarPost(Materia materia, Post post); void	Permite que o aluno adicione um novo post, ou seja, uma pergunta na lista de posts da matéria
+ editarPost(Materia materia, Post post); void	Permite que o aluno edite ou altere o post, ou seja, uma pergunta já feita por ele
+ adicionarResposta(Materia materia, Resposta resposta); void	Permite que o monitor da matéria em questão adicione uma resposta à lista de respostas da matéria
+ editarResposta(Materia materia, Resposta resposta); void	Permite que o monitor da matéria em questão edite ou altere uma resposta já feita por ele na lista de respostas da matéria
+ avaliarMonitor(Monitor monitor, Avaliacao avaliacao); void	Permite que o aluno avalie os monitores que atenderam sua discussão sendo possível escolher somente um monitor com a resposta

4. Diagramação

As diagramações são utilizadas para abstrair o sistema e demonstrar não só seus objetos e agrupá-los de modo correto mas também para como o mesmo se comporta dentro algumas ocasiões.

Para a conclusão deste trabalho, será utilizado diagramas baseado em conceitos mais vinculados a orientação a objetos tal como diagrama de classes, mas também é utilizado do Diagrama de Fluxo de Dados (DFD). Embora o DFD não seja uma parte formal da *Unified Modeling Language* (UML), que é uma linguagem de notação unificada no qual consiste em integrar um conjunto certo de diagramas, todos desenvolvidos para auxiliar e padronizar a documentação de sistemas extensos, esse diagrama pode ser usado para complementar os diagramas UML e dar uma visão adicional sobre o fluxo e os requisitos do sistema [Pressman 2005].

4.1. Diagrama de Classes

Dado um grupo de classes (Figura 2), é possível utilizar de diagramas no formato UML, especificamente o diagrama de classes (Figura 3) que descreve como todos os diferentes tipos de objetos dentro do seu sistema interagem entre si [Kim Hamilton 2006].

Materia	Aluno	Professor	Monitor	Avaliacao	Discussao	Resposta	Post
<ul style="list-style-type: none"> - id - Nome - List<Aluno> - List<Monitor> - List<Professor> - List<Discussao> + adicionarAluno() + adicionarMonitor() + adicionarProfessor() + adicionarPost() + toString() 	<ul style="list-style-type: none"> - cpf - senha - nome + login() + logout() + criarDiscussao() + editarDiscussao() + avaliarMonitor() + toString() 	<ul style="list-style-type: none"> - cpf - senha - nome + relatorioMonitor() + login() + logout() + toString() 	<ul style="list-style-type: none"> - cpf - senha - nome - Score + login() + logout() + adicionarAvalia() + adicionarRespo() + toString() 	<ul style="list-style-type: none"> - id + ranking + descricao + toString() 	<ul style="list-style-type: none"> - id - titulo - conteudo + toString() 	<ul style="list-style-type: none"> - resposta + toString() 	<ul style="list-style-type: none"> - pergunta + toString()

