

**UTFPR-UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ**

*Bacharelado em Engenharia de Software - 4º Período*

**DISCIPLINA:** *Oficina de Integração 1 - ES63G-ES41*

**PROFESSOR:** *Adriano Rivolli*

---

# **Documento de**

# **Projeto de Software**

---

**UniAgenda:**  
**agenda de serviços e reservas de sala**  
**para Universidades.**

**Daniel Gonçalves**  
**Jaine Saconi**  
**Luiz Henrique Carvalho**  
**Marina Fachin**

Cornélio Procópio

2019

## Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>2</b>
1.1	Contexto	2
1.2	Justificativa	2
1.3	Proposta	2
1.4	Organização do Documento	2
<b>2</b>	<b>DESCRIÇÃO GERAL DO SISTEMA</b>	<b>3</b>
2.1	Objetivos (Gerais e Específicos)	3
2.2	Limites e Restrições	3
2.3	Descrição dos Usuários do Sistema	3
<b>3</b>	<b>DESENVOLVIMENTO DO PROJETO</b>	<b>4</b>
3.1	Tecnologias e ferramentas	4
3.2	Metodologia de desenvolvimento	4
3.3	Cronograma previsto	4
<b>4</b>	<b>REQUISITOS DO SISTEMA</b>	<b>5</b>
4.1	Requisitos Funcionais	5
4.2	Requisitos Não-funcionais	5
4.3	Diagramas de Casos de Uso	5
<b>5</b>	<b>ANÁLISE DO SISTEMA</b>	<b>6</b>
<b>6</b>	<b>IMPLEMENTAÇÃO</b>	<b>7</b>
6.1	Protótipos de Telas	7
6.2	Descrição do código	7
<b>7</b>	<b>CONSIDERAÇÕES FINAIS</b>	<b>8</b>
<b>8</b>	<b>BIBLIOGRAFIA</b>	<b>9</b>

# 1 Introdução

## 1.1 Contexto

Nos dias atuais as universidades se tornaram uma instituição de ensino e pesquisa constituída por um conjunto de faculdades e escolas destinadas a promover a formação profissional e científica de pessoal de nível superior, e a realizar pesquisas teóricas e práticas nas principais áreas do saber humanístico, tecnológico e artístico e a divulgação de seus resultados à comunidade científica mais ampla.

Assim observamos que uma parcela relevante das Instituições de Educação Superior (IES) brasileiras prestam também outros serviços à comunidade acadêmica além da oferta de cursos superiores, serviços de secretaria e os serviços relacionados a apoio a pesquisas. Essas IES também contam com departamentos que fazem atendimentos à comunidade, prestando serviços como: atendimento psicológico, pedagógico, psicopedagógico, odontológico, relações empresariais e oferta de estágios, atendimento das chefias de departamentos e coordenações de cursos, além do atendimento de técnicos de TI e laboratórios.

A Universidade Tecnológica Federal do Paraná campus Cornélio Procopio é uma dessas IES que oferecem esses serviços citados e outros serviços como por exemplo o agendamento do uso de determinados espaços físicos como anfiteatro, quadras, salas de dança e ginástica assim como determinados laboratórios.

Diante desta realidade, esta parcela das IES realizam a gestão do atendimento de forma setorizada, ficando a critério de cada departamento formalizar seus próprios padrões de atendimento, isso dificulta a gestão do controle de atendimentos de uma forma unificada e atrasa a equipe durante o processo de tomada de decisões estratégicas para o bom desenvolvimento dos trabalhos.

Essa divisão dos atendimentos de forma setorizada também cria dificuldades para os discentes que precisam ser atendidos por alguns desses serviços. Essas dificuldades podem ser observadas na dificuldade de encontrar horários disponíveis na sua grade de aulas que sejam compatíveis com os horários disponíveis do servidor que realizará o atendimento e ou até o próprio deslocamento do discente até ao determinado departamento para realizar o pedido de atendimento presencialmente sem poder saber previamente quando ele poderá ser atendido por um servidor.

## 1.2 Justificativa

O presente projeto se justifica pela necessidade de parte das IES de aprimorar a gestão do atendimento dos serviços prestados e das reservas de espaços físicos, e assim melhorar a qualidade do trabalho da equipe de atendimento, e a qualidade do próprio serviço prestado.

A prática da divisão da gestão dos atendimentos de forma setorizada e não informatizada que podemos observar em algumas destas instituições acaba criando um problema na hora de gerar dados unificados de atendimento de cada servidor e de cada departamento de forma ágil, implicando no atraso de tomada de decisões gerenciais, atraso este provocado justamente pela dificuldade em levantar dados consistentes, e gerar informações estatísticas com rapidez, que podem diminuir a probabilidade de atrasos nas tomada de decisões.

Outro problema que pode ser observado nesses casos é a dificuldade dos estudantes em conseguir encontrar horários livres compatíveis com os horários de atendimento disponíveis em cada setor ou para cada servidor, outro fator de dificuldade criado pela fragmentação do atendimento é um maior desconhecimento por parte dos alunos em relação a alguns serviços prestados e de como eles funcionam, diminuindo assim a procura dos estudantes por esses serviços.

Também neste contexto é possível apontar o problema ocasionado no caso das reservas de espaços físicos como salas de aula, laboratórios, quadras e anfiteatros, que assim como os serviços prestados têm suas gestões realizadas por departamentos diferentes, e isso acaba causando conflitos na agenda dessas reservas.

