

INSTITUTO FEDERAL GOIANO
RIO VERDE

GUSTAVO MOURA, JOÃO SARAIVA, WÉSIO COELHO

Software de Agendamento de Atendimento e Monitoria - Null Doubt

Rio Verde
2022

INSTITUTO FEDERAL GOIANO
RIO VERDE

**AUTORIZAÇÃO PARA PUBLICAÇÃO DE TESE EM
FORMATO ELETRÔNICO**

Na qualidade de titular dos direitos de autor, **AUTORIZO** o Rio Verde da Instituto Federal Goiano – IFGoiano a reproduzir, inclusive em outro formato ou mídia e através de armazenamento permanente ou temporário, bem como a publicar na rede mundial de computadores (*Internet*) e na biblioteca virtual da IFGoiano, entendendo-se os termos “reproduzir” e “publicar” conforme definições dos incisos VI e I, respectivamente, do artigo 5º da Lei nº 9610/98 de 10/02/1998, a obra abaixo especificada, sem que me seja devido pagamento a título de direitos autorais, desde que a reprodução e/ou publicação tenham a finalidade exclusiva de uso por quem a consulta, e a título de divulgação da produção acadêmica gerada pela Universidade, a partir desta data.

Título: Software de Agendamento de Atendimento e Monitoria - Null Doubt

Autor(a): Gustavo Moura, João Saraiva, Wésio Coelho

Rio Verde , 07 de Agosto de 2022.

Gustavo Moura, João Saraiva, Wésio Coelho – Autor

Dr. Adriano Soares de Oliveira Bailão – Orientador

GUSTAVO MOURA, JOÃO SARAIVA, WÉSIO COELHO

Software de Agendamento de Atendimento e Monitoria - Null Doubt

Documentação de Software apresentada ao Programa de Graduação do Rio Verde da Instituto Federal Goiano, como requisito parcial.

Área de concentração: Ciência da Computação.

Orientador: Prof. Dr. Adriano Soares de Oliveira Bailão

Rio Verde
2022

Resumo

Esse trabalho apresenta um desenvolvimento de um software baseado em um agendamento de atendimentos e monitorias. O software será desenvolvido para facilitar a gestão de atendimentos e monitorias entre professores e alunos de escolas e universidades. Para levantar os requisitos dos sistemas foi utilizado como exemplo o Instituto Federal Goiano - campus Rio Verde. O sistema será desenvolvido para que seja acessado de todos os tipos de máquinas, ele trará uma interface amigável e dinâmica para o usuário. O estudante terá acesso aos atendimentos e monitorias disponíveis no momento, para que possa sanar suas dúvidas diante com o auxílio de um professor ou monitor, aumentando assim o seu rendimento escolar.

Palavras-chave

Software de agendamento, monitoria, atendimento.

Abstract

his work presents a development of a software based on a schedule of attendances and monitoring. The software will be developed to facilitate the management of attendance and monitoring between teachers and students from schools and universities. To survey the requirements of the systems, the Instituto Federal Goiano - campus Rio Verde was used as an example. The system will be developed so that it can be accessed from all types of machines, it will bring a friendly and dynamic interface to the user. The student will have access to the services and monitoring available at the moment, so that he can solve his doubts with the help of a teacher or monitor, thus increasing his school performance.

Keywords

Scheduling, monitoring, attendance software.

Sumário

Lista de Figuras	6
Lista de Tabelas	8
1 Introdução	9
2 Revisão Bibliográfica	11
3 Agrupamentos de objetos de dados	12
3.1 Banco de dados	12
3.2 SQL de Geração do Banco	13
3.3 Diagramas de Sequência	17
3.4 Diagrama de Classes	29
4 Método proposto	30
4.1 Colaboradores	30
4.2 Ferramentas Utilizadas	30
4.3 Requisitos do Sistema	31
4.3.1 Requisitos Funcionais	32
4.3.2 Requisitos Não Funcionais	37
4.4 Arquitetura do Sistema	39
5 Experimentos e análise de desempenho	40
5.1 Planos de Teste	40
5.1.1 Login	41
5.1.2 Cadastros	42
5.1.3 Atualização de Cadastros	43
5.1.4 Leitura de Cadastros	44
5.1.5 Desativação e Ativação de Cadastros	45
5.1.6 Visualizar Próximos Atendimentos	46
5.1.7 Gerenciar Horários, Data e Local de Atendimento	47
5.1.8 Promover e Rebaixar Discente à Monitor	48
5.1.9 CRUD Horários para a Monitoria do Monitor	49
5.1.10 Visualizar Próximas Monitorias	50
5.1.11 Agendar Atendimento ou Monitoria	51
5.1.12 Desmarcar Agendamento	52
5.1.13 Teste de Desempenho	52
5.1.14 Teste de Segurança	53
5.2 Relatório de Teste	53

5.2.1	Login	53
5.2.2	Cadastros	54
5.2.3	Atualização de Cadastros	55
5.2.4	Leituras de Cadastros	55
5.2.5	Desativação e Reativação de cadastros	56
5.2.6	Visualizar Próximos Atendimentos	56
5.2.7	Gerenciar Horários, Data e Local de Atendimento	58
5.2.8	Promover e Rebaixar Discente à Monitor	59
5.2.9	CRUD Horários para a Monitoria do Monitor	60
5.2.10	Visualizar Próximas Monitorias	61
5.2.11	Agendar Atendimento ou Monitoria	62
5.2.12	Desmarcar Agendamento	64
5.2.13	Teste de Desempenho	65
5.2.14	Teste de Segurança	65
6	Conclusões	66
	Referências Bibliográficas	67

Lista de Figuras

3.1	Modelo Lógico do Banco de Dados	12
3.2	Super-Usuário – Listar Tudo	18
3.3	Super-Usuário – Gerenciar Usuários	19
3.4	Super-Usuário – Gerenciar Cursos	19
3.5	Super-Usuário – Gerenciar Locais	20
3.6	Super-Usuário – Gerenciar Disciplinas	20
3.7	Discente – Agendar Atendimento ou Monitoria	21
3.8	Discente – Desmarcar Agendamento	21
3.9	Discente Monitor – Visualizar Próximas Monitorias	22
3.10	Discente Monitor – Visualizar Discentes Agendados	23
3.11	Docente – Visualizar Próximos Atendimentos	24
3.12	Docente – Visualizar Discentes Agendados	25
3.13	Docente – CRUD Horários de Atendimento	26
3.14	Docente – Adicionar ou Remover Monitor	27
3.15	Docente – CRUD Horários de Monitoria	28
3.16	Diagrama de Classe	29
4.1	Diagrama de Casos de Uso	32
5.1	Login efetuado com sucesso	53
5.2	Erro de login, com dados errados	54
5.3	Cadastro de Discentes	54
5.4	Cadastro de Discentes	55
5.5	Cadastro feito com sucesso	55
5.6	Leitura de Docentes	56
5.7	Visualização de Atendimentos Docentes	57
5.8	Visualização de Atendimentos Banco de Dados	57
5.9	Visualização de Atendimentos Discentes	58
5.10	Gerenciar Atendimentos	58
5.11	Notificação Após Gerenciar Atendimentos	59
5.12	Promover Monitor	59
5.13	Promover Monitor	60
5.14	Adicionar Horário de Monitoria	60
5.15	Notificação após adicionar Horário de Monitoria	61
5.16	Visualizar Monitorias / Tabela de monitorias	61
5.17	Visualizar Monitorias / Tela do Discente promovido à Monitor	62
5.18	Agendar Atendimento - Discente	62
5.19	Agendamento de Atendimento Efetuado - Discente	63
5.20	Agendamento de Atendimento Efetuado - Discente	63

5.21	Agendamento do Discente no Atendimento	64
5.22	Desmarcar Atendimento - Discente	64
5.23	Desmarcar Atendimento - Discente	65

Lista de Tabelas

4.1	REF01 – Gerenciar Cursos	33
4.2	REF02 – Gerenciar Usuários	33
4.3	REF03 - Gerenciar Locais	34
4.4	REF04 - Gerenciar Disciplinas	34
4.5	REF05 - Gerenciar Próximos Atendimentos Semanais	34
4.6	REF06 - CRUD Horários de Atendimento	34
4.7	REF07 -Adicionar ou Remover Monitor	35
4.8	REF08 - Crud Horários de Monitorias	35
4.9	REF09 - Visualizar Próximas Monitorias	35
4.10	REF10 - Marcar Atendimento ou Monitoria	35
4.11	REF11 - Desmarcar Agendamento	36
4.12	REF12 - Listar Tudo	36
4.13	REF13 - Visualizar Discentes Agendados	36
4.14	Usabilidade	37
4.15	Segurança	37
4.16	Desempenho	37
4.17	Desenvolvimento	37
4.18	Manuabilidade	38
4.19	Facilidade de Uso	38
4.20	Robustez	38
4.21	MVC - Interações	39

Introdução

Em uma universidade o processo de aprendizagem pode gerar muitas dúvidas. Para minimizar essas dúvidas dos alunos, existem os atendimentos e as monitorias. De acordo com o Artigo 56 do Regulamento dos Cursos de Graduação do IF Goiano [11], cabe ao docente disponibilizar e divulgar o horário de atendimento destinado aos estudantes. Já o Artigo 157 do Regulamento dos Cursos de Graduação do IF Goiano [11] define que o campus deverá promover programas de atendimento ao discente.

E que segundo o Artigo 8 da Resolução 029/2016 [10] o atendimento de aluno é considerado uma atividade de apoio ao ensino, uma ação realizada pelo docente para melhorar o ensino. A realização dessa atendimento extra classe pode de fato facilitar a vida acadêmica e aumentar o rendimento escolar dos discentes, e como consequência diminuir a evasão desses estudantes da universidade. Por isso, em algumas instituições é obrigatório a realização do atendimento aos alunos, temos como exemplo o Art. 9 da Resolução 029/2016 [10] a falta não justificada do docente nos locais, dias e horários previstos para ministração de atendimento a alunos implicará na aplicação das sanções estabelecidas no Art. 44 da Lei 8.112/90 [4].

De acordo com o Luis Corral[2], o desenvolvimento multiplataforma abre oportunidade para a realização de ampliação do escopo de um aplicativo, entregando a mesma experiência a um número maior de usuários em uma variedade de plataformas. Ele também afirma que o desenvolvimento em multiplataformas ocorre com o aumento do potencial de uma aplicação onde que era disponibilizados apenas para pequenos grupos de usuários específicos, com isso o temos uma base de dados onde os usuários conseguem ter um conjunto grandioso de aplicações para realizar escolha.

Segundo Paulo [3], o monitor é identificado como beneficiado, pela contribuição a sua formação geral, como profissional e como possível futuro docente. Ele também descreve os resultados das atividades de monitoria como um processo estritamente positivo, sendo identificada a contribuição pedagógica para o ensino-aprendizagem do monitor.

A monitoria pode ser vista como um instrumento de melhoria para ensino, que se utiliza de novas práticas e experiências que têm como objetivo fortalecer a articulação entre teoria e prática e a integração curricular. [6]

De acordo com Erlon [1] o ensino-aprendizagem é composto por docentes e discentes, como instrumento que propicia troca de saberes e de experiências e aprendizado mútuo entre os participantes de um cenário. Seguindo o mesmo horizonte, Victor e Luan [9] afirmaram que o exercício da atividade de monitoria gera maior interação com os alunos e que isso possibilita a ação de compartilhar e gerar novos conhecimentos com para um novos grupos de pessoas, contribuindo, também, com o aumento no domínio (por parte dos monitores) dos conteúdos abordados na disciplina.

Revisão Bibliográfica

Danilo Giacobbo[5] observou que, atualmente no Instituto Federal do Paraná não existe uma pesquisa sobre a real situação dos atendimentos, sendo omitida sua influência sobre o desempenho dos discentes, dados sobre a percepção dos mesmos sobre esses atendimentos e uma forma fácil e ágil de coletar e analisar os dados dos atendimentos realizados. Como solução, foi desenvolvido um aplicativo para resolver o problema do processo de atendimento ao aluno.

Em conformidade com Rastogi [7], o sistema de atendimento é um dos mais importantes usados em todas as organizações para acompanhar o comparecimento. Os métodos atuais de sistema de atendimento usados nas faculdades, precisa que o corpo docente forneça os detalhes do atendimento para então, ser carregado no servidor. Deste modo, foi levantada a necessidade de automatização do sistema de atendimento para reduzir o esforço de armazenamento e manutenção dos registros.

O desenvolvimento de um aplicativo de atendimento pode suprir diversas necessidades do docente, tais como registrar quais alunos estão vindo nos atendimentos, anotar ocorrências, pesquisar dados, visualizar relatórios e receber um resumo diário, semanal e mensal sobre o andamento dessa atividade de contra turno [5].

Somasundaram [8] afirmou que um software pode ser utilizado para reduzir o trabalho manual e obter mais eficiência na gestão das informações dos alunos, deixando este processo mais fácil e benéfico em todos os sentidos. Corroborando com a mesma ideia de Rastogi [7], a qual condiz que a utilização de um sistema de atendimento bem desenvolvido, pode resultar na redução do número de horas gastas na alimentação dos detalhes de atendimento no banco de dados do servidor.

O banco de dados foi desenvolvido a partir de um modelo lógico criado no brModelo - uma ferramenta desktop voltada para o desenvolvimento de projeto de banco de dados relacionais, incluindo as etapas conceitual, lógico e físico, com ampla utilização na área de computação em todo o Brasil - e implementado ao sistema com o SQL (*Standard Query Language*) gerado.

O diagrama de banco de dados relacional para o sistema de gerenciamento de disciplinas apresenta as seguintes tabelas e atributos:

- super_user**: login: VARCHAR, senha: VARCHAR.
- discente**: matrícula: VARCHAR, nome: VARCHAR, senha: VARCHAR, monitor: BOOLEAN, fk_curso: INTEGER, ativo: BOOLEAN.
- curso**: cod_curso: INTEGER, ativo: BOOLEAN, nome: VARCHAR.
- disciplinas_discente**: fk_disciplina: INTEGER, fk_discente: VARCHAR.
- agendados**: fk_discente: VARCHAR, fk_agendamento: INTEGER, concluido: INTEGER.
- agendamentos_monitoria**: fk_discente: VARCHAR, fk_monitoria: INTEGER, concluido: INTEGER.
- monitoria**: horario: DATE, dia: TIME, cod_monitoria: INTEGER, fk_local: INTEGER, fk_discente: VARCHAR, fk_disciplina: INTEGER, ativo: BOOLEAN.
- local**: departamento: VARCHAR, sala: VARCHAR, cod_local: INTEGER, ativo: BOOLEAN.
- agendamento**: horario: TIME, cod_agendamento: INTEGER, monitor: VARCHAR, monitoria: BOOLEAN, fk_local: INTEGER, fk_docente: VARCHAR, fk_disciplina: INTEGER, dia: DATE.
- disciplina**: id: INTEGER, periodo: INTEGER, data_de_encerramento: DATE, ativo: BOOLEAN, cod_disciplina: INTEGER, fk_docente: VARCHAR, nome: VARCHAR.
- docente**: nome: VARCHAR, senha: VARCHAR, matricula: VARCHAR, ativo: BOOLEAN.

As relações entre as tabelas são definidas pelas seguintes cardinalidades:

- super_user** para **discente**: (0,1) para (0,1).
- discente** para **curso**: (1,1) para (1,1).
- discente** para **disciplinas_discente**: (0,1) para (0,n).
- disciplinas_discente** para **disciplina**: (1,n) para (1,1).
- disciplina** para **agendados**: (1,1) para (0,n).
- agendados** para **agendamentos_monitoria**: (0,n) para (0,n).
- agendamentos_monitoria** para **monitoria**: (0,1) para (0,n).
- monitoria** para **local**: (1,1) para (0,n).
- local** para **agendamento**: (1,1) para (0,n).
- agendamento** para **agendados**: (0,1) para (0,n).
- agendamento** para **disciplina**: (1,1) para (1,n).
- agendamento** para **docente**: (1,1) para (1,1).

O SQL criado a partir do modelo lógico apresentado acima, foi implementado usando inicialmente o MySQL Workbench. Durante o desenvolvimento do software foi utilizado o phpMyAdmin juntamente com o Docker para hospedar o banco de dados em um servidor local. Com o software e o banco de dados hospedados, facilitou os teste e acelerou o desenvolvimento da aplicação.

O banco de dados utilizado foi pensado para que se tornasse simples sua manutenção e o seu manuseio. Para que isso fosse possível, houve diversas alterações desde o primeiro protótipo do banco de dados. Por fim, foi construído um SGBD (Sistema Gerenciador de Banco de Dados) capaz de satisfazer todas as requisições enviadas pelo aplicativo.