Figura 2. Classes do sistema

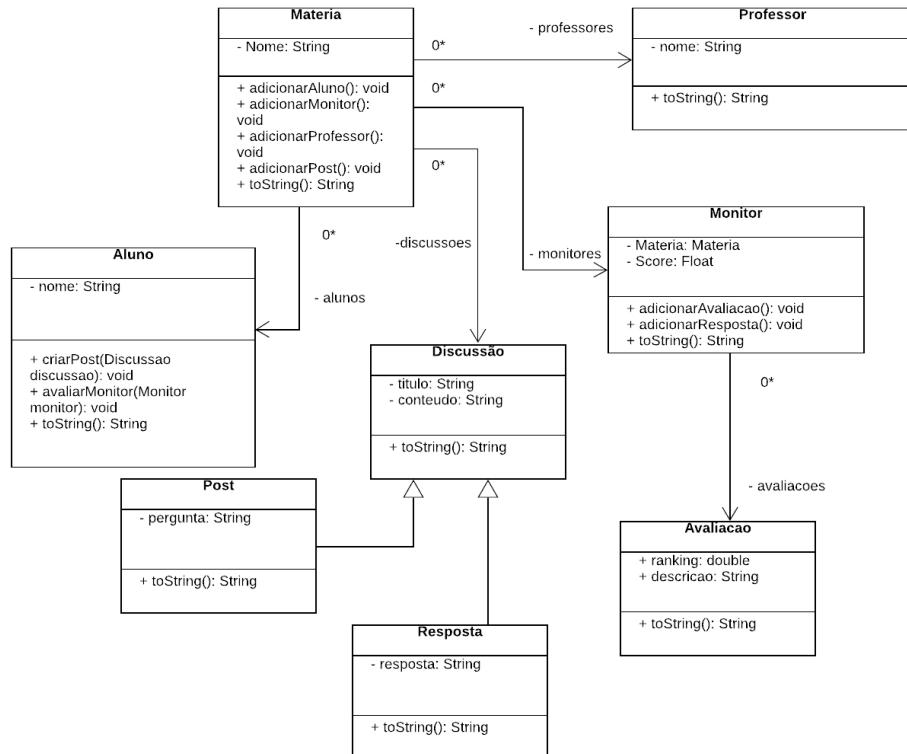


Figura 3. Diagrama de classes

O diagrama de classes exemplifica como o sistema flui e como seus dados estão intrínsecos com outros. Neste sistema, por exemplo, a comunicação de atributos entre

classes Aluno, Professor e Monitor ocorre mais frequentemente com a classe Matéria, significando que este é um sistema mais centralizado com tal classe.

4.2. Diagrama de Fluxo de Dados

O DFD adota uma visão de entrada, processo e saída de dados dentro do sistema. Os objetos de dados entram no software como entradas por entidades externas, representadas por retângulos, tais dados são transformados por elementos de processamento ou funções, representadas por círculos ou elipses e então os objetos de dados saem de outra forma do sistema, às vezes como informação para o próprio usuário ou simplesmente armazenados em banco de dados, representados por duas retas paralelas [Pressman 2005].

Um DFD possui níveis, sendo o DFD de nível zero (Figura 4), também conhecido como diagrama de contexto. Este diagrama abstrai o sistema de sua forma mais simples e visão mais macro do programa, deixando para quem o ler enxergar o seu motivo de existência e quais suas principais funcionalidades de modo mais próximo de pessoas não próximas da área de computação ou sistemas.

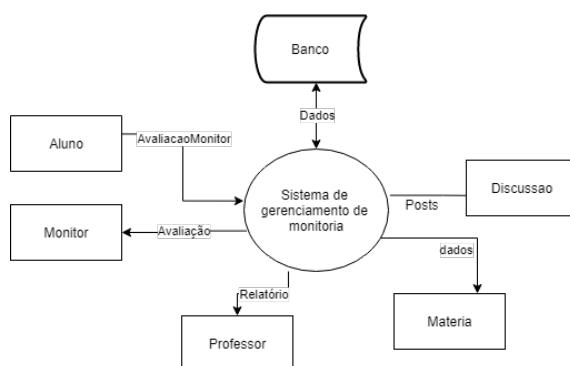


Figura 4. Diagrama de Fluxo de Dados nível 0

O diagrama de contexto desse sistema representa a função da entidade externa aluno fazendo avaliações de monitores, também entidades externas no qual respondem as discussões criadas pelos alunos e acabam sendo avaliados pelos mesmos. Um professor que orienta o monitor, pode pedir relatório sobre seu ranking atual para visualizar o esforço feito pelo monitor perante dúvidas de monitoria.

Após utilizar do DFD de nível zero, é criado DFDs mais específicos (Figura 5), onde existe a possibilidade de que cada processo pode ser quebrado em subprocessos, cada um criando o seu DFD, tal como ocorre com o processo 2 (Figura 6) e processo 3 (Figura 7) que acabam requisitando de subprocessos para explanar melhor como os dados estão sendo transmitidos no sistema, ficando cada vez mais próximo do sistema real.