Portanto pretende-se com este projeto impactar servidores e alunos das IES, por meio de um sistema informatizado de agendamento de atendimento e reservas de espaços físicos com a função de unificar os registros gerados e automatizar o agendamento destes, a fim, de otimizar o controle gerencial por parte da IES e ao mesmo tempo facilitar o acesso aos serviços pela comunidade acadêmica.

### **1.3 Proposta**

O presente projeto apresenta a proposta de desenvolvimento de uma aplicação web com o objetivo de melhorar o gerenciamento dos agendamentos de serviços prestados pelas instituições de ensino e também automatizar os processos de reservas de espaços físicos dentro das IES brasileiras.

A referida aplicação terá funções que permitem realizar o cadastro dos usuários, diferenciando as funções do usuário servidor da instituição e as funções relativas aos usuários discentes. Os usuários do tipo servidor terão acesso a serviços como: cadastro do usuário, cadastro de horários de atendimento, consultas dos agendamentos realizados por dia, semana e meses, geração de relatórios de números de atendimentos, relatórios de agendamentos que foram atendidos e os que não foram atendidos, impressão da agenda semanal e mensal de atendimentos, cadastros de espaços físicos a serem reservados, agendar a reserva desses espaços físicos e consultar a agenda de horários dos discentes a serem atendidos.

Esta aplicação também disponibilizará serviços destinados aos usuários discentes que poderão ter acesso aos serviços de cadastramento de usuários, e também ao cadastro da agenda discente que poderá ser preenchida com os horários de suas aulas e demais atividades que compõem sua agenda de compromissos. Através desta aplicação o usuário discente poderá requisitar agendamento de atendimentos realizados por diversos setores da universidade, além de também poder realizar reservas de espaços físicos como: quadras, laboratórios, anfiteatros e demais espaços pertencentes a universidade que permitam serem reservados por estudantes.

O sistema proposto pretende impactar a comunidade acadêmica de servidores e discentes, com o objetivo de resolver os problemas referentes aos agendamentos de atendimento de serviços prestados pela IES, fazendo a automação e a otimização de parte dos processos de agendamento de serviços e escolhas de horários livres para atendimento de discentes. Também pretende-se resolver

de forma informatizada o processo de reservas de espaços físicos, proporcionando maior controle da utilização destes espaços e agilizando as tomadas de decisões por parte dos gestores.

O armazenamento e a gestão do projeto a ser desenvolvido será feito na plataforma de gestão de projetos Github. O endereço para acesso do repositório do projeto é <https://github.com/silveirah/Oficina1-Agendamentos/>.

## 1.4 Equipe de desenvolvimento

A equipe de desenvolvimento para este projeto será composta por quatro membros, a seguir apresentamos um perfil de cada membro contendo a descrição de suas habilidades e suas preferências técnicas e como elas poderão contribuir para o projeto.

Membro 1: Daniel Gonçalves da Silva - É graduando do 6º período do curso de Engenharia de software na UTFPR campus Cornélio Procopio, possui habilidades de engenharia de software e programação nas linguagens C,C++, Java, Visual Basic 6.0, .net e Delphi . Tem preferências técnicas de aprendizagem nas áreas de engenharia de software, desenvolvimento de aplicações Restfull e em nuvem, servless, investigação de técnicas de mineração dados, aprendizado de máquinas e IA. Possui 11 anos de experiência profissional de desenvolvimento com Delphi, aplicado a sistemas desktop para o setor contábil e de departamento pessoal além de sistemas de ERPs legados do sistema operacional DOS com uso de linguagem Clipper V5.3 e Basic. Possui como características pessoais boa comunicação, curiosidade sobre o funcionamento das coisas, vontade de aprender e espírito investigativo. Contribuirá para o projeto com conhecimentos de modelagem de banco de dados, análise de sistema, linguagem java e framework spring boot e gestão de RH.

Membro 2: Jaine Saconi Coelho - Técnico em Informática pela ETEC Marins Teodoro de Freitas Almeida, estagiária de desenvolvimento na empresa CDS Software, graduando em engenharia de software, responsável pela documentação e desenvolvimento.

Membro 3: Luiz Henrique Carvalho - Técnico em Informática pela UTFPR, campus Campo Mourão e graduando em Engenharia de Software e responsável pela documentação.

Membro 4: Marina Fachin - Graduanda em engenharia de software e responsável pela implementação do sistema na parte de front-end.

## 1.5 Organização do Documento

O documento ainda está em fase construção, sendo assim a organização do documento será feita durante as próximas etapas de desenvolvimento.

Para esta primeira entrega o documento está organizado em quatro capítulos, sendo: 1-introdução, 2-Descrição Geral do Sistema, 3-Desenvolvimento do projeto e 4-Requisitos do sistema

## 2 Descrição Geral do Sistema

## 2.1 Objetivos (Gerais e Específicos)

O objetivo deste projeto é cumprir a demanda de agendamentos de salas de aula, horários de atendimento por parte de servidores da Universidade e também reserva para o uso de diversas áreas do campus, como a quadra esportiva, academia e anfiteatro através do desenvolvimento de uma plataforma Web integrando aluno e servidor respeitando as restrições e cumprindo o cronograma de execução.