3.2 SQL de Geração do Banco

```
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
```

```
START TRANSACTION;
```

```
SET time_zone = "+00:00";
```

```
CREATE TABLE `agendados` (  
  `fk_discente` varchar(250) DEFAULT NULL,  
  `fk_agendamento` int DEFAULT NULL,  
  `concluido` int NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

```
CREATE TABLE `agendados_monitoria` (  
  `fk_discente` varchar(250) DEFAULT NULL,  
  `fk_monitoria` int DEFAULT NULL,  
  `concluido` int NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

```
CREATE TABLE `agendamento` (  
  `horario` time DEFAULT NULL,  
  `dia` int DEFAULT NULL,  
  `cod_agendamento` int NOT NULL,  
  `fk_local` int DEFAULT NULL,  
  `fk_docente` varchar(250) DEFAULT NULL,  
  `fk_disciplina` int DEFAULT NULL,  
  `ativo` tinyint(1) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

```
CREATE TABLE `curso` (  
  `cod_curso` int NOT NULL,  
  `ativo` tinyint(1) DEFAULT NULL,  
  `nome` varchar(250) DEFAULT NULL
```

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

```
CREATE TABLE `discente` (  
  `matricula` varchar(250) NOT NULL,  
  `nome` varchar(250) DEFAULT NULL,  
  `senha` varchar(250) DEFAULT NULL,  
  `monitor` tinyint(1) DEFAULT NULL,  
  `fk_curso` int DEFAULT NULL,  
  `ativo` tinyint(1) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

```
CREATE TABLE `disciplina` (  
  `id` int NOT NULL,  
  `nome` varchar(250) DEFAULT NULL,  
  `periodo` int DEFAULT NULL,  
  `data_de_encerramento` date DEFAULT NULL,  
  `ativo` tinyint(1) DEFAULT NULL,  
  `cod_disciplina` int DEFAULT NULL,  
  `monitor` varchar(250) DEFAULT NULL,  
  `fk_docente` varchar(250) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

```
CREATE TABLE `disciplinas_discente` (  
  `fk_disciplina` int DEFAULT NULL,  
  `fk_discente` varchar(250) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

```
CREATE TABLE `docente` (  
  `nome` varchar(250) DEFAULT NULL,  
  `senha` varchar(250) DEFAULT NULL,  
  `matricula` varchar(250) NOT NULL,  
  `ativo` tinyint(1) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

```
CREATE TABLE `local` (  
  `departamento` varchar(250) DEFAULT NULL,  
  `sala` varchar(250) DEFAULT NULL,  
  `ativo` tinyint(1) DEFAULT NULL,
```



```
    `cod_local` int NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

```
CREATE TABLE `monitoria` (  
    `horario` time DEFAULT NULL,  
    `dia` int DEFAULT NULL,  
    `cod_monitoria` int NOT NULL,  
    `fk_local` int DEFAULT NULL,  
    `fk_discente` varchar(250) DEFAULT NULL,  
    `fk_disciplina` int DEFAULT NULL,  
    `ativo` tinyint(1) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

```
CREATE TABLE `super_user` (  
    `login` varchar(250) NOT NULL,  
    `senha` varchar(250) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

```
ALTER TABLE `agendados`  
    ADD KEY `FK_agendados_1` (`fk_discente`),  
    ADD KEY `FK_agendados_2` (`fk_agendamento`);
```

```
ALTER TABLE `agendados_monitoria`  
    ADD KEY `FK_agendados_monitoria_1` (`fk_discente`),  
    ADD KEY `FK_agendados_monitoria_2` (`fk_monitoria`);
```

```
ALTER TABLE `agendamento`  
    ADD PRIMARY KEY (`cod_agendamento`),  
    ADD KEY `FK_agendamento_2` (`fk_local`),  
    ADD KEY `FK_agendamento_3` (`fk_docente`),  
    ADD KEY `FK_agendamento_4` (`fk_disciplina`);
```

```
ALTER TABLE `curso`  
    ADD PRIMARY KEY (`cod_curso`);
```

```
ALTER TABLE `discente`  
    ADD PRIMARY KEY (`matricula`),  
    ADD KEY `FK_discente_2` (`fk_curso`);
```

```
ALTER TABLE `disciplina`  
  ADD PRIMARY KEY (`id`),  
  ADD KEY `FK_disciplina_2` (`fk_docente`);
```

```
ALTER TABLE `disciplinas_discente`  
  ADD KEY `FK_disciplinas_discente_1` (`fk_disciplina`),  
  ADD KEY `FK_disciplinas_discente_2` (`fk_discente`);
```

```
ALTER TABLE `docente`  
  ADD PRIMARY KEY (`matricula`);
```

```
ALTER TABLE `local`  
  ADD PRIMARY KEY (`cod_local`);
```

```
ALTER TABLE `monitoria`  
  ADD PRIMARY KEY (`cod_monitoria`),  
  ADD KEY `FK_monitoria_2` (`fk_local`),  
  ADD KEY `FK_monitoria_3` (`fk_discente`),  
  ADD KEY `FK_monitoria_4` (`fk_disciplina`);
```

```
ALTER TABLE `super_user`  
  ADD PRIMARY KEY (`login`);
```

```
ALTER TABLE `agendados`  
  ADD CONSTRAINT `FK_agendados_1` FOREIGN KEY (`fk_discente`)  
  REFERENCES `discente` (`matricula`) ON DELETE SET NULL,  
  ADD CONSTRAINT `FK_agendados_2` FOREIGN KEY (`fk_agendamento`)  
  REFERENCES `agendamento` (`cod_agendamento`) ON DELETE SET NULL;
```

```
ALTER TABLE `agendados_monitoria`  
  ADD CONSTRAINT `FK_agendados_monitoria_1` FOREIGN KEY (`fk_discente`)  
  REFERENCES `discente` (`matricula`) ON DELETE SET NULL,  
  ADD CONSTRAINT `FK_agendados_monitoria_2` FOREIGN KEY (`fk_monitoria`)  
  REFERENCES `monitoria` (`cod_monitoria`) ON DELETE SET NULL;
```

```
ALTER TABLE `agendamento`  
  ADD CONSTRAINT `FK_agendamento_2` FOREIGN KEY (`fk_local`)
```

```
REFERENCES 'local' ('cod_local') ON DELETE CASCADE,
ADD CONSTRAINT 'FK_agendamento_3' FOREIGN KEY ('fk_docente')
REFERENCES 'docente' ('matricula') ON DELETE CASCADE,
ADD CONSTRAINT 'FK_agendamento_4' FOREIGN KEY ('fk_disciplina')
REFERENCES 'disciplina' ('id') ON DELETE CASCADE;

ALTER TABLE 'discente'
ADD CONSTRAINT 'FK_discente_2' FOREIGN KEY ('fk_curso')
REFERENCES 'curso' ('cod_curso') ON DELETE RESTRICT;

ALTER TABLE 'disciplina'
ADD CONSTRAINT 'FK_disciplina_2' FOREIGN KEY ('fk_docente')
REFERENCES 'docente' ('matricula') ON DELETE RESTRICT;

ALTER TABLE 'disciplinas_discente'
ADD CONSTRAINT 'FK_disciplinas_discente_1' FOREIGN KEY ('fk_disciplina')
REFERENCES 'disciplina' ('id') ON DELETE RESTRICT,
ADD CONSTRAINT 'FK_disciplinas_discente_2' FOREIGN KEY ('fk_discente')
REFERENCES 'discente' ('matricula') ON DELETE SET NULL;

ALTER TABLE 'monitoria'
ADD CONSTRAINT 'FK_monitoria_2' FOREIGN KEY ('fk_local')
REFERENCES 'local' ('cod_local') ON DELETE CASCADE,
ADD CONSTRAINT 'FK_monitoria_3' FOREIGN KEY ('fk_discente')
REFERENCES 'discente' ('matricula') ON DELETE CASCADE,
ADD CONSTRAINT 'FK_monitoria_4' FOREIGN KEY ('fk_disciplina')
REFERENCES 'disciplina' ('id') ON DELETE CASCADE;
COMMIT;
```

3.3 Diagramas de Sequência

Diagrama de sequência (ou Diagrama de Sequência de Mensagens) é um diagrama usado em UML (Unified Modeling Language), representando a sequência de processos (mais especificamente, de mensagens passadas entre objetos) num programa de computador. Como um projeto pode ter uma grande quantidade de métodos em classes diferentes, pode ser difícil determinar a sequência global do comportamento. O diagrama de sequência representa essa informação de uma forma simples e lógica.

A partir dos objetos utilizados, foram feitos os diagramas de sequência correspondentes a cada um dos casos de uso presentes no sistema, listados abaixo:

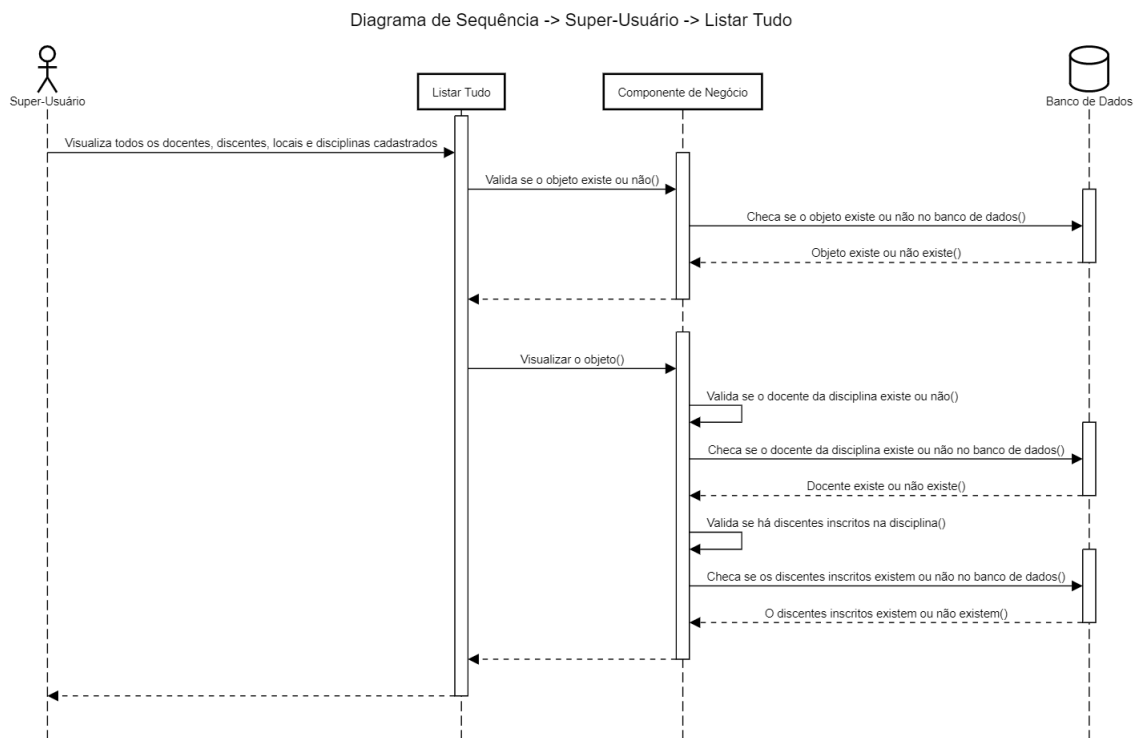


Figura 3.2: Super-Usuário – Listar Tudo

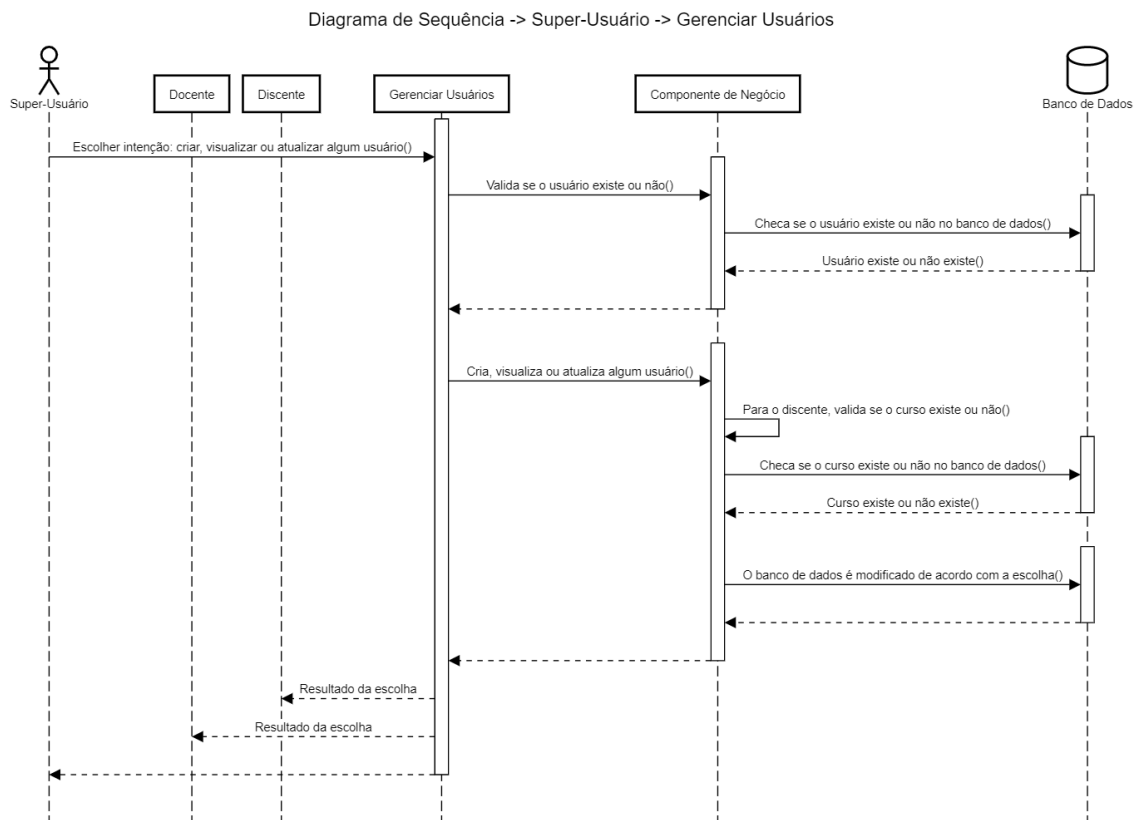


Figura 3.3: *Super-Usuário – Gerenciar Usuários*

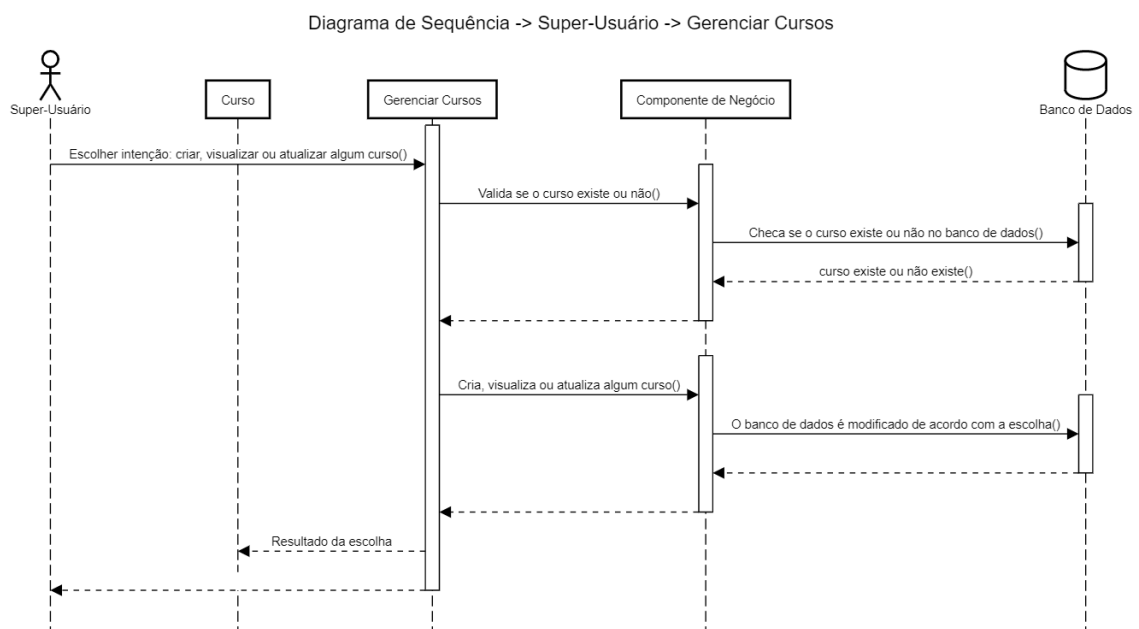


Figura 3.4: *Super-Usuário – Gerenciar Cursos*

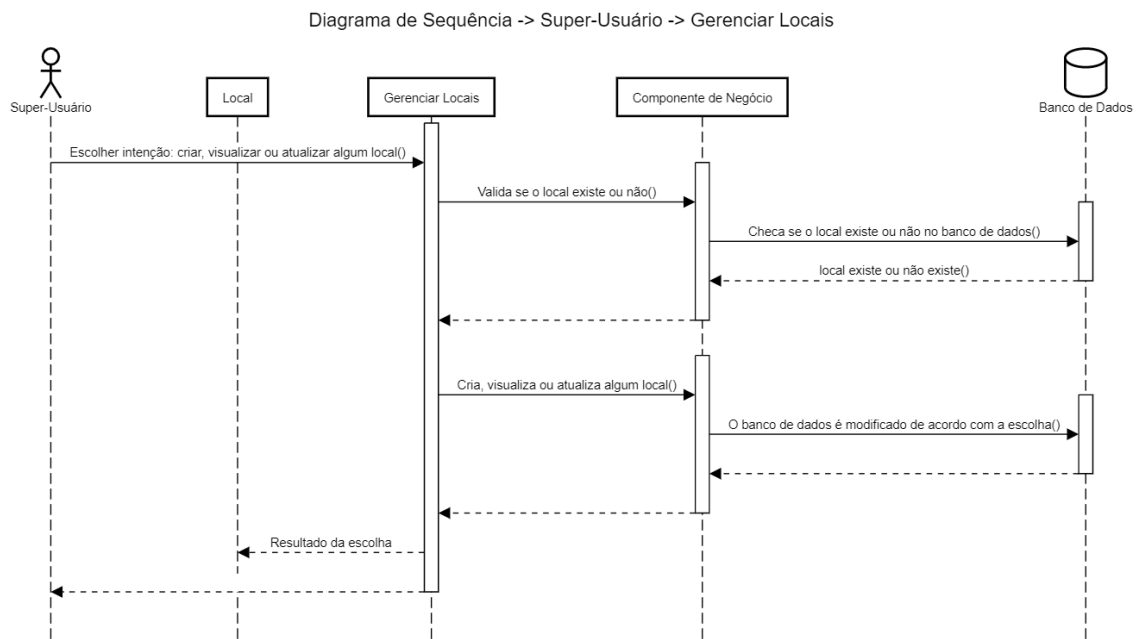


Figura 3.5: *Super-Usuário – Gerenciar Locais*

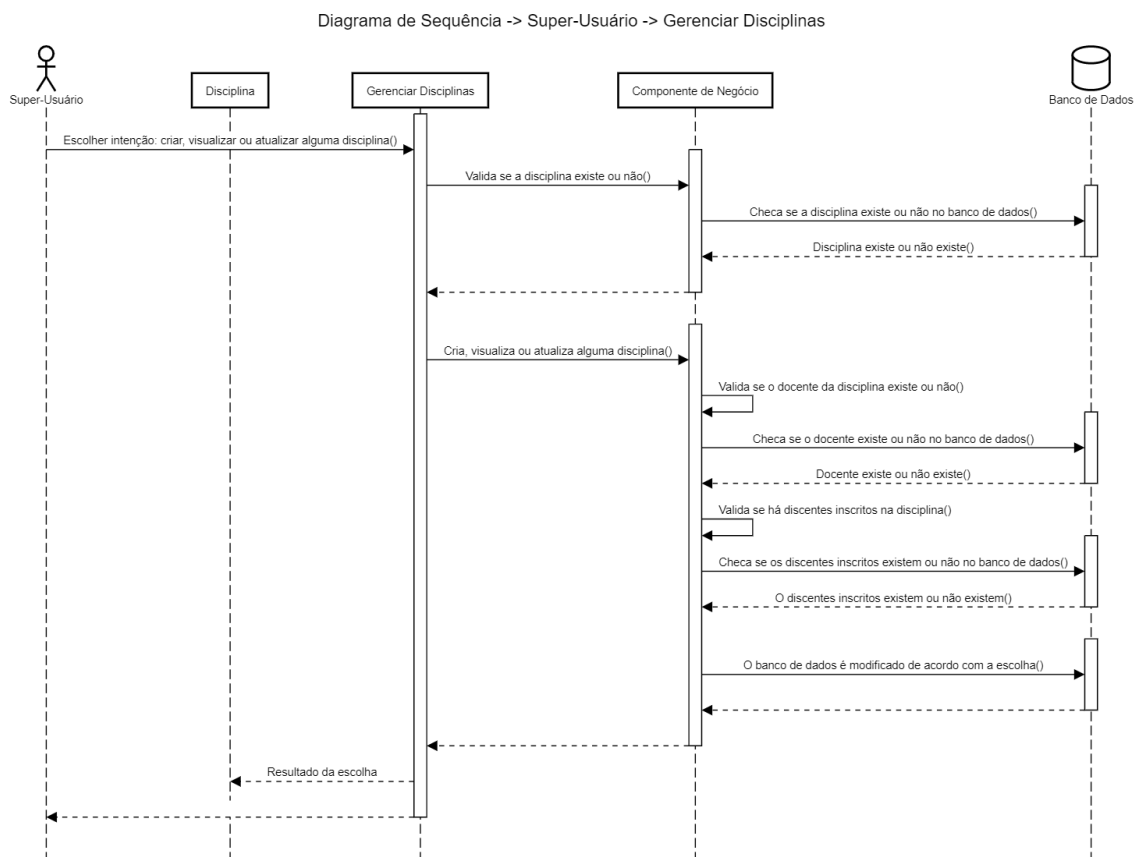


Figura 3.6: *Super-Usuário – Gerenciar Disciplinas*

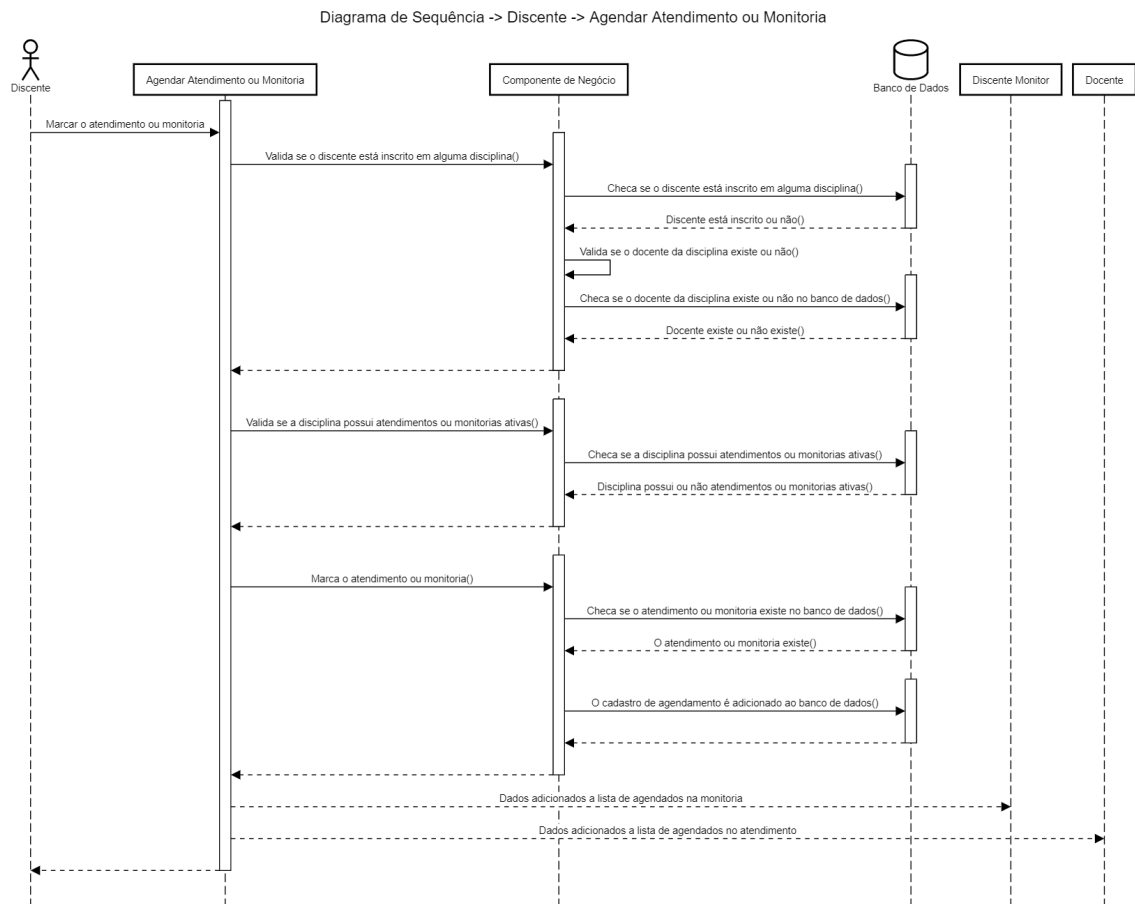


Figura 3.7: *Discente – Agendar Atendimento ou Monitoria*

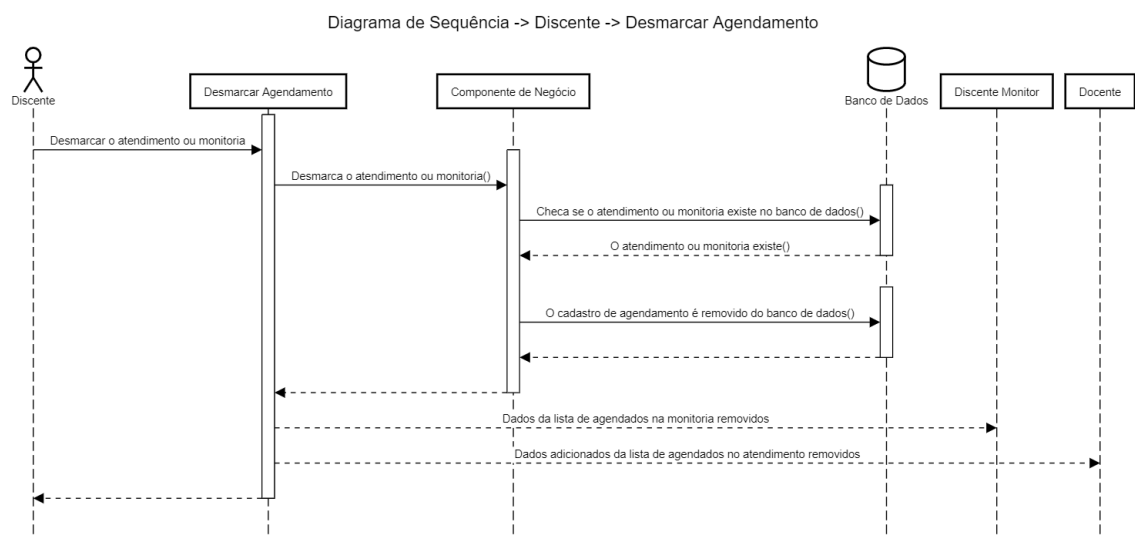


Figura 3.8: *Discente – Desmarcar Agendamento*

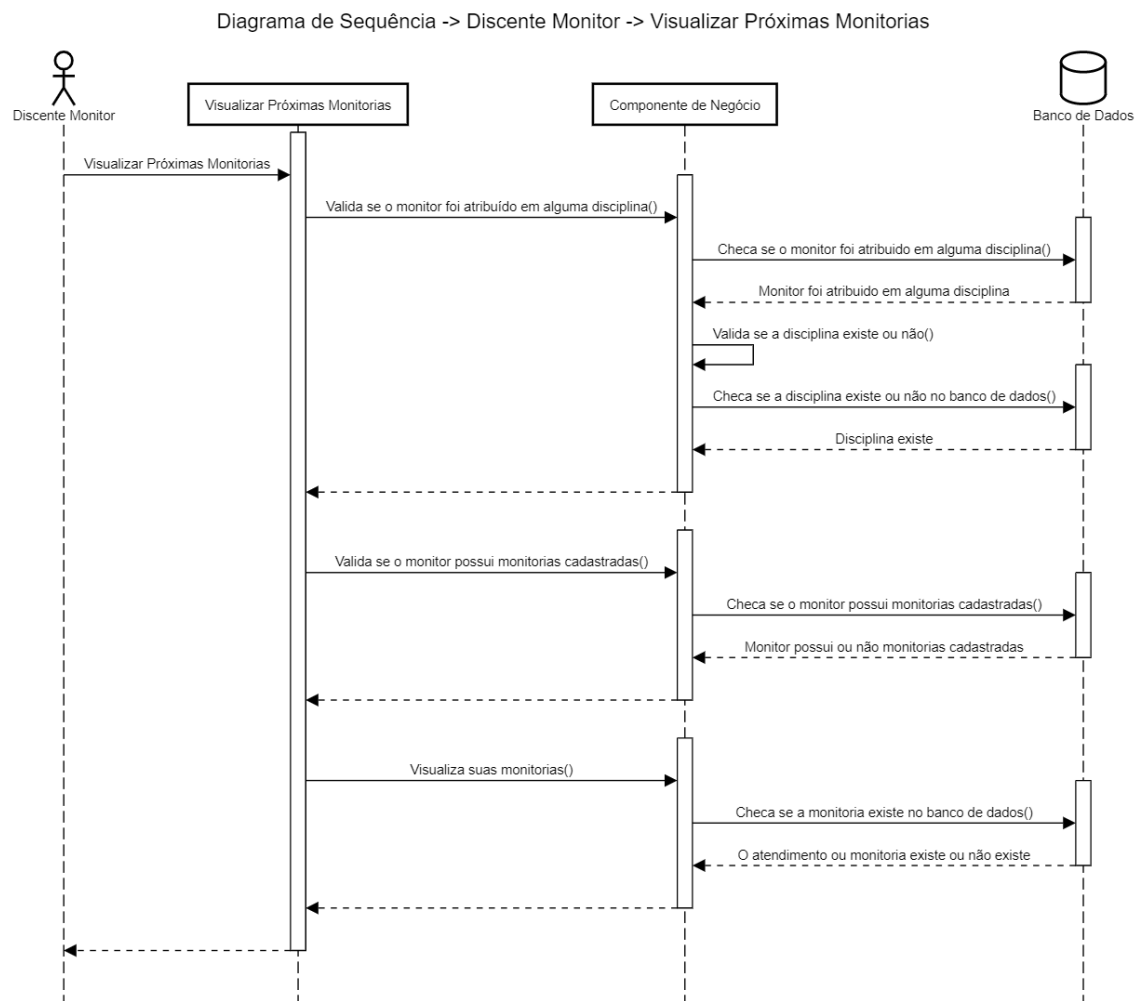


Figura 3.9: Discente Monitor – Visualizar Próximas Monitorias

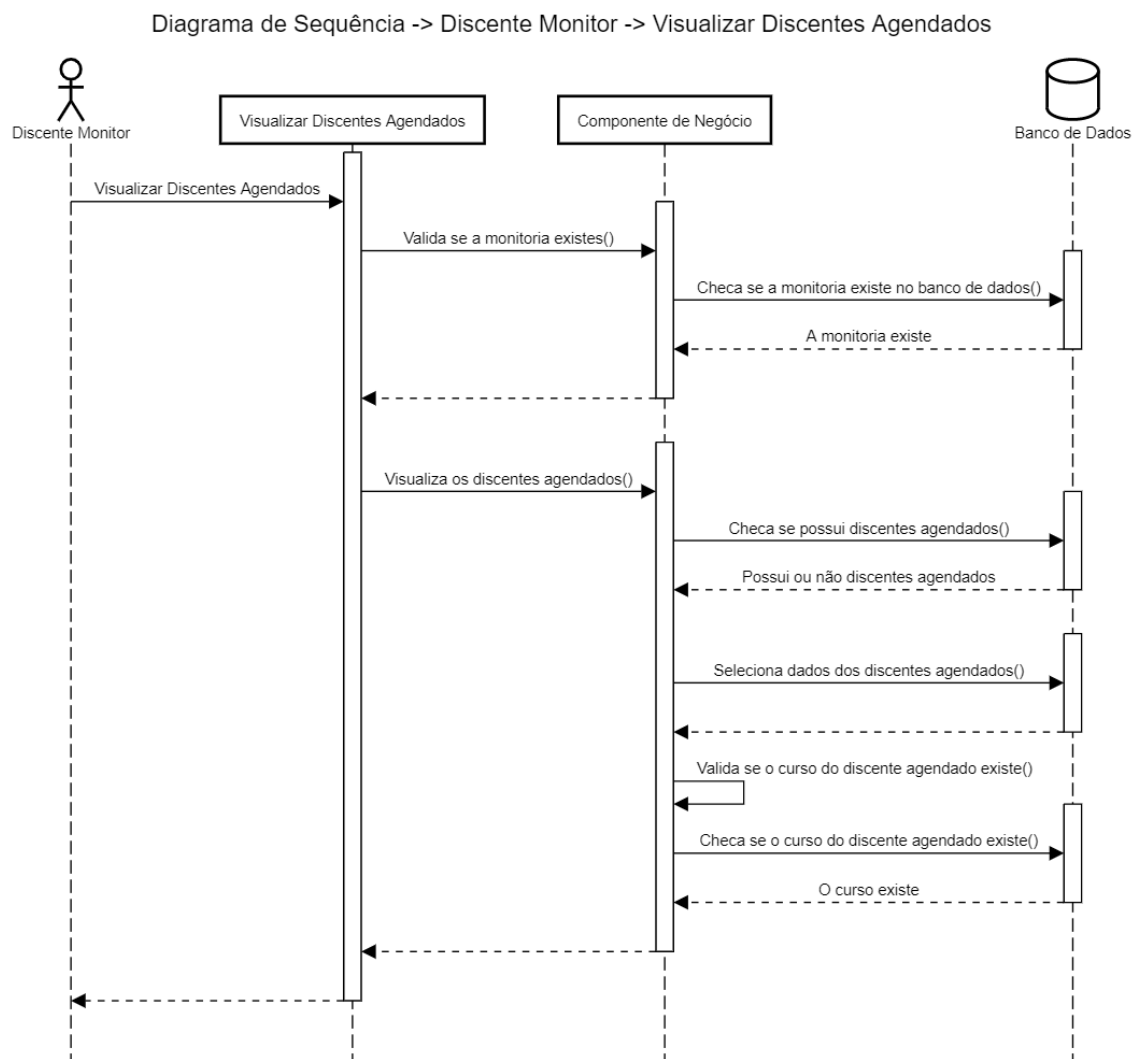
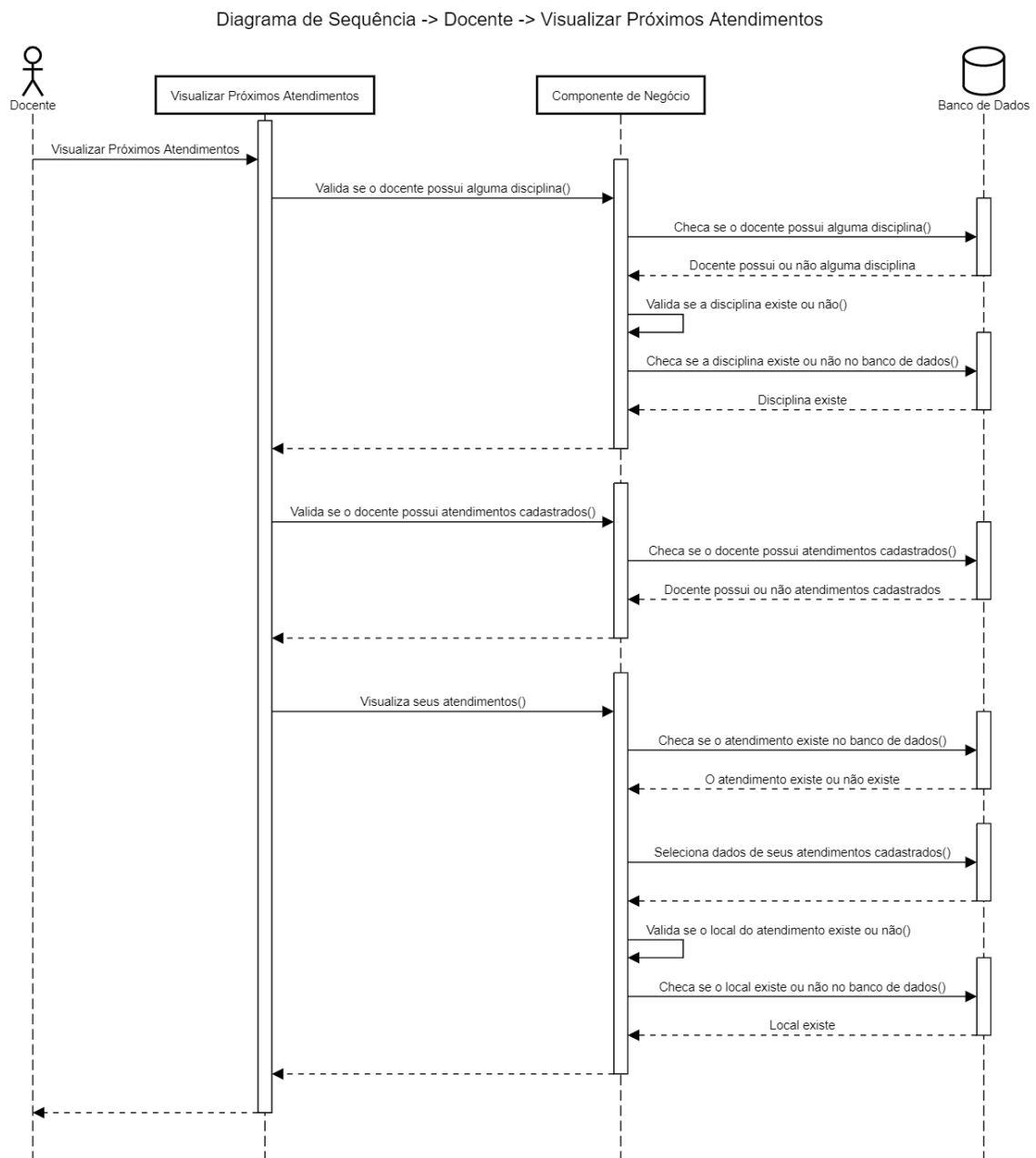