O processo 2 virou 3 subprocessos menores, o processo de criar discussão, o processo de alterar discussão e o processo de responder uma discussão. Cada processo possui o seu gama de atributos, entidades externas e armazenamento de dados que fazem do comportamento esperado do sistema dado certo grupo de entrada, certo processo que compõe um grupo de funções e sua saída de dados como retorno para entidades externas ou sendo armazenadas.

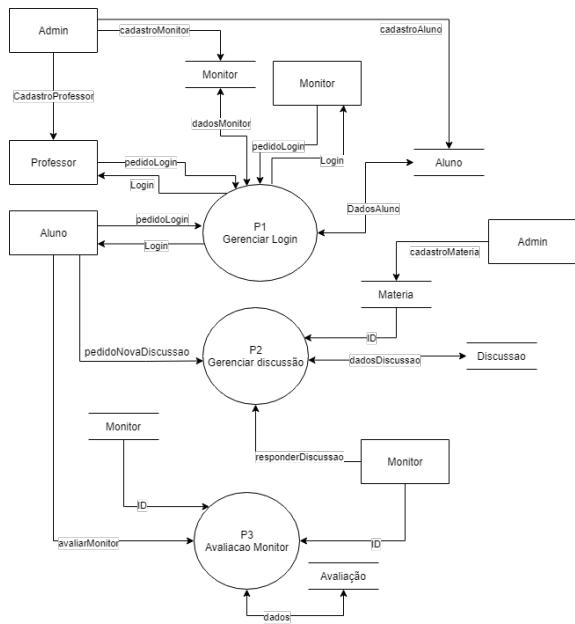


Figura 5. Diagrama de Fluxo de Dados nível 1

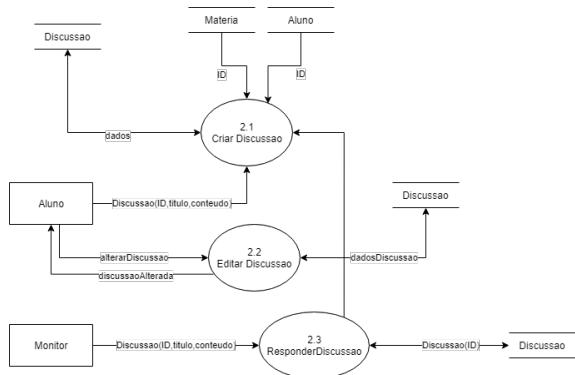


Figura 6. Diagrama de Fluxo de Dados nível 2.1

O processo 3 "explodiu" somente em outros 2 subprocessos, o de selecionar resposta e o de avaliar resposta. Ao comparar o processo 3 com seus subprocessos, percebe-se que ficou relativamente mais simples de se analisar de forma correta e mais concisa como os dados são atribuídos e o caminho que percorrem.

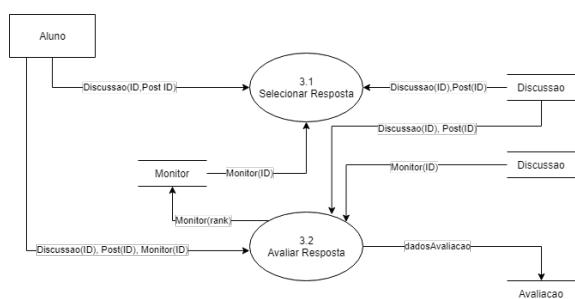


Figura 7. Diagrama de Fluxo de Dados nível 3.1

4.3. Diagrama de Sequência

Os diagramas de sequência tratam de capturar a ordem das interações entre as partes do sistema. Usando um diagrama de sequência, você pode descrever quais interações serão acionadas quando um caso de uso específico for executado e em que ordem essas interações ocorrerão. O diagrama de sequência mostra muitas informações sobre uma interação. Ele utiliza do tempo para mostrar como a interação ocorre conforme o processo continua [Kim Hamilton 2006].