- A aplicação terá o objetivo de gerenciar a agenda de cada servidor, que ficará responsável de cadastrar seus horários para atendimento de acordo com sua disponibilidade.
- Controlar o agendamento por parte dos alunos também será um dos objetivos específicos da plataforma, gerenciando informações relativas ao aluno e suas solicitações de horários de atendimento.
- Deixar à disposição dos usuários interessados em áreas da Universidade que necessitam agendamento prévio a possibilidade de reserva dessas áreas.

## 2.2 Limites e Restrições

### 2.2.1 Limites

- Todos os alunos frequentadores do órgão que possui a implementação do sistema podem acessá-lo e fazer uma reserva de reunião. Porém, a opção cadastro de servidor é acessível apenas aos servidores e administradores.
- Os servidores não são responsáveis pelo não comparecimento de alunos nas reuniões, cancelamentos e atrasos. Caso ocorra uma mudança de horário na agenda do servidor é de total responsabilidade do próprio alterar no sistema e avisar o cancelamento ou mudança de horário da reunião, caso haja alguma agendada para o horário específico.
- O sistema não possui acesso ao banco de dados do órgão que o utiliza, portanto não importa os dados dos alunos e servidores de forma automática. Os alunos devem fazer seu cadastro, e após isso aguardar um período para utilizar o sistema, no qual será avaliado pelo administrador para verificar se os dados informados são verdadeiros ou não.
- O sistema não se responsabiliza pelo envio de e-mail de confirmação do agendamento.
- O sistema não se responsabiliza pela exclusão do discente que não atua mais no órgão que o utiliza.
- O sistema não se responsabiliza pelo cadastramento das salas que serão disponíveis para agendamento.
- O sistema não se responsabiliza pela entrega e devolução da chave da sala requerida.

### 2.2.2 Restrições

- Aplicação não funcionará offline, impossibilitando agendamentos sem o acesso à internet;

## **2.3 Descrição dos Usuários do Sistema**

Tendo em vista facilitar a comunicação e agendamento de serviços entre servidor e discente da instituição de ensino superior, os usuários principais do sistema proposto serão os próprios servidores e os discentes. Portanto os atores do sistema são os usuários que fazem parte da comunidade acadêmica, os servidores terão acesso através do código SIAPE e o usuário discente terá acesso ao sistema pelo número do registro de matrícula (RA).

O papel do servidor será alimentar o sistema com seus horários para atendimento, validar os atendimentos realizados e os cancelados e ter acessos as consultas e relatórios gerenciais relativos ao seu departamento.

O papel do discente será alimentar o sistema com os seus horários de aula e demais compromisso, requisitar agendamento de serviços prestados na universidade ou agendamento de uso de espaços físicos.

## 3 Desenvolvimento do Projeto

### 3.1 Tecnologias e ferramentas

As tecnologias usadas no desenvolvimento do projeto estão listadas abaixo:

#### **Organizacional**

- Google drive

#### **Gerenciamento de atividades**

- Github

#### **Modelagem**

- Astah Community
- MySQL Workbench

#### **Tecnologias**

- Java 8 SE

#### **Desenvolvimento e codificação**

- Netbeans
- Framework Spring Boot

#### **Banco de Dados**

- MySql

### 3.2 Metodologia de desenvolvimento

A metodologia utilizada para o desenvolvimento do projeto será o Scrum, um framework simples para gerenciar projetos complexos que, de acordo com o CHAOS Report (2015), se mostra mais eficiente que metodologias tradicionais como a cascata, tendo uma proporção de projetos bem sucedidos visivelmente maior quando comparado a este outro.

Baseado no trabalho em equipe, o scrum foi bem adequado a projetos formados por pequenos grupos de pessoas multidisciplinares, que é o perfil do software a ser desenvolvido nesse projeto. Uma das características principais do scrum é o fato dele ser aplicado em um ambiente onde os requisitos são pouco



conhecidos e as tecnologias que envolvem o problema são complexas.

O scrum é fundamentado em três pilares que geram a sua dinâmica, são eles *transparência, inspeção, adaptação*. A transparência permeia a ideia de que todos envolvidos no projeto têm conhecimento dos requisitos, processos e do andamento dos projetos. Já na inspeção, o tempo todo é inspecionado o que está sendo desenvolvido no projeto e por fim a adaptação, que é a mutação constante do projeto conforme necessidades.

Composto também por práticas fundamentais, como papéis (Scrum master, product owner e dev team), eventos básicos (Planejamento do sprint - Sprint Planning -, execução do sprint, revisão do sprint, reuniões diárias e retrospectiva) e artefatos gerados (Product backlog, Sprint backlog, incremento/entrega parcial), a metodologia de desenvolvimento se molda em ser resiliente à dinâmica de mudanças do projeto.

Nos papéis, o Product Owner é quem tem os poderes de liderança sobre o produto. Ele é o responsável por decidir quais recursos e funcionalidades serão construídos e a ordem em que devem ser feitos. O scrum master é o responsável por ajudar todos os envolvidos a entender e alinhar princípios e práticas do scrum. O Dev Team é o conjunto de pessoas que de fato vão construir o projeto e decidem como fazê-lo.