Figura 3.10: *Discente Monitor – Visualizar Discentes Agendados*

**Figura 3.11:** *Docente – Visualizar Próximos Atendimentos*

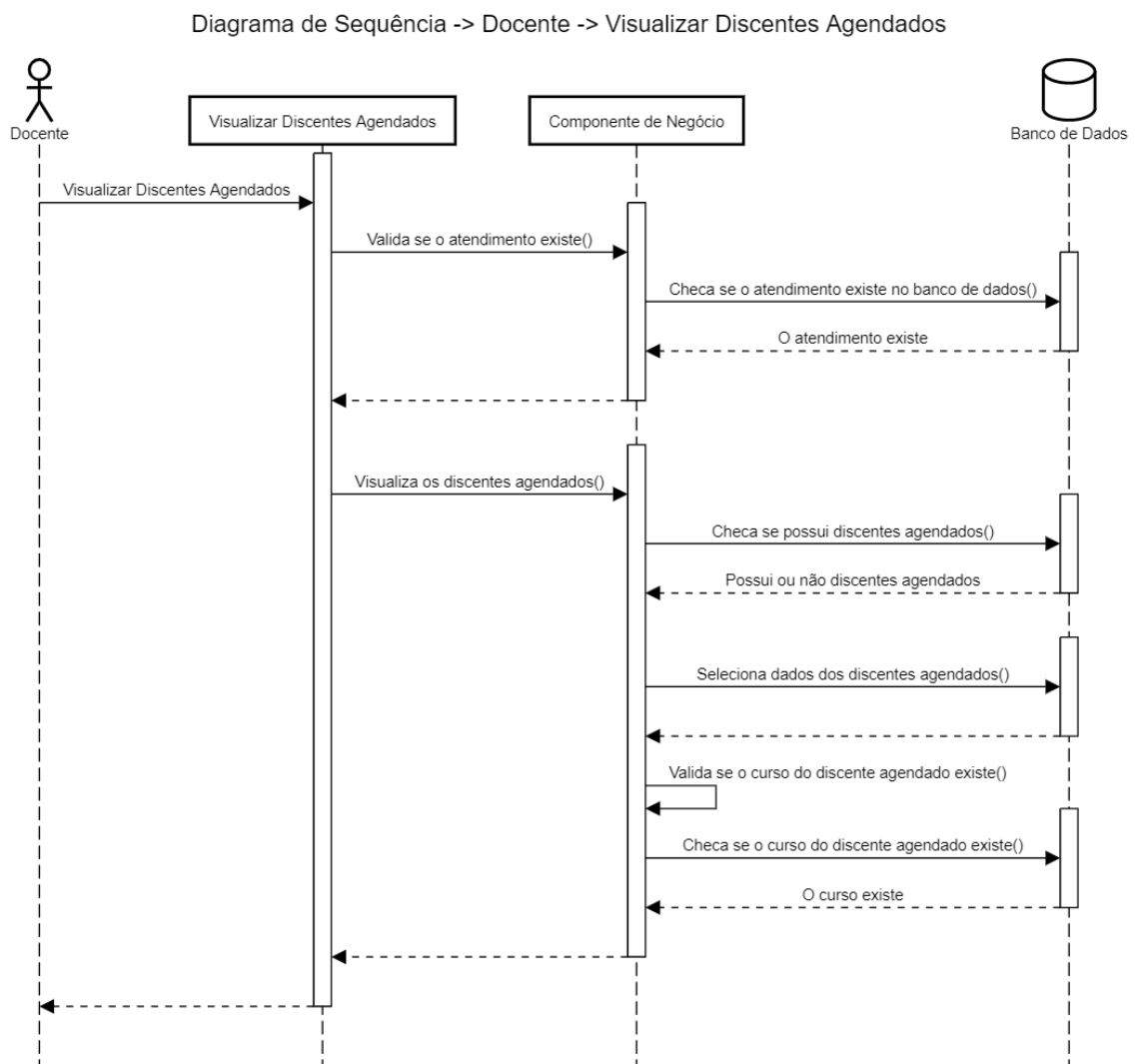
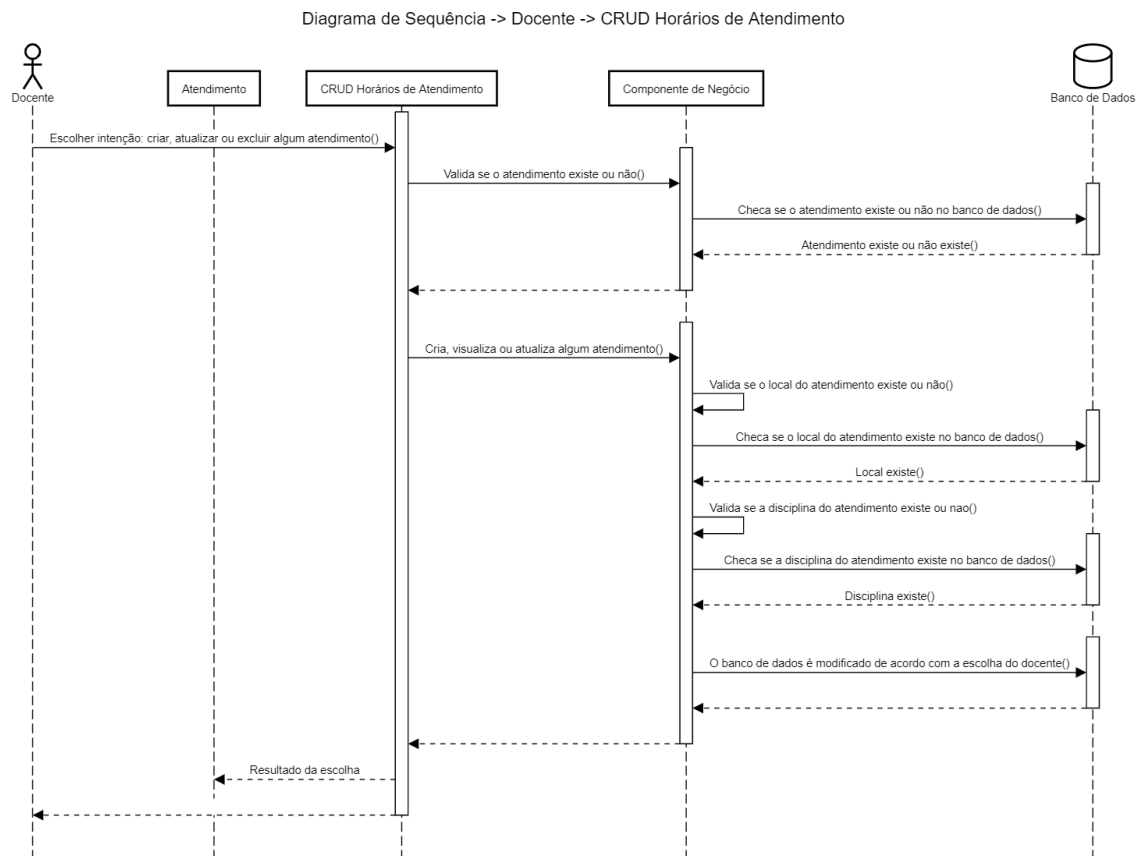


Figura 3.12: *Docente – Visualizar Discentes Agendados*

**Figura 3.13:** *Docente – CRUD Horários de Atendimento*

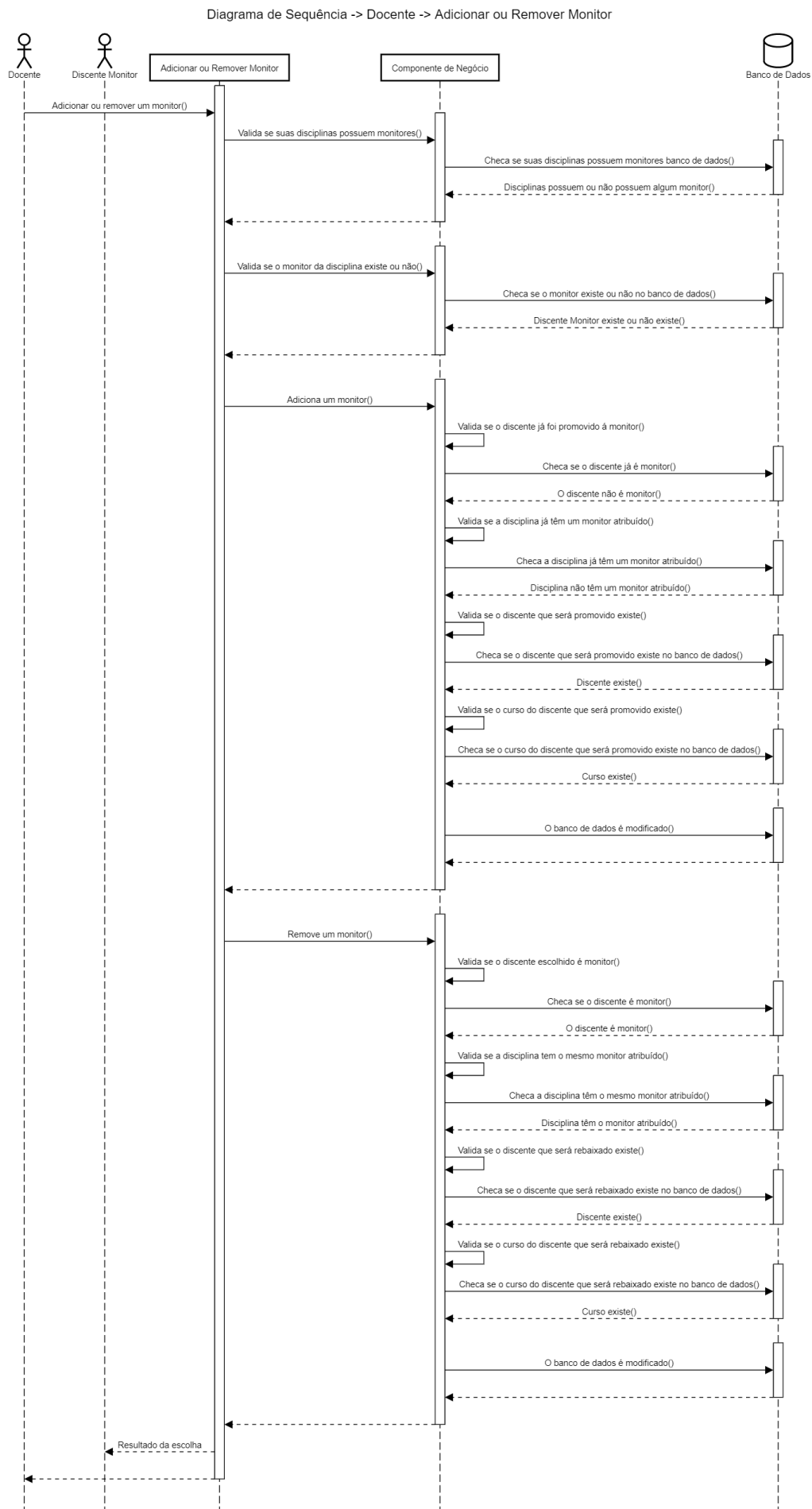


Figura 3.14: *Docente – Adicionar ou Remover Monitor*

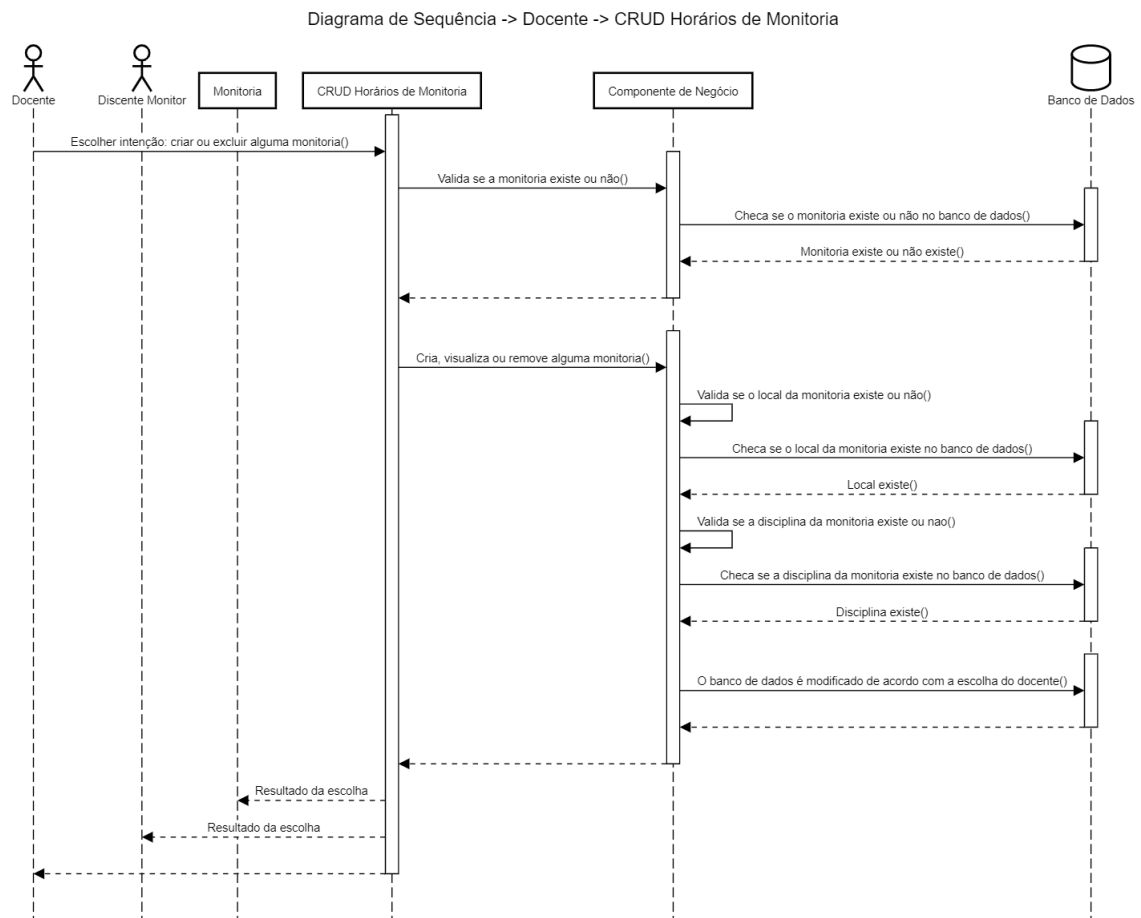


Figura 3.15: *Docente – CRUD Horários de Monitoria*

3.4 Diagrama de Classes

O diagrama de classes é uma representação da estrutura e relações das classes que servem de modelo para objetos. O diagrama de classe foi feito com todas as classes e objetos para auxiliar no desenvolvimento do software. A partir dele é possível entender todo o funcionamento do sistemas e suas funcionalidades.