Demonstra-se o comportamento do sistema perante algumas funcionalidades mais essenciais utilizando do diagrama de sequência, no qual documenta a sequência de eventos tomado pelo sistema de modo visual, o que torna o processo de criação do mesmo descomplicado. As funcionalidades exemplificadas são focadas na entidade Aluno e são a de login (Figura 8), de criar uma discussão (Figura 9), de alterar uma discussão (Figura 10) e de avaliação de uma resposta dada por um monitor (Figura 11).

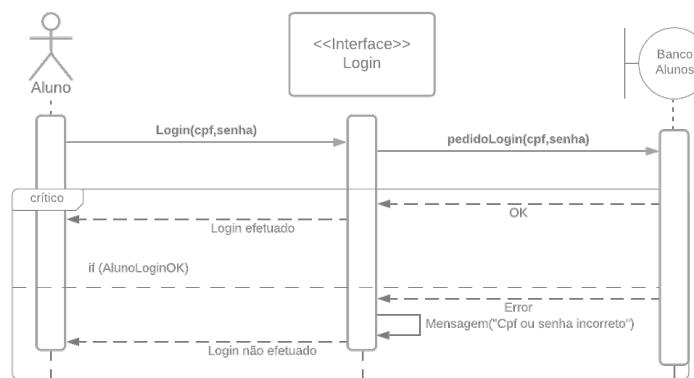


Figura 8. Diagrama de sequência da funcionalidade Login

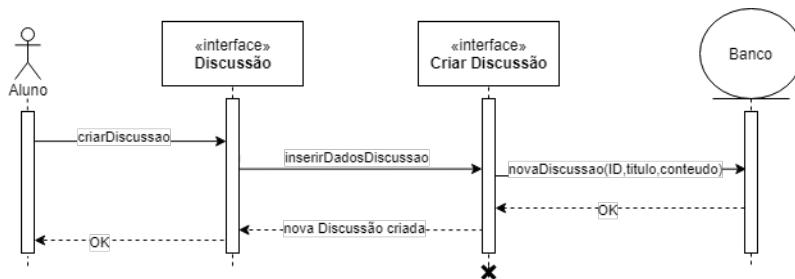


Figura 9. Diagrama de sequência da funcionalidade criar discussão

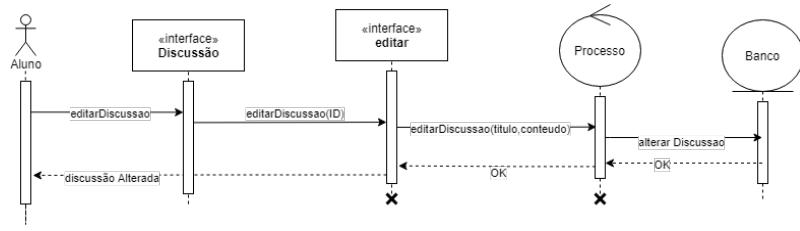


Figura 10. Diagrama de sequência da funcionalidade editar discussão

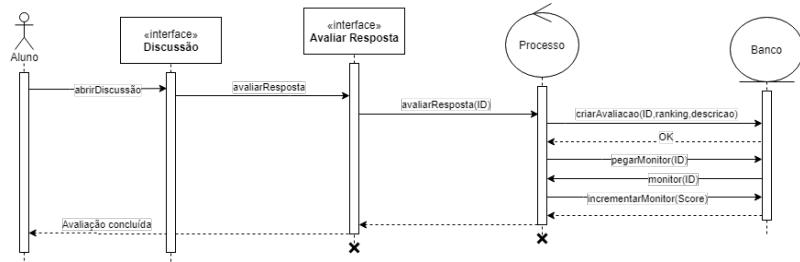


Figura 11. Diagrama de sequência da funcionalidade avaliar monitor

5. Interfaces do sistema

A interface possui como funcionalidade interagir entre duas ou mais partes distintas do sistema no qual não existe uma conexão direta entre elas. As interfaces existentes são a de login (Figura 12), menu principal de aluno (Figura 13), menu principal do monitor (Figura 14), menu principal do professor (Figura 15), menu principal do administrador (Figura 16), a interface de criação de discussão pelo aluno (Figura 17), a interface de visualização de discussão (Figura 18)