A dinâmica do scrum começa com a visão do produto que é direcionada pelo product owner, que descreve o que ele quer e onde quer chegar, fazendo um desmembramento em funcionalidades (product backlog). O projeto é planejado em sprints, que são períodos de tempo em que alguns itens selecionados do product backlog serão construídos e entregues de acordo com o time-boxed, que é o tempo fixo de duração de cada sprint.

Antes de cada sprint começar é feita uma reunião de planejamento dos sprints (sprint planning), onde é criado o backlog da sprint.

Com base na capacidade e velocidade da equipe scrum, é definido quantas funcionalidades podem ser completamente construídas no tempo do sprint. Primeiro, as funcionalidades priorizadas no product backlog são selecionadas e construídas, e, após o término da sprint, é esperado que o incremento (parte do software) funcionando seja entregue. Durante o desenvolvimento das sprints, o product owner pode verificar necessidades de mudanças que devem ser inseridas no backlog, e o processo então é repetido até que todo backlog seja construído e o produto final esteja pronto.

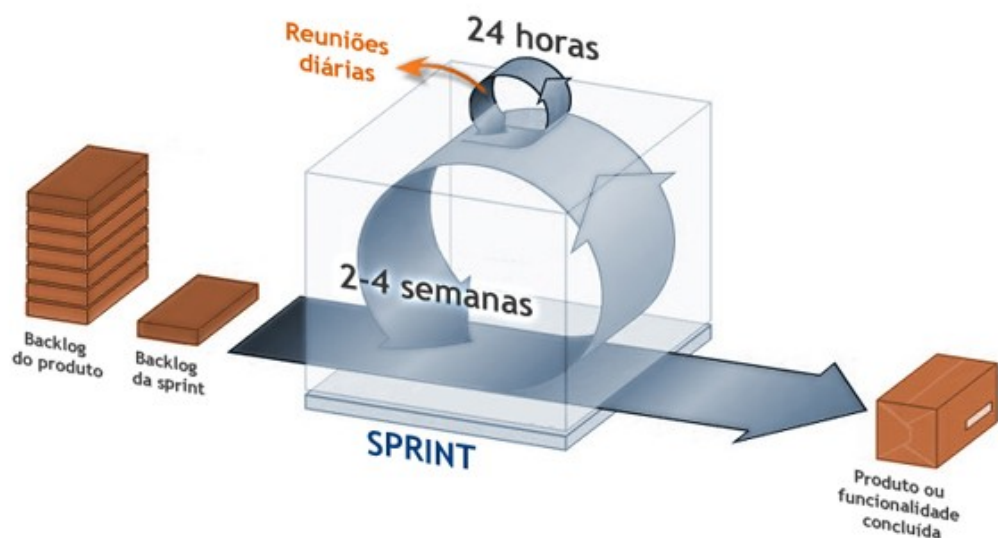
Na aplicação comum do framework, todo dia é feita uma reunião de 15 minutos onde cada membro do time deve responder a três perguntas básicas: o que eu fiz ontem? O que vou fazer hoje? Há algum impedimento? Com essa reunião, todos conseguem visualizar de uma maneira geral como está progredindo o trabalho da sprint.

Durante o projeto, porém, serão realizadas reuniões semanais, ao invés das tradicionais reuniões diárias, para discutir o andamento do projeto, e, portanto, algumas das perguntas básicas passam a ser: o que eu fiz durante a última semana? O que vou fazer na próxima?

As sprints serão divididas para abranger três principais pontos: o levantamento de requisitos e geração dos diagramas, a documentação do projeto e a implementação, aplicando múltiplas sprints para o desenvolvimento destes dois último itens, enquanto que para a atividade inicial será realizada apenas uma sprint. Cada sprint terá uma duração média de uma semana.

No final de cada sprint existem duas atividades finais que são fundamentais. A sprint review, que tem por objetivo validar e adaptar o produto que está sendo construído, e a retrospectiva, a qual tem por objetivo verificar necessidades de adaptação no processo, compreender o que foi feito de positivo e negativo, o que se deve melhorar e o que se deve parar de fazer.

Para entender melhor a figura simula o fluxo da metodologia scrum:



*Figura 1 – Ilustração do processo SCRUM*

Durante a implementação do sistema, serão realizadas oito sprints, cada uma com uma semana, diferentemente da aplicação padrão da metodologia.

Em cada sprint da implementação, serão produzidos diferentes componentes do sistema. Inicialmente nas primeiras quatro sprints, serão as reuniões de análise e levantamento de dados e definição dos requisitos do sistema, definição do contexto, proposta e escopo a ser desenvolvido.

A partir da quinta sprint será realizada a preparação do ambiente para desenvolvimento, preparação da estrutura do sistema que será executada no servidor

(back-end) e desenvolvimento do padrão de telas para a aplicação que será executada na máquina do cliente (front-end).

Na sexta sprint, será iniciada a codificação da estrutura lógica do back-end, com a preparação das classes controladoras e entidades, aperfeiçoamento do front-end e desenvolvimento da comunicação entre a aplicação cliente e a aplicação servidor.

Já na sétima sprint, serão aperfeiçoadas as comunicações entre os dois extremos do sistema (lado cliente e lado servidor) e será realizada a integração completa de ambas as aplicações.

Por fim, na última sprint, serão realizados testes de aceitação e eventuais ajustes para garantir que os objetivos do sistema sejam alcançados e que o sistema está funcionando corretamente e se comportando conforme o esperado e será documentada a conclusão do projeto.