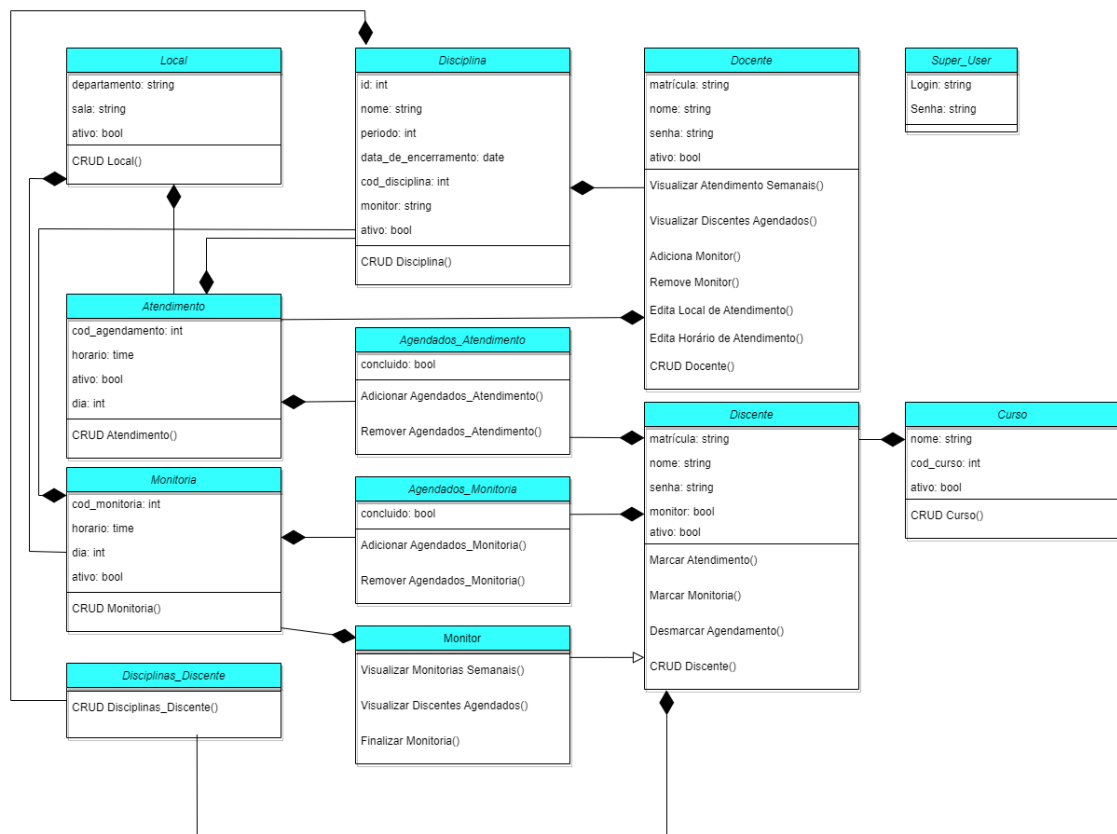


Figura 3.16: Diagrama de Classe

Método proposto

Neste capítulo vão ser apresentadas técnicas e ferramentas que foram utilizadas para o desenvolvimento do software de agendamento de atendimentos e monitorias, bem como seus requisitos, sua estruturação a partir da linguagem UML (Unified Modeling Language) e sua arquitetura. Os métodos propostos por esse trabalho surgiu da iniciativa de de implementar um software de agendamentos para auxiliar docentes e discentes em atendimentos e monitorias com sistemas e arquiteturas organizacionais, simples e padrões.

4.1 Colaboradores

Função	Pessoa	Responsabilidades
Programador	Gustavo Moura, João Saraiva e Wésio Coelho	Responsável por desenvolver o código-fonte do software.
Administrador do Banco de Dados	Gustavo Moura e Wésio Coelho	Responsável pela criação e o gerenciamento do Banco de Dados.
Analista de Testes	Gustavo Moura, João Saraiva e Wésio Coelho	Executa e relata os testes.

4.2 Ferramentas Utilizadas

Para a implementação da aplicação foram utilizados os seguintes softwares:

Visual Studio Code: Editor de código-fonte desenvolvido pela Microsoft para Windows, Linux e macOS.

Draw.io: É um software de desenho gráfico de plataforma cruzada gratuito e de código aberto desenvolvido em HTML5 e JavaScript.

Overleaf: É um editor LaTeX colaborativo baseado em nuvem usado para escrever, editar e publicar documentos científicos.

MySQL Workbench: É uma ferramenta visual unificada para arquitetos de banco de dados, desenvolvedores e DBAs, o qual fornece modelagem de dados, desenvolvimento de SQL e ferramentas de administração abrangentes para configuração de servidor, administração de usuários, backup e muito mais, disponível em Windows, Linux e Mac OS X

PhpMyAdmin: É um aplicativo web livre e de código aberto desenvolvido em PHP para administração do MySQL pela Internet.

Docker: É um conjunto de produtos de plataforma como serviço que usam virtualização de nível de sistema operacional para entregar software em pacotes chamados contêineres.

Xampp: É um pacote com os principais servidores de código aberto do mercado, incluindo FTP, banco de dados MySQL e Apache com suporte as linguagens PHP e Perl.

Google Drive: É um serviço de armazenamento e sincronização de arquivos.

GitHub: É uma plataforma de hospedagem de código-fonte e arquivos com controle de versão usando o Git.

Browser: O sistema usa o browser para ser executado e usado.

Google Docs: É um pacote de aplicativos do Google baseado em AJAX.

BrModelo: Ferramenta para modelagem de banco de dados 100% online e gratuita.

Whimsical: Espaço de trabalho visual e colaborativo que permite criar fluxogramas, diagramas e notas adesivas virtuais.

Notion: É uma aplicação que fornece componentes tais como notas, bases de dados, quadros, wikis, calendários e lembretes.

SequenceDiagram: É um software web que permite fazer criação de Diagramas de Sequência.

Overleaf: Overleaf é um editor LaTeX colaborativo baseado em nuvem usado para escrever, editar e publicar documentos científicos.

Ferramenta	Função
Projeto de Teste	Draw.io
Gerenciamento de Teste	Overleaf
Gerenciamento do Projeto	Overleaf e Draw.io
Gerenciamento de Database	MySQL e phpMyAdmin

4.3 Requisitos do Sistema

Requisitos são objetivos, propriedades, restrições que o sistema deve possuir para satisfazer contratos, padrões ou especificações de acordo com o(s) usuário(s). De

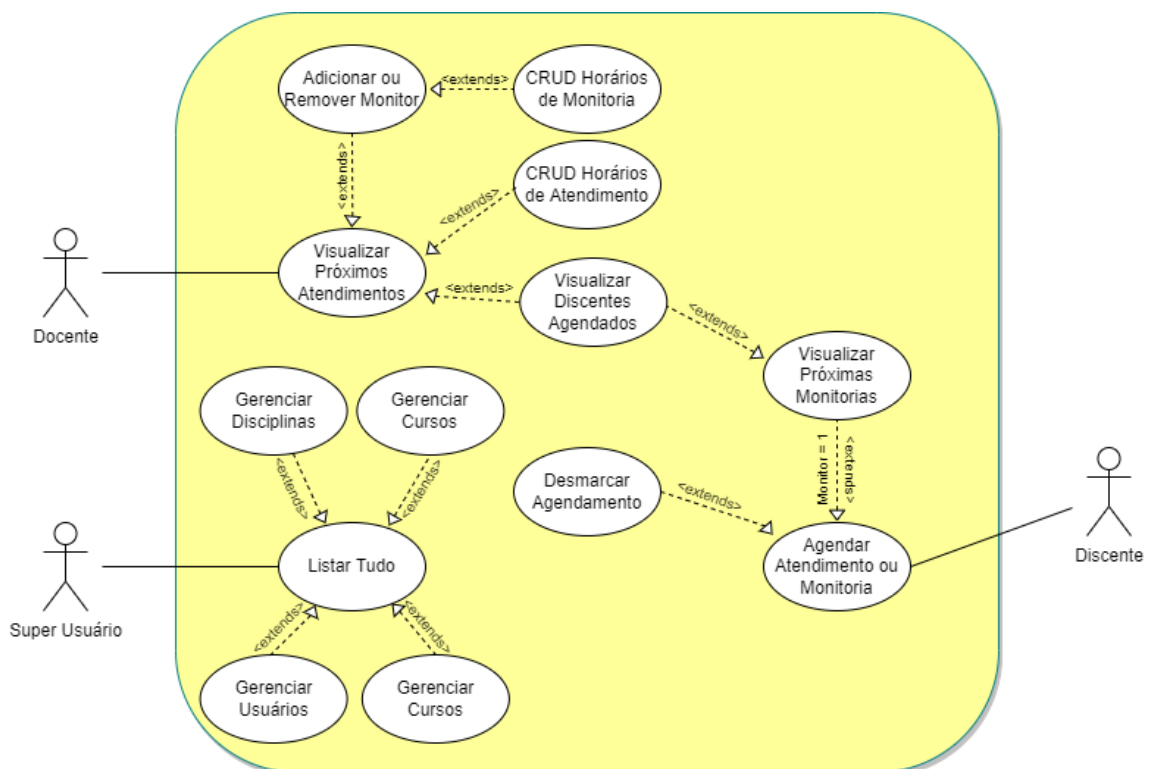
forma mais geral um requisito é uma condição necessária para satisfazer um objetivo. Diante disso, conclui-se que um requisito é um aspecto que o sistema proposto deve fazer ou uma restrição no desenvolvimento do sistema contribuindo para resolver os problemas do cliente e não o que o programador ou um arquiteto deseja. Existem dois tipos de classificação de requisitos, são eles: Requisitos Funcionais (RF) e Requisitos Não-Funcionais (RNF).

4.3.1 Requisitos Funcionais

Os requisitos funcionais referem-se sobre o que o sistema deve fazer, ou seja, suas funções e informações. Esse processo consiste em entender a natureza do software a ser construído. Além disso, o domínio do problema deve ser compreendido, bem como as funcionalidades e o comportamento esperado.

Esses requisitos seguem como base o diagrama de caso de uso. No qual é um UML (Unified Modeling Language). Uma linguagem de Modelagem Padrão, o qual é um diagrama que descreve todas as funcionalidade que foram implementadas no sistema. Esse, documenta o que irá acontecer no sistema ao ponto de vista do usuário. Descrevendo assim, as principais funcionalidades do sistema e a interação dessas funcionalidades com os usuários do mesmo sistema.

Figura 4.1: Diagrama de Casos de Uso



Os requisitos funcionais do projetos são:

- 1) REF01 – Gerenciar Cursos;
- 2) REF02 – Gerenciar Usuários;
- 3) REF03 – Gerenciar Locais;
- 4) REF04 – Gerenciar Disciplinas;
- 5) REF05 – Visualizar Próximos Atendimentos;
- 6) REF06 – CRUD Horários de Atendimento;
- 7) REF07 – Adicionar ou Remover Monitor;
- 8) REF08 – CRUD Horários de Monitoria;
- 9) REF09 – Visualizar Próximas Monitorias;
- 10) REF10 – Agendar Atendimento ou Monitoria;
- 11) REF11 – Desmarcar Agendamento;
- 12) REF12 – Listar Tudo;
- 13) REF13 – Visualizar Discentes Agendados;

Tabela 4.1: *REF01 – Gerenciar Cursos*

REF01 – Gerenciar Cursos	Prioridade Alta
Especificação da Regra de Negócio: - A criação, visualização e atualização de cursos é feita pelo super usuário - O sistema deve verificar se o curso já está cadastrado no Banco de Dados	
Detalhes da implementação prevista: O sistema reagirá de acordo com as interações do super usuário, ou seja, criando, visualizando ou atualizando o curso	

Tabela 4.2: *REF02 – Gerenciar Usuários*

REF02 – Gerenciar Usuários	Prioridade Alta
Especificação da Regra de Negócio: - A criação, visualização e atualização de discentes e docentes é feita pelo super usuário - O sistema deve verificar se o usuário já está cadastrado no Banco de Dados - O sistema deve verificar se o curso que será associado ao discente, está cadastrado no Banco de Dados	
Detalhes da implementação prevista: O sistema reagirá de acordo com as interações do super usuário, ou seja, criando, visualizando ou atualizando o usuário	

Tabela 4.3: REF03 - Gerenciar Locais

REF03 – Gerenciar Locais	Prioridade Alta
Especificação da Regra de Negócio: - A criação, visualização e atualização de locais é feita pelo super usuário - O sistema deve verificar se o local já está cadastrado no Banco de Dados	
Detalhes da implementação prevista: O sistema reagirá de acordo com as interações do super usuário, ou seja, criando, visualizando ou atualizando o local	

Tabela 4.4: REF04 - Gerenciar Disciplinas

REF04 – Gerenciar Disciplinas	Prioridade Alta
Especificação da Regra de Negócio: - A criação, visualização e atualização de disciplinas é feita pelo super usuário - O sistema deve verificar se a disciplina já está cadastrada no Banco de Dados - O sistema deve verificar se o docente, que será associado à disciplina já foi cadastrado anteriormente	
Detalhes da implementação prevista: O sistema reagirá de acordo com as interações do super usuário, ou seja, criando, visualizando ou atualizando a disciplina	

Tabela 4.5: REF05 - Gerenciar Próximos Atendimentos Semanais

REF05 – Gerenciar Próximos Atendimentos Semanais	Prioridade Alta
Especificação da Regra de Negócio: - A visualização dos próximos atendimentos é feita pela docente - O discente poderá visualizar os atendimentos ativos - O docente é responsável por ativar e desativar os seus atendimentos - O docente é responsável por finalizar o atendimento após cumprir com o horário - O sistema deve retornar todos os campos atribuídos ao atendimento e um link que redireciona-o para os alunos que estão atualmente agendados naquele atendimento - O sistema é responsável por atualizar a tabela de atendimentos disponíveis (exibida aos discentes) após a ativação ou desativação	
Detalhes da implementação prevista: O sistema listará os próximos atendimentos do docente, assim como suas respectivas informações, os alunos inscritos, e opções de edição de cada atendimento.	

Tabela 4.6: REF06 - CRUD Horários de Atendimento

REF06 – CRUD Horários de Atendimento	Prioridade Alta
Especificação da Regra de Negócio: - O docente é responsável por remover, adicionar e atualizar o atendimento - O sistema deve verificar se o atendimento já foi feito cadastrado anteriormente no Banco de Dados	
Detalhes da implementação prevista: O sistema atualizará a tabela de atendimentos após um cadastro ou remoção de um atendimento	

Tabela 4.7: REF07 -Adicionar ou Remover Monitor

REF07 – Adicionar ou Remover Monitor	Prioridade Média
Especificação da Regra de Negócio: - O docente promove um discente à monitor - O sistema deve verificar a existência do discente antes da promoção à monitor - O sistema deve verificar se o discente já foi promovido anteriormente à monitor	
Detalhes da implementação prevista: O sistema fornecerá ao docente um formulário para promover ou rebaixar um discente à monitor.	

Tabela 4.8: REF08 - Crud Horários de Monitorias

REF08 – CRUD Horários de Monitoria	Prioridade Média
Especificação da Regra de Negócio: - O docente por gerenciar os horários de monitoria do seus monitores - O sistema deve atualizar sempre que houver uma mudança de estado - O horário de monitoria deve ser visualizado pelo monitor responsável	
Detalhes da implementação prevista: O sistema reagirá de acordo com as interações do docente, ou seja, criando, visualizando ou removendo a monitoria.	

Tabela 4.9: REF09 - Visualizar Próximas Monitorias

REF09 – Visualizar Próximas Monitorias	Prioridade Média
Especificação da Regra de Negócio: - A visualização das próximas monitorias é feita pelo discente que é um monitor - O monitor é responsável por finalizar a monitoria após cumprir com seu horário - O sistema deve verificar se existe horários de monitoria no Banco de Dados	
Detalhes da implementação prevista: O sistema listará as próximas monitorias do monitor, assim como suas respectivas informações e os alunos inscritos.	

Tabela 4.10: REF10 - Marcar Atendimento ou Monitoria

REF10 – Marcar Atendimento ou Monitoria	Prioridade Alta
Especificação da Regra de Negócio: - O discente só visualiza os atendimentos e monitorias ativos - O discente tem a opção de marcar um atendimento e monitoria - O sistema deve verificar se há algum atendimento ou horário de monitoria relacionado com as disciplinas do docente cadastrado no Banco de Dados - O sistema deve atualizar a tabela de discentes agendados que será visualizado pelo docente	
Detalhes da implementação prevista: O sistema listará os atendimentos e monitorias disponíveis para o discente, assim como suas respectivas informações e uma opção de agendar para cada.	

Tabela 4.11: REF11 - Desmarcar Agendamento

REF11 – Desmarcar Agendamento	Prioridade Baixa
Especificação da Regra de Negócio: - O discente só visualiza os atendimentos e monitorias ativos - O discente tem a opção de desmarcar um atendimento e monitoria - O sistema deve verificar se há algum atendimento ou horário de monitoria relacionado com as disciplinas do docente cadastrado no Banco de Dados - O sistema deve retornar a opção de desmarcar se o atendimento ou monitoria foi marcado anteriormente. - O sistema deve atualizar a tabela de discentes agendados que será visualizado pelo docente	
Detalhes da implementação prevista: O sistema removerá o registro do agendamento do discente do atendimento/monitoria.	

Tabela 4.12: REF12 - Listar Tudo

REF12 – Listar Tudo	Prioridade Baixa
Especificação da Regra de Negócio: - A visualização da listagem de usuários, cursos, disciplinas e locais cadastrados no banco de dados é feita pelo super usuário - O sistema deve verificar se ao menos cada uma dessas classes foram cadastradas no Banco de Dados	
Detalhes da implementação prevista: O sistema listará os usuários, cursos, disciplinas e locais cadastrados no banco de dados.	

Tabela 4.13: REF13 - Visualizar Discentes Agendados

REF13 – Visualizar Discentes Agendados	Prioridade Baixa
Especificação da Regra de Negócio: - A visualização de discentes agendados é feita pelo docente e pelo discente que foi promovido à monitor. - A tabela é sempre atualizada quando um discente agenda uma monitoria ou atendimento - O sistema deve verificar se possui algum discente agendado naquele atendimento ou monitoria e no Banco de Dados e retornar uma tela com os dados dos discentes.	
Detalhes da implementação prevista: O sistema listará todos os discentes do atendimento/monitoria.	