Figura 12. Tela de Login

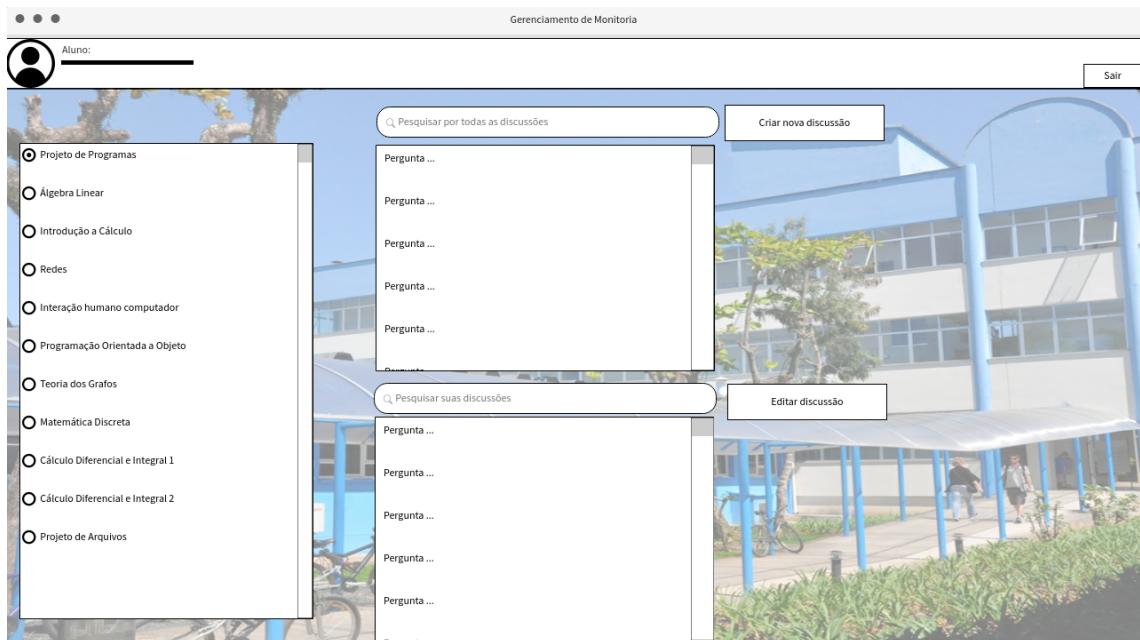


Figura 13. Tela principal do aluno

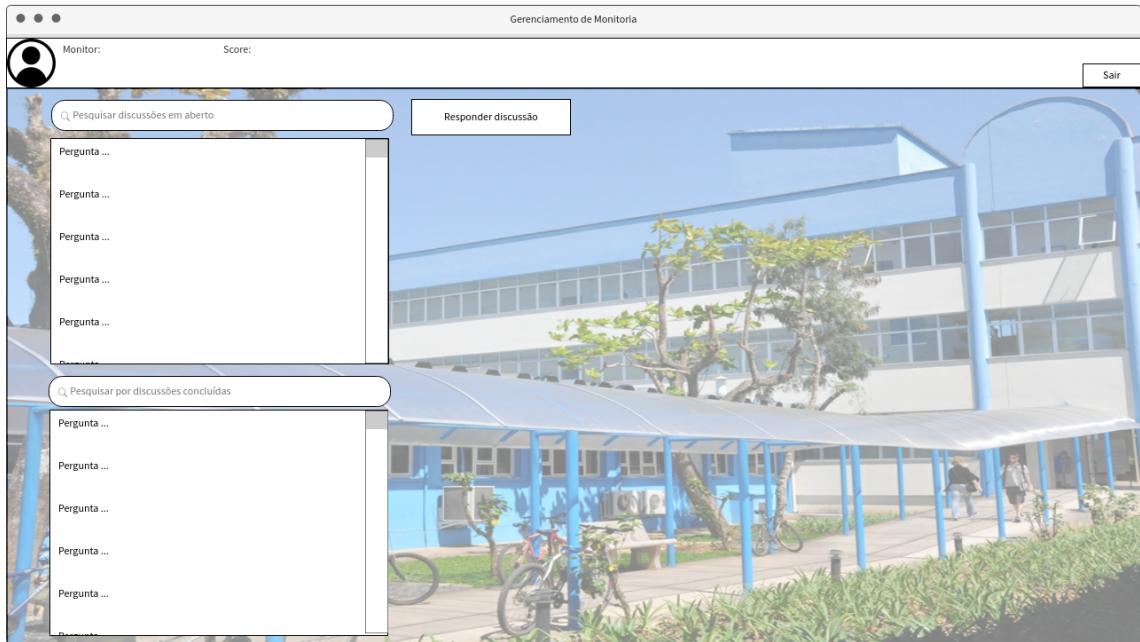


Figura 14. Tela principal do monitor

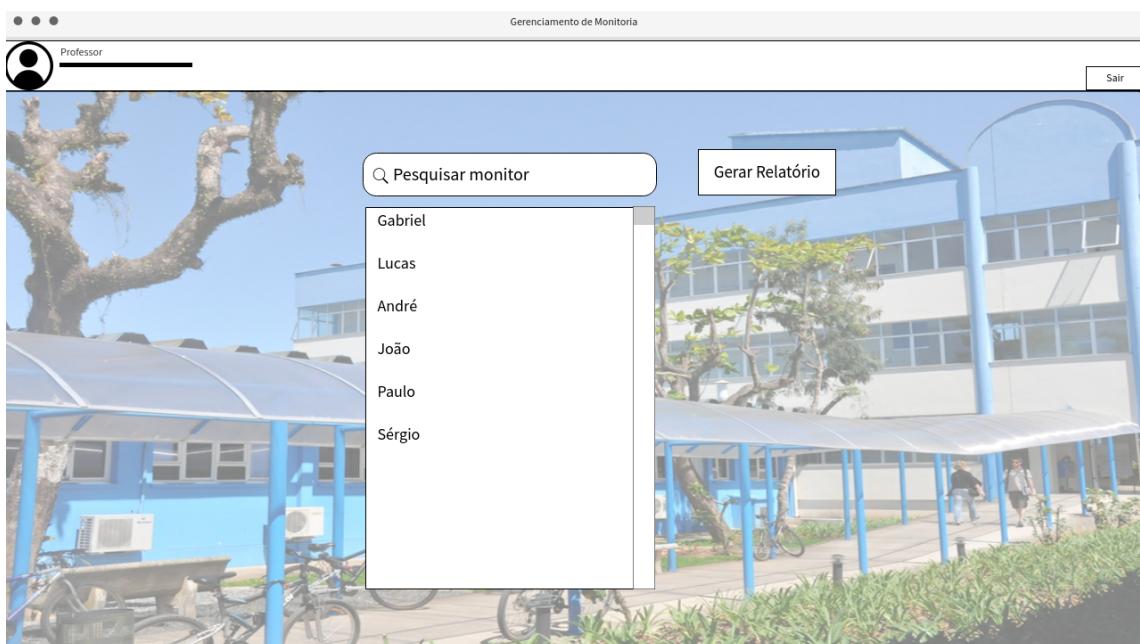


Figura 15. Tela principal do professor

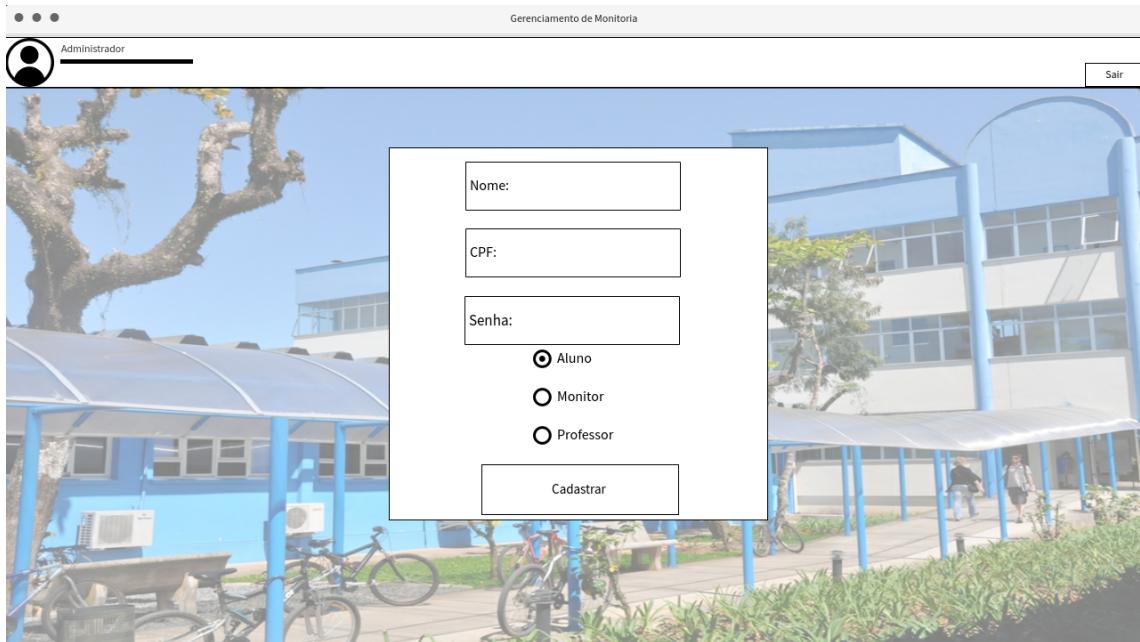