### **3.3 Cronograma previsto**

O cronograma a seguir explicita o desenvolvimento previsto para ser ocorrido ao longo do período de execução do projeto, separando-o em semanas e demonstrando as tarefas realizadas em cada uma.

#### **Primeira semana**

Durante a primeira semana, será realizado o levantamento de requisitos para o sistema.

##### **Responsáveis:**

- 1.1 Contexto (Daniel Gonçalves e Marina Fachin)
- 1.2 Justificativa (Daniel Gonçalves)
- 1.3 Proposta (Daniel Gonçalves)
- 1.4 Organização do Documento (Daniel Gonçalves)

#### **Segunda semana**

Ao longo da primeira semana ocorrerá o desenvolvimento do texto do projeto, especificamente os itens 1 e 2 deste documento.

##### **Responsáveis:**

- 4.1 Requisitos Funcionais (Daniel Gonçalves )
- 4.2 Requisitos Não-funcionais ( Luiz Henrique Carvalho )

- 4.3 Diagrama de Caso de Uso (Marina Fachin, Daniel Gonçalves )
- 2.1 Objetivos (Gerais e Específicos) (Luiz Henrique Carvalho)
- 2.2 Limites e Restrições (Jaine Saconi)
- 2.3 Descrição dos Usuários do Sistema ( Jaine Saconi, Daniel G.)

### **Terceira semana**

Ao longo da segunda semana será continuado o desenvolvimento da documentação do projeto, envolvendo os itens 3.

#### **Responsáveis:**

- 3.1 Tecnologias e ferramentas ( Daniel Gonçalves )
- 3.2 Metodologia de desenvolvimento ( Daniel Gonçalves )
- 3.3 Cronograma previsto ( Daniel Gonçalves )

### **Quarta semana**

Ao longo da terceira semana a documentação do projeto será continuada, produzindo especificamente os itens 5 e 6.

#### **Responsáveis:**

- 5.1 Modelo do Banco de Dados ( Daniel Gonçalves )
- 5.1.1 Modelo Conceitual ( Daniel Gonçalves)
- 5.1.2 Modelo Lógico ( Daniel Gonçalves )
- 5.2 Diagrama de Classes (Marina)
- 5.3 Diagrama de Atividades (Marina )

### **Quinta semana**

Durante a quinta semana, será continuado o desenvolvimento do item 6 desta documentação, ou seja, a implementação do sistema, além de algumas correções.

#### **Responsáveis:**

- 5.6 Correção dos itens 3 e 4 - Todos os membros
- 5.7 Preparação do ambiente para desenvolvimento - Todos os membros
- 5.8 Preparação da estrutura da aplicação servidor -
- 5.9 Desenvolvimento do padrão de telas da aplicação cliente

### **Sexta semana**

Ao longo da sexta semana, será continuado o desenvolvimento da implementação do sistema.

#### **Responsáveis:**

- 6.1 Codificação da estrutura básica da aplicação servidor -
- 6.2 Aperfeiçoamento das telas da aplicação cliente -
- 6.3 Desenvolvimento da comunicação entre as aplicações os servidor - Todos os membros
- 6.4 Item 6.2 (Descrição do código) - Todos os membros

### **Sétima semana**

Na sétima semana, o desenvolvimento da implementação do sistema continuará a ser realizado.

#### **Responsáveis:**

- 7.1 Aperfeiçoamento da comunicação e integração total entre as aplicações cliente e servidor - Todos os membros
- 7.2 Item 6.2 (Descrição do código) - Todos os membros

### **Oitava semana**

Na última semana, será continuado o desenvolvimento do item 6 e a produção do item 7 desta documentação.

#### **Responsáveis:**

- 8.1 Testes do sistema - Todos os membros
- 8.2 Correções de erros do sistema - Todos os membros
- 8.3 Item 6.2 (Descrição do código) - Todos os membros
- 8.4 Item 7 (Considerações finais) e correção dos itens 5 e 6 - Todos os membros;

## 4 Requisitos do Sistema

### 4.1 Requisitos Funcionais

Os requisitos funcionais listados abaixo estão classificados por ordem de acordo com suas prioridades:

Essencial – deve ser implementado para que o sistema funcione.

Importante – sem este requisito o sistema pode funcionar, mas não da maneira esperada.

ID	Funcionalidade	Prioridade
01	Cadastrar usuário do tipo servidor (Crud)	Essencial
02	Cadastrar usuário do tipo discente (Crud)	Essencial
03	Cadastrar horários de atendimento dos servidores (Crud)	Essencial
04	Cadastrar horários de aulas dos alunos (Crud)	Essencial
05	Servidor Cadastrar Salas (Crud)	Essencial
06	Servidor Cadastrar Cargos dos Servidores (Crud)	importante
07	Servidor Cadastrar Curso dos Alunos (Crud)	importante
08	Discente e Servidor efetua Agendamento de sala (Crud)	Essencial
09	Discente efetua Agendamento de Atendimento (Crud)	Essencial
10	Selecionar salas disponíveis para agendamento por data e horário.	Essencial
11	Selecionar horários disponíveis para atendimento com servidor por data	Essencial
12	Listar quantidade de atendimentos por data	importante
13	Fazer Login/logoff de usuário	Importante
14	Listar agendamentos do aluno, por dia, semana e mês	Essencial
15	Listar agendamentos do servidor, por dia, semana e mês	Essencial
16	Listar agendamentos de sala, por dia, semana e mês	Essencial
17	Listar horários livres para atendimento entre o servidor e o aluno requerente	Importante

Descrição detalhada dos requisitos funcionais previstos:

**RF 01 - Cadastrar usuário do tipo servidor:** O sistema deve permitir o cadastro, consulta, alteração e exclusão de usuários do tipo servidor..