Os requisitos são categorizados com níveis de prioridades, a saber:

Alta: A não implementação dos requisito deste tipo, torna o software insuficiente para atender as necessidades dos usuários.

Média: Requisitos importantes. Mas, não afeta as funcionalidades e os objetivos principais do software.

Baixa: Requisito não tão importante, e de utilização menos frequente pelo usuário.

4.3.2 Requisitos Não Funcionais

Os requisitos não funcionais referem-se aos critérios que qualificam os requisitos funcionais. A partir disso, pode-se concluir que esse tipo de requisito é caracterizado pela restrição das funções que o sistema deve apresentar, tais como restrições de tempo, de uso de recursos entre outras. Os requisitos não funcionais podem ser classificados de diferentes maneiras, tais como requisitos de desempenho, requisitos de portabilidade, requisitos legais, requisitos de conformidade etc. (Sommerville 2003). Os requisitos não funcionais do projetos são:

Tabela 4.14: *Usabilidade*

4.1 Usabilidade	Prioridade Alta
Descrição: - O sistema deve ter uma interface simples e de fácil navegação	
As telas serão arquitetadas para serem simples, intuitiva e direta. Será construído a interface a partir do UX Design (Experiência do Usuário).	

Tabela 4.15: *Segurança*

4.2 Segurança	Prioridade Alta
Descrição: Os dados dos clientes serão protegidos no banco de dados e as informações de contato estarão disponíveis após a confirmação da reserva.	
O software vai ser construído com base na Lei de Proteção Geral de Dados (LGPD). Terá probabilidade de indisponibilidade e uma taxa de ocorrência de falhas	

Tabela 4.16: *Desempenho*

4.3 Desempenho	Prioridade Alta
Descrição: -O sistema deve ser rápido.	
O software terá uma transação processada em 3 segundos por cada alteração de tela.	

Tabela 4.17: *Desenvolvimento*

4.4 Desenvolvimento	Prioridade Alta
Descrição: O sistema deve ser desenvolvido utilizando a linguagem PHP, XML, XSLT, HTML e o SGBD Mysql O sistema deve ser desenvolvido em no máximo 4 meses.	
O software deve ter como tecnologia para a parte de interface a linguagem PHP, XML, XSLT, HTML e webs. Para o armazenamento e tratamento de dados vão ser utilizados SGBD's, o qual é o sistema gerenciador do banco de dados, além do MySQL que criará o banco de dados para armazenamento e manipulação de dados.	

Tabela 4.18: *Manuabilidade*

4.5 Manutenibilidade	Prioridade Alta
Descrição: O sistema deve ser fácil de dar manutenção.	
O sistema deve ser de fácil atualização e alteração quando sujeito à falhas.	

Tabela 4.19: *Facilidade de Uso*

4.6 Facilidade de Uso	Prioridade Alta
Descrição: O sistema deve ser intuitivo e direto.	
As telas serão desenvolvidas para trazer uma boa dinâmica de uso. Todas as telas serão auto informativas para facilitar a navegação. Além disso, terá um botão “Help” com todas as perguntas frequentes e acesso ao suporte.	

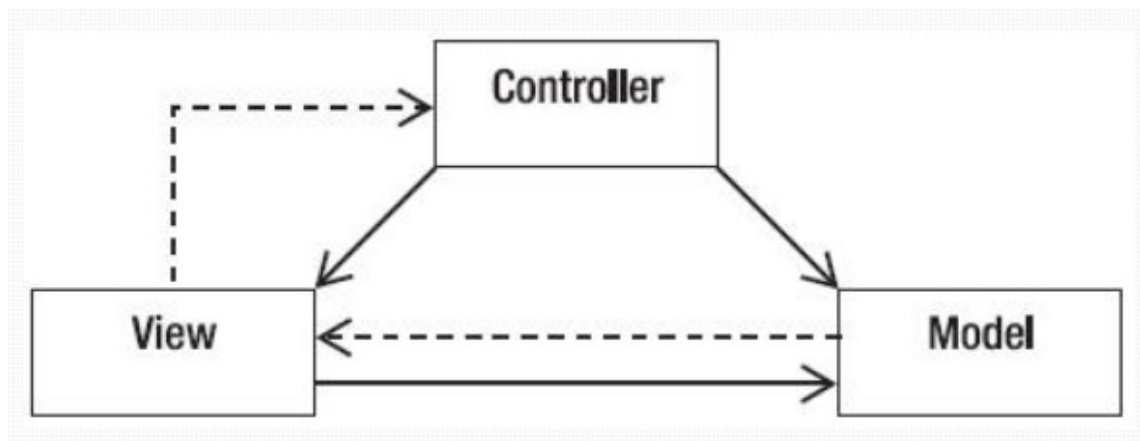
Tabela 4.20: *Robustez*

4.7 Robustez	Prioridade Alta
Descrição: O sistema deve reiniciar em 10 segundos após uma falha.	
Ao apresentar falhas na execução ou sincronização com o banco de dados, o sistema será reiniciado de forma automática.	

4.4 Arquitetura do Sistema

A arquitetura aplicada para desenvolver o software foi a MVC (Model-View-Controller), na qual é separada em três camadas, são elas: modelo, visualização e controlador. A vantagem de usar o padrão de arquitetura MVC é seu isolamento das regras de negócios da lógica de apresentação, que é a interface com o usuário. Dessa maneira possibilita uma existência de várias interfaces com o usuário que podem ser alteradas sem a necessidade de modificar as regras de negócios, proporcionando muito mais flexibilidade e oportunidades de reuso das classes. O controlador faz a interpretação das entradas do mouse ou do teclado enviadas pelo usuário e mapeia essas ações do usuário em comandos que são enviados para o modelo ou para a janela de visualização para efetuar a alteração apropriada. O modelo gerencia um ou mais elementos de dados, no qual responde a perguntas sobre o seu estado e responde a instruções para mudar de estado. Assim, o modelo sabe o que o sistema quer fazer e é a principal estrutura computacional da arquitetura, pois é ele quem modela o problema a ser resolvido. A visualização gerencia a área retangular do display, responsável por apresentar as informações para o usuário através de uma combinação de gráficos e textos. A visão não sabe nada sobre o que a aplicação está atualmente fazendo, pois tudo que ela realmente faz é receber instruções do controle e informações do modelo, para então exibi-las. A visão também se comunica de volta com o modelo e com o controlador para reportar o seu estado.

Tabela 4.21: MVC - Interações



Experimentos e análise de desempenho

Quando desejamos um produto de software de alta qualidade, deve-se assegurar que cada uma de suas partes constituintes possua alta qualidade. Visto isso, nesse capítulo será apresentado os experimentos e análises de desempenho, abordando o plano de teste e o relatório de teste, a fim de se provar com bons resultados, um software íntegro, verificado e certificado diante sua demanda.

5.1 Planos de Teste

O plano de teste visa planejar as atividades a serem realizadas para definir os métodos a serem empregados, planejar a capacidade necessária, estabelecer métricas e formas de acompanhamento do processo. Esse, funciona como um mapa para execução e controle das atividades de testes, sendo um ‘integrador’ entre diversas atividades de testes no projeto. As técnicas utilizadas na elucidação no plano de testes da aplicação, são:

1. Login
2. Cadastros
3. Atualização de Cadastros
4. Desativação e Reativação de Cadastros
5. Visualizar Próximos Atendimentos
6. Gerenciar Horários, Data e Local de Atendimento
7. Selecionar e Converter Discente à Monitor
8. CRUD Horários para a Monitoria do Monitor
9. Visualizar Próximas Monitorias
10. Agendar Atendimento ou Monitoria
11. Desmarcar Agendamento
12. Teste de Desempenho
13. Teste de Segurança

5.1.1 Login

Objetivo	Assegurar que todos os usuários estão conseguindo logar na plataforma.
Comportamento Esperado	<ol style="list-style-type: none">1. Ao digitar sua matrícula, senha e escolher o tipo de conta corretamente, o usuário será redirecionado para a página correspondente.2. Usuário deverá ser redirecionado para o <i>home</i> do usuário correspondente.
Verificações	<ol style="list-style-type: none">1. <i>Login</i> no Sistema2. Usuário Inválido.3. Campo Obrigatório.
Critérios de Conclusão	<ol style="list-style-type: none">1. Todos os três tipos de usuários devem conseguir logar no sistema.

5.1.2 Cadastros

Objetivo	Assegurar que os cadastros estão sendo registrados no Banco de Dados de forma correta
Comportamento Esperado	<ol style="list-style-type: none">1. Retornar uma mensagem após o cadastro.2. Atualizar tabela após um cadastro realizado com sucesso.
Verificações	<ol style="list-style-type: none">1. Fazer dezenas de cadastros para todas as tabela presente no banco de dados2. Conferir o banco de dados para assegurar que os dados foram preenchidos nas tabelas conforme esperado.3. Verificar se a chave primária não foi cadastrada anteriormente.
Critérios de Conclusão	<ol style="list-style-type: none">1. As informações devem permanecerem íntegras mesmo após a queda do servidor.

5.1.3 Atualização de Cadastros

Objetivo	Garantir que seja atualizado qualquer cadastro (usuários, cursos, locais e disciplinas) no banco de dados
Comportamento Esperado	<ol style="list-style-type: none">1. Atualizar tabela após um cadastro realizado com sucesso.
Verificações	<ol style="list-style-type: none">1. Fazer dezenas de atualizações de cadastros para todas as tabela presente no banco de dados2. Conferir o banco de dados para assegurar que os dados foram atualizados nas tabelas conforme esperado.3. Fazer comparação entre as tabelas do banco de dados atualizado com as tabelas do banco de dados desatualizados. .
Critérios de Conclusão	<ol style="list-style-type: none">1. As informações de cadastro quando atualizadas devem estar íntegras no banco de dados.

5.1.4 Leitura de Cadastros

Objetivo	Assegurar que os cadastros estão sendo lidos no Banco de Dados, e listados de forma correta
Comportamento Esperado	<ol style="list-style-type: none">1. Ler os cadastros no Banco de Dados.2. Listar a tabela.
Verificações	<ol style="list-style-type: none">1. Ler todos os cadastros das tabelas correspondentes presentes no Banco de Dados.2. Listar a tabela correspondente.3. Verificar se a tabela está vazia.
Critérios de Conclusão	<ol style="list-style-type: none">1. As informações devem permanecerem íntegras mesmo após a queda do servidor.

5.1.5 Desativação e Ativação de Cadastros

Objetivo	Alternação booleana do atributo "Ativo" da tabela no banco de dados
Comportamento Esperado	<ol style="list-style-type: none">1. Preencher com valor booleano de verdadeiro na tabela do banco.2. Botão para ativar e desativar.3. Quando desativado, atualiza a tabela e retornar "Ativo" em verde.4. Quando ativado, atualiza a tabela e retornar "Inativo" em vermelho.5. Atualizar tabela após uma mudança de estado.
Verificações	<ol style="list-style-type: none">1. Fazer dezenas de mudanças de estado de ativação2. Conferir o banco de dados para assegurar que os dados foram atualizados no campo de ativo
Crítérios de Conclusão	<ol style="list-style-type: none">1. As informações devem permanecerem íntegras mesmo após a queda do servidor.2. Os usuário devem receber atualizações mediante a uma leitura.

5.1.6 Visualizar Próximos Atendimentos

Objetivo	Garantir que o Docente está visualizando os atendimentos que ocorrerão durante a semana.
Comportamento Esperado	<ol style="list-style-type: none">1. Retornar uma tabela com todas as informações sobre o atendimento referente ao docente.2. Botão para ativar e desativar.3. Quando desativado, atualiza a tabela e retornar "Ativo" em verde.4. Quando ativado, atualiza a tabela e retornar "Inativo" em vermelho.5. Atualizar tabela após uma mudança de estado.
Verificações	<ol style="list-style-type: none">1. Conferir o banco de dados para assegurar que os dados foram atualizados no campo de ativo.
Critérios de Conclusão	<ol style="list-style-type: none">1. As informações da tabela devem estar listadas corretamente para cada usuário.

5.1.7 Gerenciar Horários, Data e Local de Atendimento

Objetivo	Certificar que as informações referentes ao agendamento estão sendo modificadas no Banco de Dados
Comportamento Esperado	<ol style="list-style-type: none">1. Retornar uma mensagem após o término da ação.2. Atualizar a tabela de agendamentos de acordo com as intenções do docente.
Verificações	<ol style="list-style-type: none">1. Fazer dezenas de cadastros, atualizações e eliminações de agendamentos.2. Conferir o banco de dados para assegurar que os dados foram preenchidos, editados ou excluídos da tabela conforme esperado.
Critérios de Conclusão	<ol style="list-style-type: none">1. Todos os testes executados devem ser bem sucedidos.2. As informações devem permanecerem íntegras mesmo após a queda do servidor.

5.1.8 Promover e Rebaixar Discente à Monitor

Objetivo	Certificar que o Docente consegue adicionar um monitor a partir de um Discente já cadastrado.
Comportamento Esperado	<ol style="list-style-type: none">1. Retornar uma mensagem após promover um Discente à Monitor.2. Retornar mensagem de promoção realizada com sucesso.
Verificações	<ol style="list-style-type: none">1. Fazer dezenas de promoções de Discentes à Monitor.2. Fazer dezenas de rebaixamento de Monitor à Discente.3. Conferir o banco de dados para assegurar que os dados foram alterados nas tabelas conforme esperado.4. Verificar se a matrícula do Discente promovido foi inserido na tabela de Disciplina.
Critérios de Conclusão	<ol style="list-style-type: none">1. As informações devem permanecerem íntegras mesmo após a queda do servidor.2. O discente promovido deve ter acesso às funcionalidades exclusivas de monitoria.

5.1.9 CRUD Horários para a Monitoria do Monitor

Objetivo	Assegurar que os cadastros, leituras, atualizações e eliminações dos horários dos monitores estão ocorrendo de forma correta
Comportamento Esperado	<ol style="list-style-type: none">1. Retornar uma mensagem após um cadastro, atualização ou remoção realizada com sucesso.2. Atualizar tabela após um cadastro, atualização ou remoção realizado com sucesso.
Verificações	<ol style="list-style-type: none">1. Fazer três cadastros para vários monitores.2. Editar vários cadastros.3. Deletar alguns cadastros feitos e editados.4. Conferir o banco de dados para assegurar que os dados foram preenchidos nas tabelas conforme esperado.
Critérios de Conclusão	<ol style="list-style-type: none">1. Todas as modificações no banco de dados deverão ser bem sucedidas.2. As informações devem permanecerem íntegras mesmo após a queda do servidor.

5.1.10 Visualizar Próximas Monitorias

Objetivo	Garantir que o Monitor está visualizando os atendimentos que ocorrerão durante a semana.
Comportamento Esperado	<ol style="list-style-type: none">1. Retornar uma tabela com todas as informações sobre a monitoria referente ao monitor.2. retornar botão para acessar aos Discentes agendados para determinado horário.
Verificações	<ol style="list-style-type: none">1. Verificar que se a busca feita no banco de dados retornar corretamente os dados à tabela.2. Verificar se o há um link que direciona o monitor aos à lista de Discentes agendados.
Critérios de Conclusão	<ol style="list-style-type: none">1. As informações devem permanecerem íntegras mesmo após a queda do servidor.2. O Monitor deve ver apenas informações referente a sua própria monitoria.

5.1.11 Agendar Atendimento ou Monitoria

Objetivo	Assegurar que a inscrição dos discentes nos atendimentos e monitorias estão sendo registradas no Banco de Dados de forma correta
Comportamento Esperado	<ol style="list-style-type: none">1. Retornar uma mensagem após o agendamento.2. Atualizar tabela após um agendamento realizado com sucesso.
Verificações	<ol style="list-style-type: none">1. Fazer dezenas de agendamentos para todas as disciplinas presentes no Banco de Dados2. Conferir o banco de dados para assegurar que os dados foram preenchidos nas tabelas conforme esperado.
Crítérios de Conclusão	<ol style="list-style-type: none">1. O nome do discente deverá aparecer no agendamento do docente.

5.1.12 Desmarcar Agendamento

Objetivo	Certificar que será excluído a linha referente ao agendamento na tabela do banco de dados.
Comportamento Esperado	<ol style="list-style-type: none">1. Atualizar tabela após desmarcação do atendimento.2. Atualizar a página para poder fazer novos agendamentos.
Verificações	<ol style="list-style-type: none">1. Fazer dezenas de desmarcações de agendamento no banco de dados2. Conferir o banco de dados para assegurar que os dados do agendamento foram excluídos.3. Verificar se há agendamentos desmarcados.
Critérios de Conclusão	<ol style="list-style-type: none">1. O agendamento deve ser retirado do banco de dados e da página web.

5.1.13 Teste de Desempenho

Objetivo	Certificar que a execução da aplicação seja efetuada no menor tempo possível durante sua execução.
Técnica	Fazer a verificação do tempo de execução em relação aos processos com ferramentas
Critérios de Conclusão	<ol style="list-style-type: none">1. Para a medição exata e controle estimado de desempenho, esse deverá ser testado em um período estipuladamente esforçado ou em uma máquina eficiente.

5.1.14 Teste de Segurança

Objetivo	Certificar que tal funcionalidade seja feita somente pelo usuário permitido a fazê-la.
Técnica	Uso de pentests
Critérios de Conclusão	<ol style="list-style-type: none">1. Quando não houver falhas de segurança encontradas em um certo período de tempo, a aplicação será continuada.2. Se encontradas 1 ou mais falhas de segurança durante o período de teste, será paralisada a aplicação para medidas de correções, com novos testes em seguidas.

5.2 Relatório de Teste

5.2.1 Login

Em todos os testes de login o sistema retornou exclusivamente a página do usuário correspondente quando a senha e o login corresponde a um cadastro do banco. Os usuários que entrou com senha ou matrícula errada são verificados pelo sistema, no qual confere que os dados eram incorretos. Por tanto, os três usuários conseguiram logar ao usar as credenciais corretas depois da verificação feita sistema.