Figura 16. Tela principal do administrador

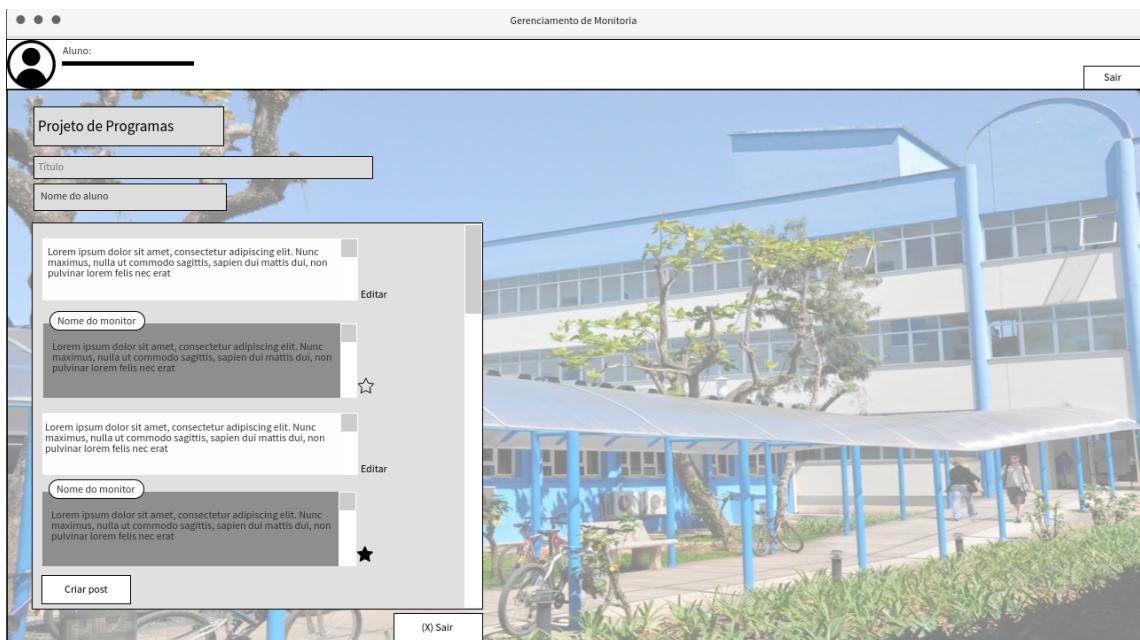


Figura 17. Tela de criação de discussão

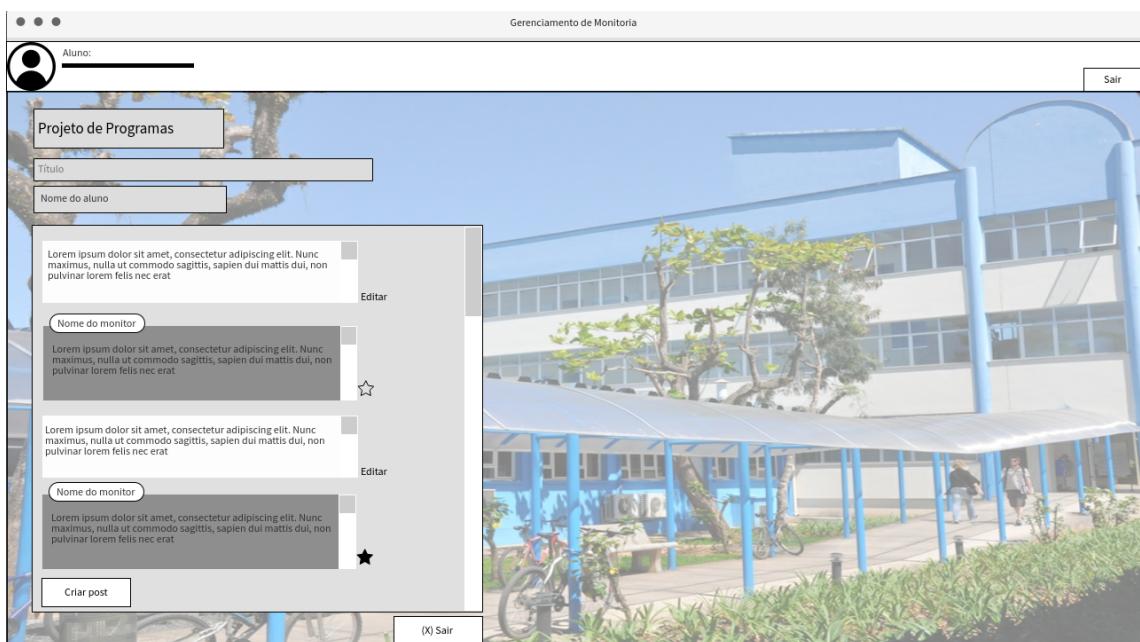


Figura 18. Tela de criação de discussão

6. Considerações

Ao ser integrado, o sistema poderá melhorar a eficiência dos monitores de forma efetiva, uma vez que as dúvidas colocadas no fórum ficarão permanentemente disponíveis ao acesso de futuros alunos. Além disso, os professores terão uma forma rápida e prática de avaliar seus monitores, criando um ambiente mais claro e simples.

Por outro lado, o fórum também permitirá que alunos e professores respondam as dúvidas e discussões em geral, permitindo uma avaliação por parte do professor para possíveis futuros monitores.

Referências

- Kim Hamilton, R. M. (2006). *Learning uml 2.0*. Publisher: O'Reilly, Pub Date: April.
- Pressman, R. S. (2005). *Software engineering: a practitioner's approach*. Palgrave macmillan.