**RF 02 - Cadastrar usuário do tipo discente:** O sistema deve permitir o cadastro, consulta, alteração e exclusão de usuários do tipo discente

**RF 03 - Cadastrar horários de atendimento dos servidores:** O sistema deve permitir o cadastro, consulta, alteração e exclusão dos horários de atendimento do servidor.

**RF 04- Cadastrar horários de aulas dos alunos:** O sistema deve permitir o cadastro, consulta,

alteração e exclusão dos horários de aulas de cada discente.

**RF 05- Servidor Cadastrar Salas:** O sistema deve permitir o cadastro, consulta, alteração e exclusão das salas disponíveis para agendamento pelo usuário do tipo servidor.

**RF 06-Servidor Cadastrar Cargos dos Servidores:** O sistema deve permitir o cadastro, consulta, alteração e exclusão dos cargos ocupados por servidores.

**RF 07-Servidor Cadastrar Curso dos Alunos:** O sistema deve permitir o cadastro, consulta, alteração e exclusão dos cursos existentes no campus pelo usuário do tipo servidor.

**RF 08-Discente e Servidor efetua Agendamento de sala:** O sistema deve permitir o agendamento de salas de aulas e demais espaços físicos pelos usuários discente e servidor.

**RF 09-Discente efetua Agendamento de Atendimento :** O sistema deve permitir que discentes façam o agendamento de horário de atendimento para serviços prestado por algum servidor.

**RF 10-Selecionar salas disponíveis para agendamento por data e horário:** O sistema deve permitir que o usuário selecione data e hora

**RF 11-Selecionar horários disponíveis para atendimento com servidor por data:** O sistema deve permitir que discentes façam o agendamento de data e horário de atendimento para serviços prestado por algum servidor

**RF 12-Listar quantidade de atendimentos por data:** O sistema deve permitir que o servidor liste a quantidade de atendimentos feitos por data.

**RF 13-Fazer Login/logoff de usuário:** O sistema deve permitir que o usuário faça login para acessar demais áreas do sistema e também permita fazer o logoff quando desejar sair do sistema

**RF 14-Listar agendamentos do aluno, por dia, semana e mês:** O sistema deve permitir a listagem dos agendamentos do aluno por dia, semana e mês.

**RF 15-Listar agendamentos de sala, por dia, semana e mês:** Sistema deve permitir a listagem dos agendamentos de sala e demais espaços físicos por dia, semana e mês.

**RF 16-Listar agendamentos de serviços, por dia, semana e mês:** O sistema deve permitir a listagem dos agendamentos de serviços realizados e a realizar por dia, semana e mês

**RF 17-Listar horários livres para atendimento entre o servidor e o aluno requerente:** O sistema deve permitir a listagem dos horários livres para atendimento entre servidor e os discentes que requererem o agendamento.

## 4.2 Requisitos Não-funcionais

ID	Requisito	Categoria
01	O sistema deve ter um bom funcionamento em diferentes browsers	Portabilidade
02	O Design das interfaces gráficas será responsivo.	Usabilidade
03	Para o login, o usuário do tipo servidor deverá possuir seu código SIAPE.	Segurança
04	Para o login, o usuário do tipo discente deverá possuir seu código do Registro Acadêmico (RA)	Segurança