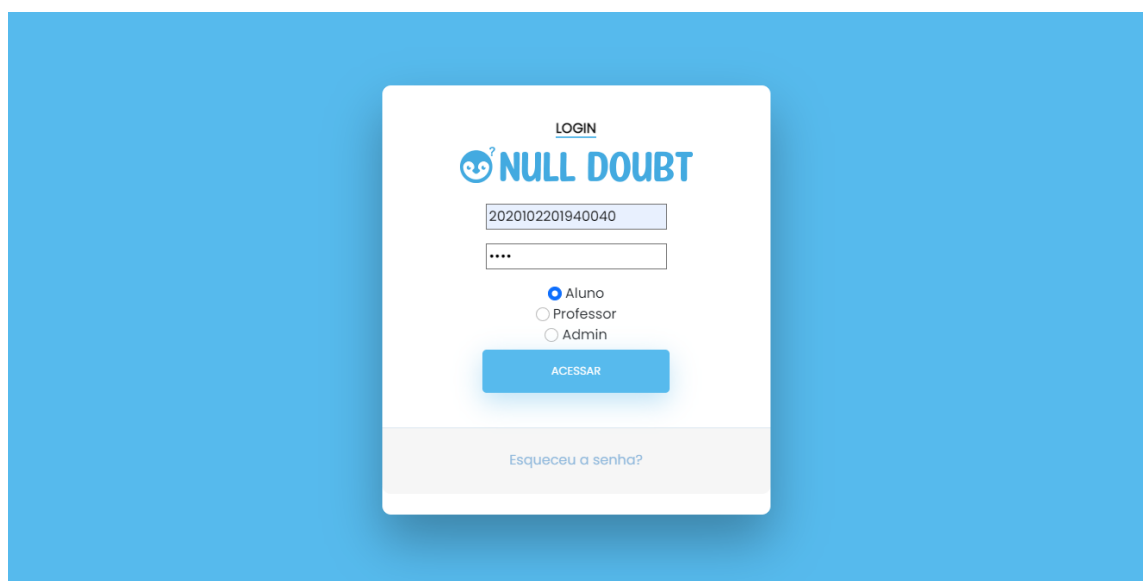


Figura 5.1: Login efetuado com sucesso

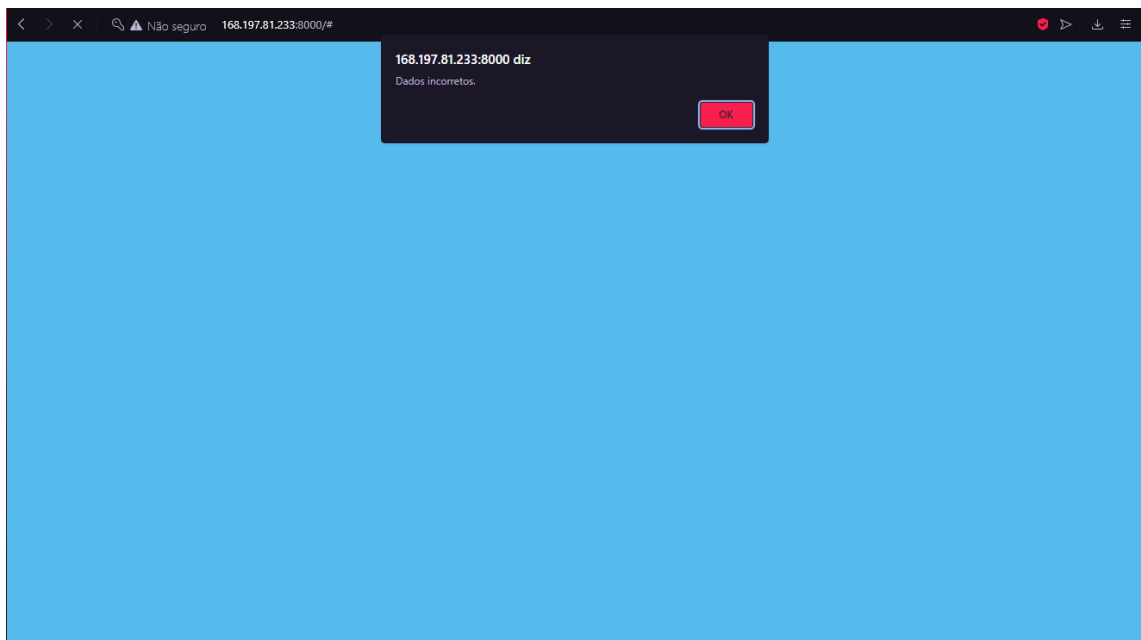


Figura 5.2: Erro de login, com dados errados

5.2.2 Cadastros

Todos os cadastros foram efetuados corretamente. Ao analisar o banco de dados dos cadastros realizados não foram encontrado anormalidades e todas as senhas se encontravam criptografadas.

O sistema ainda não consegue retornar um erro legível ao usuário quando um cadastro é duplicado. Entretanto, o banco não é alterado. Portanto, não foi totalmente aceito.

Discentes					
Matricula	Nome	Curso	Ativo	Monitor	
2020102201940202	Daniel Abreu de Carvalho	Química	Ativo	1	Desativar
2020102201940040	Gustavo Moura Barros	Ciências da Computação	Ativo	1	Desativar
2020102201940067	João Saraiva	Ciências da Computação	Ativo	0	Desativar
2020102201940326	Pedro Lima Bezerra	Ciências da Computação	Inativo	0	Ativar
<input type="text" value="2020102201940083"/>	<input type="text" value="Wésio Coelho Messias"/>	<input type="text" value="1"/>	<input type="text" value="....."/>		Adicionar

Figura 5.3: Cadastro de Discentes

matricula	nome	senha	monitor	fk_curso	ativo
2020102201940040	Gustavo Moura Barros	84a26c4612a7f9958174ee6552625282	1	1	1
2020102201940067	João Saraiva	dccd96c256bc7dd39bae41a405f25e43	0	1	1
2020102201940083	Wésio Coelho Messias	8e7c5b50738fb0ac3375faa330b96a61	0	1	1
2020102201940202	Daniel Abreu de Carvalho	aa47f8215c6f30a0dcdb2a36a9f4168e	1	4	1
2020102201940326	Pedro Lima Bezerra	c6cc8094c2dc07b700fcc36d64e2138	0	1	0

Figura 5.4: *Cadastro de Discentes*

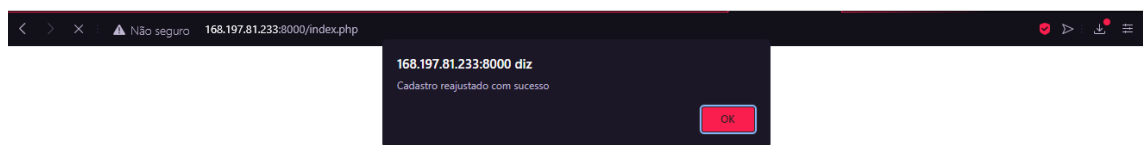


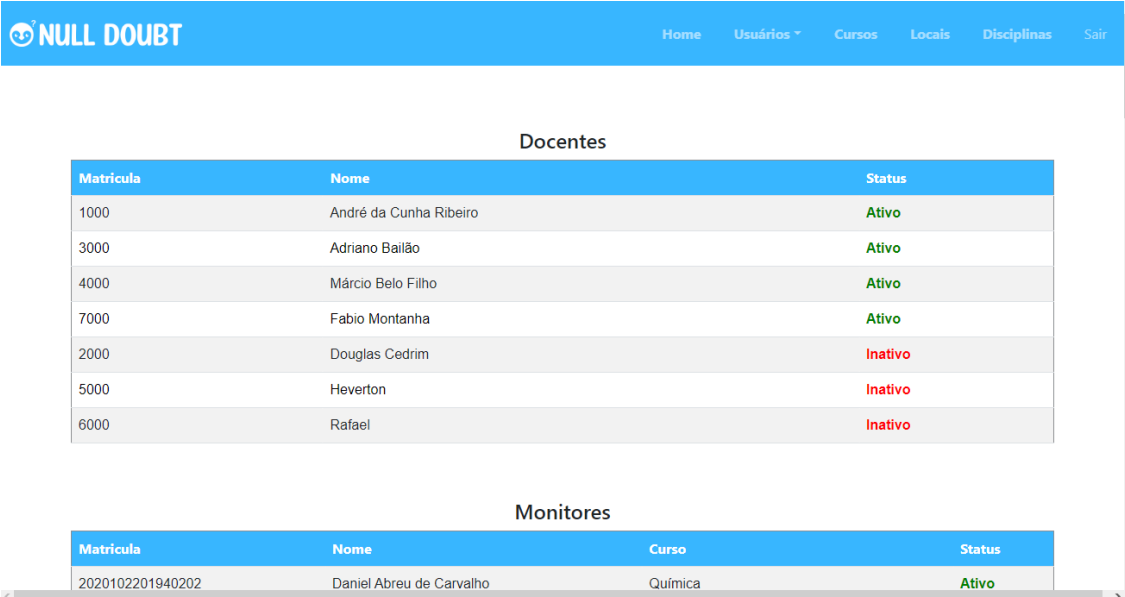
Figura 5.5: *Cadastro feito com sucesso*

5.2.3 Atualização de Cadastros

O teste de atualização de cadastros de todas as tabelas não foi executado. Pois, o requisito apresentou falhas na implementação, tendo em vista sua baixa prioridade para o sistema. Sendo assim, o teste não foi aceito.

5.2.4 Leituras de Cadastros

Os cadastros estão sendo lidos exatamente como estão armazenados no banco de dados. Todos os cadastros são listados em tabelas separadas, como no banco. O sistema também retorna uma mensagem quando a tabela está vazia. Portanto, esse teste foi aceito.



The screenshot shows a web application interface with a blue header bar containing the logo 'NULL DOUBT' and navigation links: Home, Usuários, Cursos, Locais, Disciplinas, and Sair. Below the header, there are two tables. The first table, titled 'Docentes', has three columns: 'Matricula', 'Nome', and 'Status'. It lists seven teachers with their respective matriculation numbers, names, and status (Ativo or Inativo). The second table, titled 'Monitores', has four columns: 'Matricula', 'Nome', 'Curso', and 'Status'. It lists one monitor with their matriculation number, name, course, and status (Ativo).

Matricula	Nome	Status
1000	André da Cunha Ribeiro	Ativo
3000	Adriano Bailão	Ativo
4000	Márcio Belo Filho	Ativo
7000	Fabio Montanha	Ativo
2000	Douglas Cedrim	Inativo
5000	Heverton	Inativo
6000	Rafael	Inativo

Matricula	Nome	Curso	Status
2020102201940202	Daniel Abreu de Carvalho	Química	Ativo

Figura 5.6: *Leitura de Docentes*

5.2.5 Desativação e Reativação de cadastros

Todos os cadastros foram ativados e desativados várias vezes consecutivas e as tabelas foram atualizadas com sucesso, retornando as cores esperadas durante as mudanças de estados.

5.2.6 Visualizar Próximos Atendimentos

Para este teste foi utilizado os cadastros de docentes, discentes e disciplinas para retornar os agendamentos correspondente para cada usuário. Os docentes e os discentes cadastrados só visualizaram os atendimentos que corresponderam à suas disciplinas do semestre atual. Quando um dos docentes desativou um dos seus atendimentos, o discente não conseguiu visualizar o mesmo na sua página, ou seja, as tabelas foram atualizadas com sucesso. Portanto, o teste foi aceito.



Figura 5.7: Visualização de Atendimentos Docentes

horario	dia	cod_agendamento	fk_local	fk_docente	fk_disciplina	ativo
08:00:00	3	3	24	3000	4	1
14:30:00	4	5	110	1000	2	0
12:00:00	1	6	110	1000	1	1
07:00:00	5	7	15	1000	3	1

Figura 5.8: Visualização de Atendimentos Banco de Dados



The screenshot shows the NULL DOUBT website interface. At the top is a blue navigation bar with the logo and links for Home, Atendimentos, Monitoria, and Sair. Below the navigation bar is a yellow banner with the text "CORONAVÍRUS (COVID-19)" and a subtext "clique e saiba como se prevenir". Below the banner is a section titled "Atendimentos Disponíveis" containing a table with columns: Horário, Dia, Disciplina, Local, Docente, and a button.

Horário	Dia	Disciplina	Local	Docente	
12:00	Segunda	Grafos	Lab do André	André da Cunha Ribeiro	Desmarcar
08:00	Quarta	Projeto de Software	Sala 29	Adriano Bailão	Agendar
07:00	Sexta	TCC - 1	Sala de Monitoria	André da Cunha Ribeiro	Agendar

Figura 5.9: Visualização de Atendimentos Discentes

5.2.7 Gerenciar Horários, Data e Local de Atendimento

Todos os horários foram cadastrados, lidos e removidos como esperado. Porém, a atualização do cadastro não foi possível. Sendo assim, o teste foi parcialmente aceito.



The screenshot shows the NULL DOUBT website interface. At the top is a blue navigation bar with the logo and links for Home, Atendimentos, Monitoria, and Sair. Below the navigation bar is a section titled "Atendimentos Semanais" containing a table with columns: Horário, Dia, Disciplina, Departamento, Sala, Status, and a button. Below the table is a section titled "Novo Atendimento" containing a form with fields for Horário, Dia, Disciplina, and Local, and an "Adicionar" button.

Horário	Dia	Disciplina	Departamento	Sala	Status	
12:00	Segunda	Grafos	Computação	Lab do André	Ativo	Excluir
07:00	Sexta	TCC - 1	Computação	Sala de Monitoria	Inativo	Excluir

Horário	Dia	Disciplina	Local	
16:30	Terça	Análise e Projeto de Algoritmos	Bloco III/Sala 29	Adicionar

Figura 5.10: Gerenciar Atendimentos

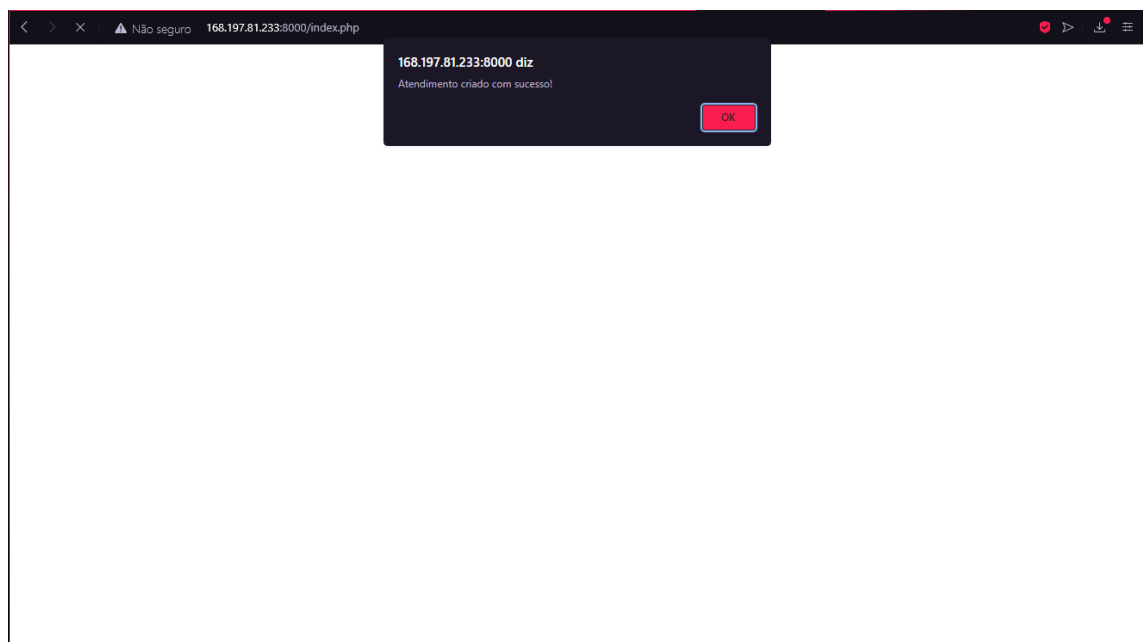


Figura 5.11: *Notificação Após Gerenciar Atendimentos*

5.2.8 Promover e Rebaixar Discente à Monitor

A promoção foi feita para vários discentes e o rebaixamento feito para vários monitores. Suas páginas foram atualizadas com as novas funcionalidades exclusivas para o monitor, em caso de promoção. Só foi possível promover ou rebaixar um discente anteriormente cadastrado. O teste retornou o esperado, então foi aceito.