## 4.3 Diagramas de Casos de Uso

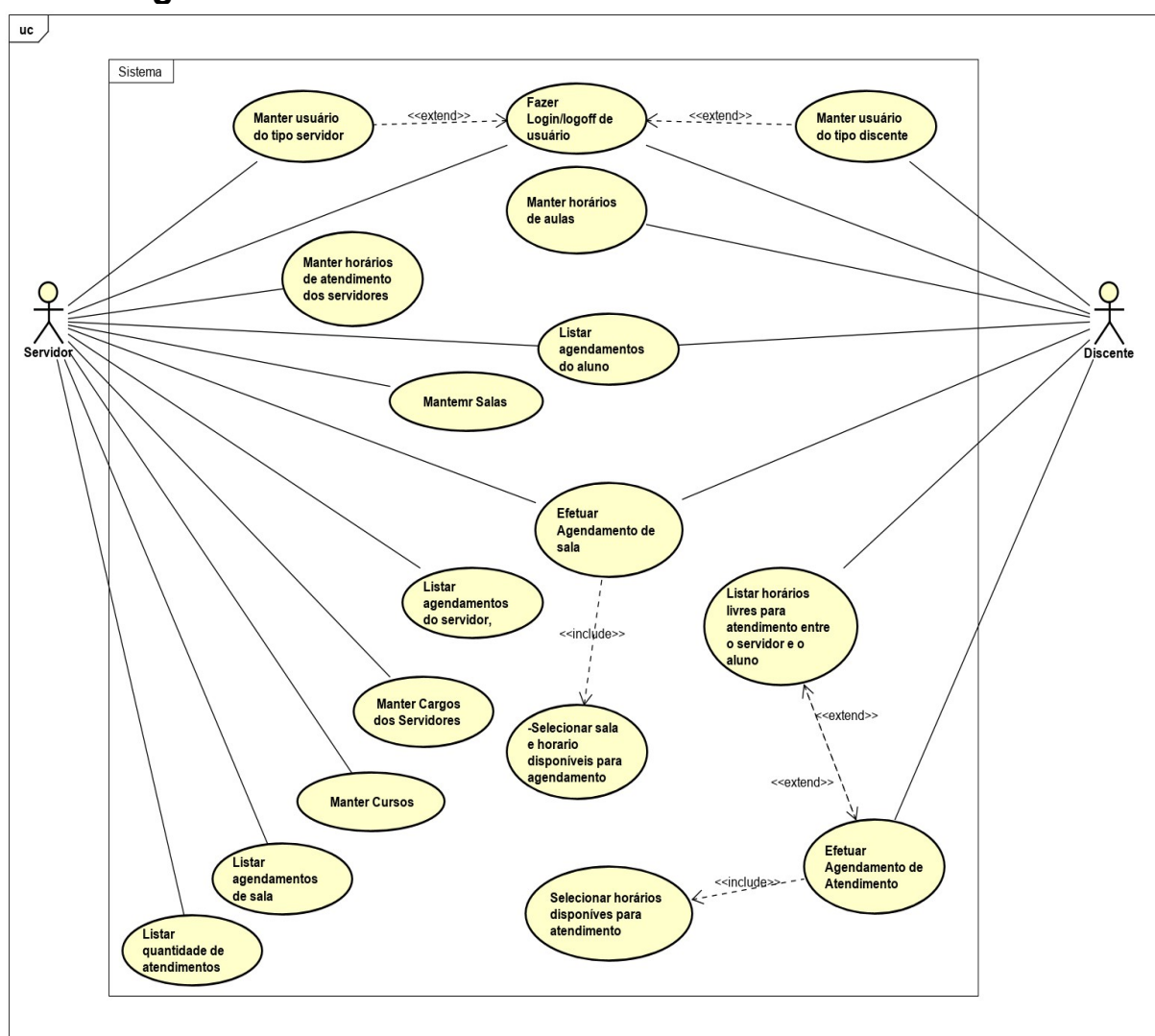


Figura 2. Diagrama de Caso de Uso Geral



## 5 Análise do Sistema

Este item apresenta a documentação da análise do sistema conforme o processo ou ciclo de vida descrito no capítulo 3.

### - Modelo Lógico do Banco de Dados

O diagrama abaixo foi criado com o auxílio da ferramenta Workbench, que pertence ao pacote MySQL Community, este diagrama de entidade-relacionamento mostra as principais entidades que compõem o escopo do projeto, destacando os campos chaves primárias com o símbolo de Exclamação amarelo, as chaves estrangeiras com símbolos da cor vermelha e os símbolos azuis são os demais campos da entidade.