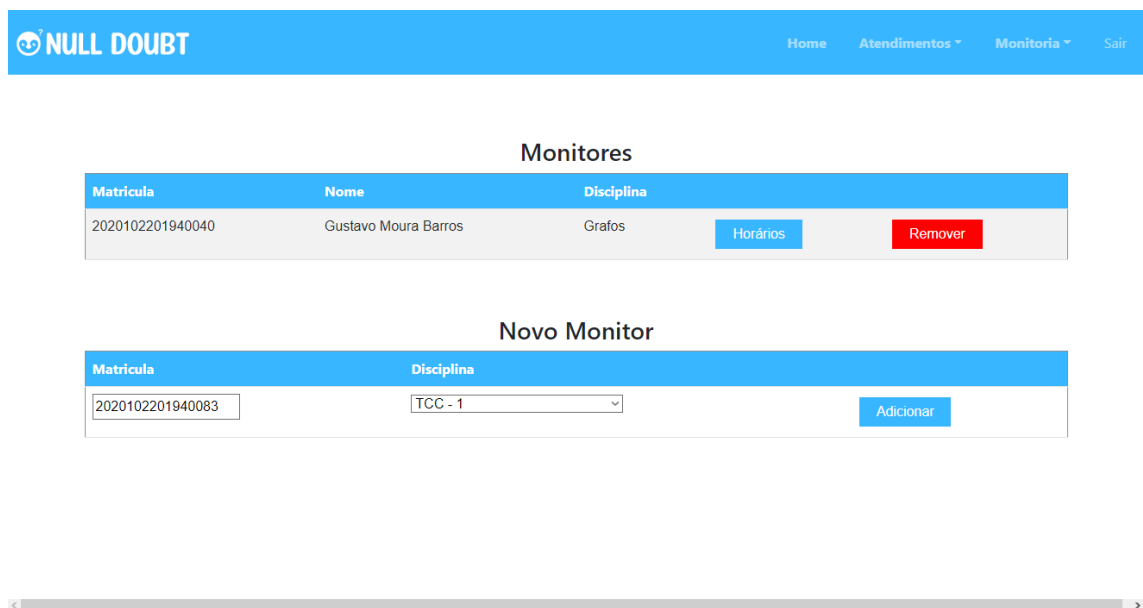


Figura 5.12: *Promover Monitor*

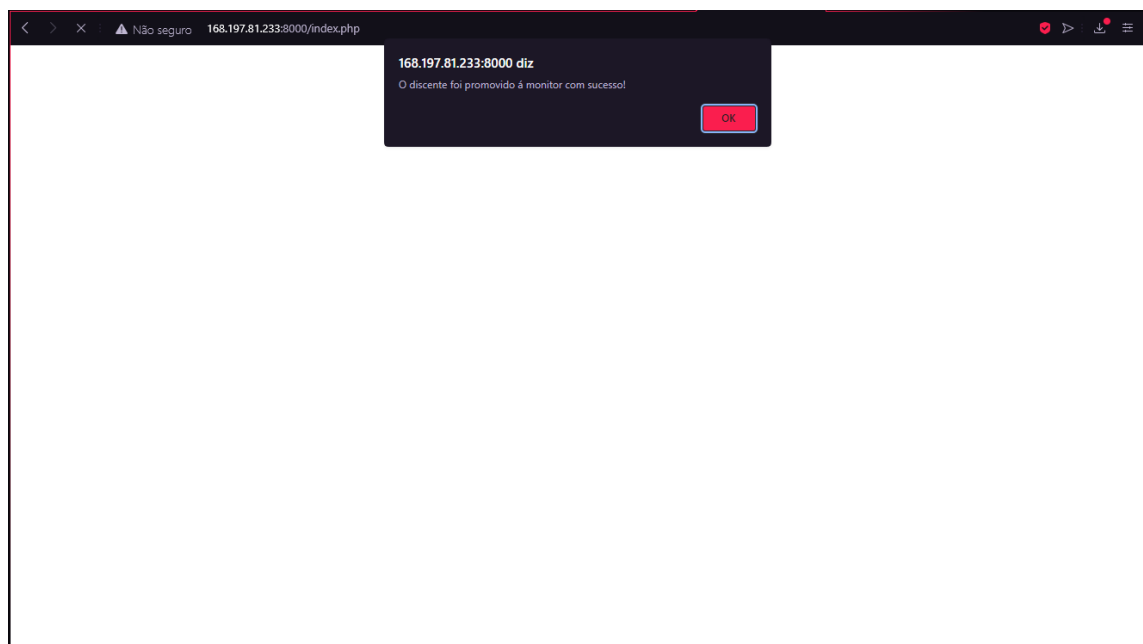


Figura 5.13: *Promover Monitor*

5.2.9 CRUD Horários para a Monitoria do Monitor

Todos os horários de monitorias foram cadastrados, lidos e removidos como esperado. Porém, a atualização do cadastro não foi possível. Sendo assim, o teste foi parcialmente aceito.

A screenshot of a web application interface. At the top is a blue header with the logo 'NULL DOUBT' and navigation links: 'Home', 'Atendimentos', 'Monitoria', and 'Sair'. Below the header, the title 'Horários de Gustavo Moura Barros' is centered. Underneath is a table with three columns: 'Horário', 'Dia', and 'Local'. The first row of the table contains the text 'Nenhum horário cadastrado'. Below the table, there is a form with three input fields: a time picker set to '08:00', a dropdown menu set to 'Terça', and another dropdown menu set to 'Sala de Monitoria'. To the right of these fields is a blue button labeled 'Adicionar'.

Figura 5.14: *Adicionar Horário de Monitoria*

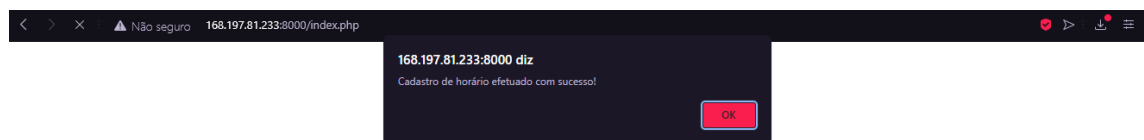


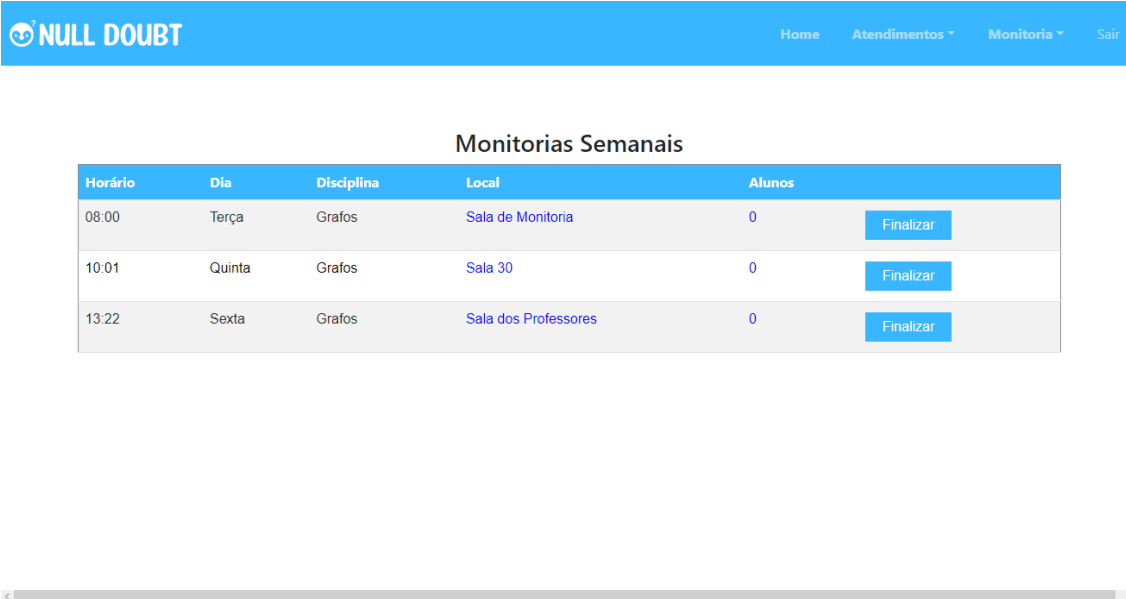
Figura 5.15: *Notificação após adicionar Horário de Monitoria*

5.2.10 Visualizar Próximas Monitorias

Todas as informações da tabela de monitorias foram lidas como o esperado. Logo, o teste foi totalmente aceito.

horario	dia	cod_monitoria	fk_local	fk_discente	fk_disciplina	ativo
20:42:00	3	3	110	2020102201940202	4	1
08:00:00	2	4	15	2020102201940040	1	1
13:22:00	5	5	2023	2020102201940040	1	1
10:01:00	4	6	2024	2020102201940040	1	1

Figura 5.16: *Visualizar Monitorias / Tabela de monitorias*

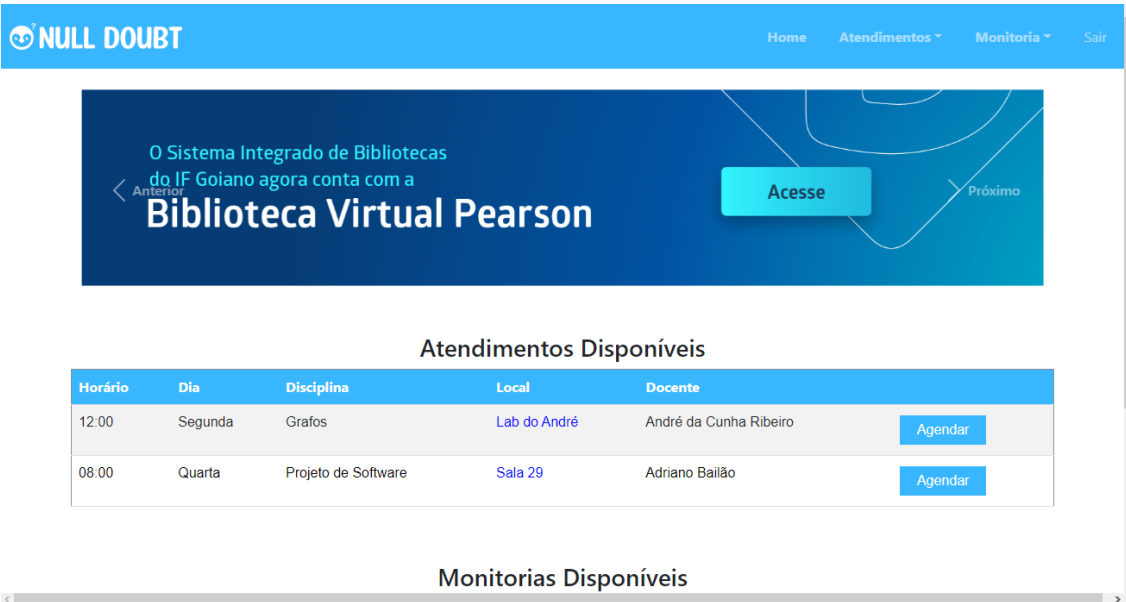


Horário	Dia	Disciplina	Local	Alunos	
08:00	Terça	Grafos	Sala de Monitoria	0	Finalizar
10:01	Quinta	Grafos	Sala 30	0	Finalizar
13:22	Sexta	Grafos	Sala dos Professores	0	Finalizar

Figura 5.17: Visualizar Monitorias / Tela do Discente promovido à Monitor

5.2.11 Agendar Atendimento ou Monitoria

Todas as informações da tabela de atendimentos monitorias foram lidas e listadas como o esperado. Outrossim, o cadastro de agendamento do discente tanto pro atendimento, quanto pra monitoria, foi efetuado com sucesso. Logo, o teste foi totalmente aceito.



Horário	Dia	Disciplina	Local	Docente	
12:00	Segunda	Grafos	Lab do André	André da Cunha Ribeiro	Agendar
08:00	Quarta	Projeto de Software	Sala 29	Adriano Bailão	Agendar

Figura 5.18: Agendar Atendimento - Discente

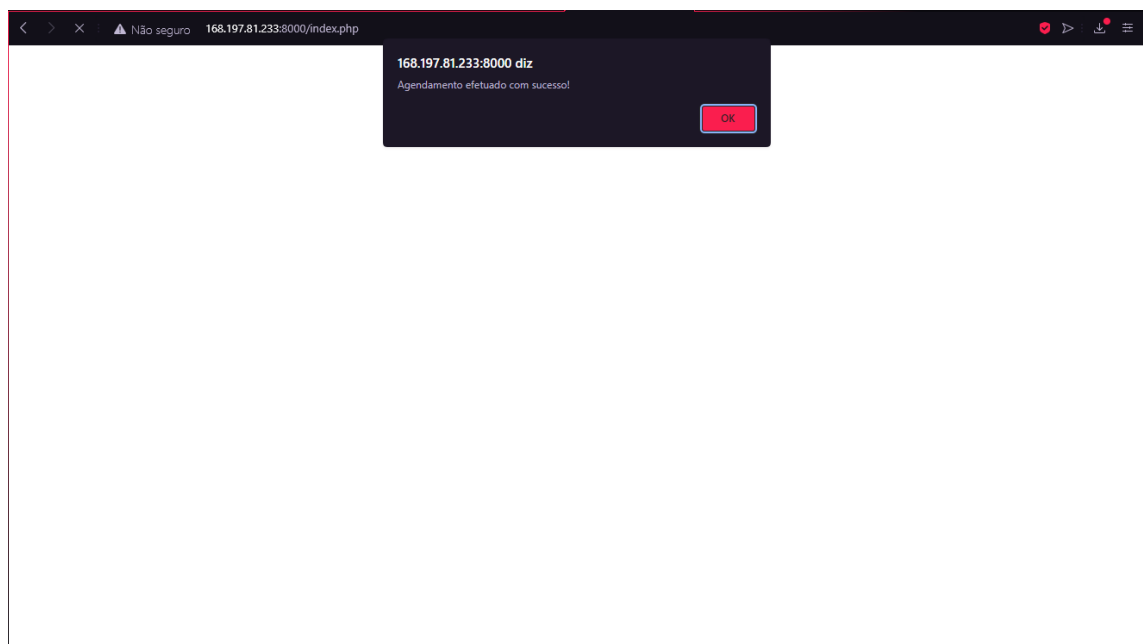


Figura 5.19: *Agendamento de Atendimento Efetuado - Discente*



Figura 5.20: *Agendamento de Atendimento Efetuado - Discente*

Atendimento			
Horário	Dia	Disciplina	Local
12:00	Segunda	Grafos	Lab do André

Discentes Agendados		
Matricula	Nome	Curso
2020102201940040	Gustavo Moura Barros	Ciências da Computação

Figura 5.21: *Agendamento do Discente no Atendimento*

5.2.12 Desmarcar Agendamento

Todas as informações da tabela de atendimentos monitorias foram atualizadas como o esperado depois de desmarcar o agendamento. A tela foi atualizada e foi possível fazer outros agendamentos. Sendo assim, o teste foi aceito.

Atendimentos Disponíveis					
Horário	Dia	Disciplina	Local	Docente	
12:00	Segunda	Grafos	Lab do André	André da Cunha Ribeiro	Desmarcar
07:00	Sexta	TCC - 1	Sala de Monitoria	André da Cunha Ribeiro	Desmarcar
08:00	Quarta	Projeto de Software	Sala 29	Adriano Bailão	Agendar

Figura 5.22: *Desmarcar Atendimento - Discente*



Figura 5.23: Desmarcar Atendimento - Discente

5.2.13 Teste de Desempenho

O sistema foi testado em diversas sessões, resultando em nenhum comprometimento do servidor, concluindo-se em um software com um bom desempenho.

5.2.14 Teste de Segurança

Foram realizados diversos testes de segurança e tentativas de manipulação no processo de registro de evidências providas pela plataforma. Nesta análise foram encontradas diversas proteções de segurança para evitar ataques simples e sofisticados no processo de coleta de informações fornecido pela plataforma, bem como em outros pontos do sistema.

Conclusões

O software Null Doubt foi criado com o intuito de aumentar o desempenho acadêmico dos discentes, a partir do saciamento de dúvidas. Além disso, facilitará o gerenciamento dos atendimentos e monitorias prestados durante o semestre.

Todos os requisitos de alta prioridades foram atendidos. Os teste foram feitos, executados e retornaram os resultados esperados.

Como implementações futuras serão implementados os históricos completos e um histórico gráfico analítico. O histórico completo será implementado de forma tabular para todos os usuários do sistema, mostrando todos os agendamentos efetuados. Já o histórico gráfico analítico, trará um gráfico que mostrará o nível de aprendizado dos discentes, que usaram o software para tirar suas dúvidas a partir de um atendimento ou monitoria.

Referências Bibliográficas

- [1] ANDRADE, E. G. R. D.; RODRIGUES, I. L. A.; NOGUEIRA, L. M. V.; SOUZA, D. F. D. **Contribuição da monitoria acadêmica para o processo ensino-aprendizagem na graduação em enfermagem.** *Revista Brasileira de Enfermagem*, 71:1596–1603, 2018.
- [2] CORRAL, L.; JANES, A.; REMENCIUS, T. **Potential advantages and disadvantages of multiplatform development frameworks—a vision on mobile environments.** *Procedia Computer Science*, 10:1202–1207, 2012.
- [3] DE MIRANDA COSTA, P. **Monitoria: Avaliação docente.** *Série Acadêmica*, 25(jan./dez.):1–14, 2009.
- [4] EXECUTIVO. **Regime jurídico dos servidores públicos civis da união, das autarquias e das fundações públicas federais.** 1990.
- [5] GIACOBO, D. **Scaa-software de controle de atendimento ao aluno.** In: *Anais dos Workshops do IX Congresso Brasileiro de Informática na Educação*, p. 104–107. SBC, 2020.
- [6] LINS, L. F.; FERREIRA, L. M. C.; FERRAZ, L. V.; CARVALHO, S. S. G. D. **A importância da monitoria na formação acadêmica do monitor.** *Jornada de ensino, pesquisa e extensão, IX*, 2009.
- [7] RASTOGI, A.; GUPTA, K. **Student attendance through mobile devices.** PhD thesis, 2013.
- [8] SOMASUNDARAM, V.; KANNAN, M.; SRIRAM, V. **Mobile based attendance management system.** *Indian Journal of Science and Technology*, 9(35):1–4, 2016.
- [9] SOUZA, V. F.; DA SILVA, L. O.; DOS SANTOS, A. D. F. **Monitoria geral 2020.5-programação 1.** *Seminário de Projetos de Ensino (ISSN: 2674-8134)*, 5(1):1–5, 2021.
- [10] SUPERIOR, C. **Regulamento de atividades dos docentes - RAD - do IF goiano.** 2016.

- [11] SUPERIOR, C. **Regulamento dos cursos de graduação do instituto federal de educação, ciência e tecnologia goiano**. 2019.