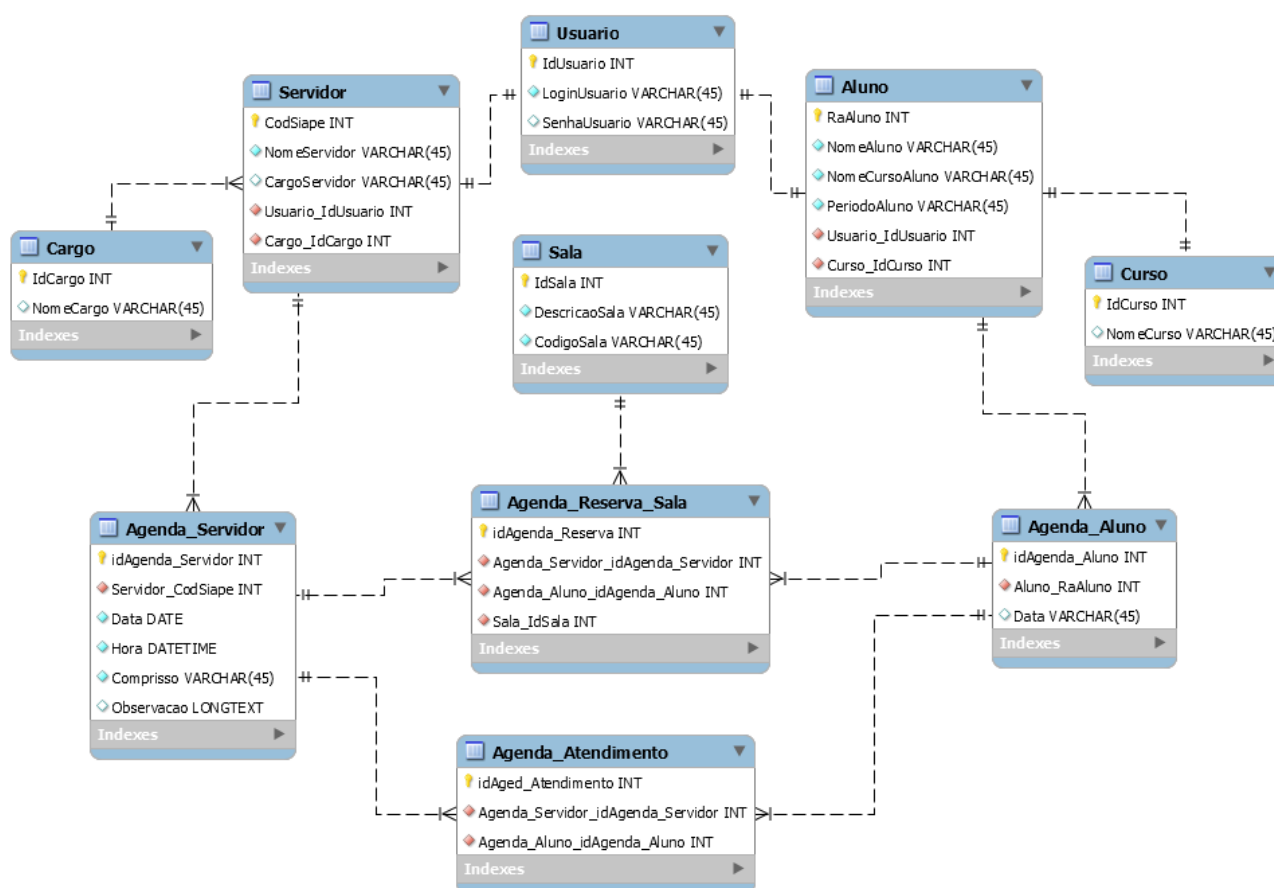


Figura 3. Diagrama EER criado no MySQL Workbench

### - Modelo Físico do Banco de Dados

O modelo físico do banco de dados será gerado através do Framework Java Spring Boot e para criar o código conforme as anotações requeridas foi usado a ferramenta JDL-Studio, segue abaixo o código fonte para implementação das entidades e relacionamentos do banco de dados para MySql com H2 e Spring Boot.

```
entity Cargo {  
    nomeCargo String  
}  
  
entity Sala{  
    nomeSala String,  
    codigoSala String  
}  
  
entity Aluno {  
    raAluno Double,  
    nomeAluno String,  
    periodo Double,  
    curso Curso  
}  
  
entity Servidor {  
    codSiape Double,  
    nomeServidor String  
}  
  
entity AgendaServidor{  
  
    status StatusAgenda,  
  
}  
  
entity AgendaAtendimentoServidor{  
    status StatusAgenda  
}
```

```
entity AgendaAluno{  
    status StatusAgenda  
}
```

```
entity AgendaSala{  
    status StatusAgenda,  
  
}
```

```
entity AgendaReservaSala{  
    status StatusAgenda  
  
}
```

```
entity DiasReservaSala{  
    status StatusDia,  
    horario Horario,  
    diaMes DiaMes,  
    diaSemana DiaSemana,  
    mes Mes  
}
```

```
entity DiasAtendimentoServidor{  
    mes Mes,  
    diaMes DiaMes,  
    diaSemana DiaSemana,  
    statusDia StatusDia
```

```
}
```

```
entity DiasAulaAluno{  
    semestre Semestre,  
    horario Horario,  
    diaMes DiaMes,  
    diaSemana DiaSemana  
}
```

```
enum Semestre{  
    Semestre  
}
```

```
enum StatusAgenda{  
    Livre, Ocupado, Atendido, Remarcado, Faltou, Agendado  
}
```

```
enum Mes{  
    Janeiro, Fevereiro, Marco, Abril, Maio, Junho, Julho,  
    Agosto, Setembro, Outubro, Novembro, Dezembro  
}
```

```
enum Curso {  
    Engenharia_Eletrica,      Engenharia_Computacao,      Engenharia_Eletronica,  
    Engenharia_Mecanica,      Engenharia_Control_e_Automacao,      Engenharia_Software,  
    Analise_Desenvolvimento_Sistemas  
}
```

```
enum DiaSemana {
```

```
Segunda_feira, Terca_feira, Quarta_feira, Quinta_feira, Sexta_feira  
}
```

```
enum DiaMes{  
D1,D2,D3,D4,D5,D6,D7,D8,D9,D10,D11,D12,D13,D14,D15,D16,D17,D18,D19,D20,D21,D22,D23,D24,D25,D26,D27,D28,D29,D30,D31  
}
```

```
enum StatusDia{  
Disponivel, Indisponivel  
}
```

```
enum Horario{  
H8,H9,H10,H11,H12,H13,H14,H15,H16,H17,H18H,H19,H20,H21,H22  
}
```

```
relationship OneToMany {  
Cargo{nomeCargo} to Servidor,  
Servidor{codSiape} to DiasAtendimentoServidor,  
DiasAtendimentoServidor{status} to AgendaServidor,  
DiasAulaAluno{sem} to AgendaAluno,  
AgendaServidor{status} to AgendaAtendimentoServidor,  
Aluno{raAluno} to AgendaAluno,  
Sala{codigoSala} to AgendaSala,  
AgendaSala{status} to AgendaReservaSala,  
Aluno{raAluno} to AgendaReservaSala,  
AgendaAluno{status} to AgendaAtendimentoServidor,  
DiasReservaSala{status} to AgendaReservaSala  
}
```

### - Diagrama de Classes

Segue abaixo o diagrama de classes gerado pela ferramenta JDL-Studio a partir do código fonte que gera o banco de dados.

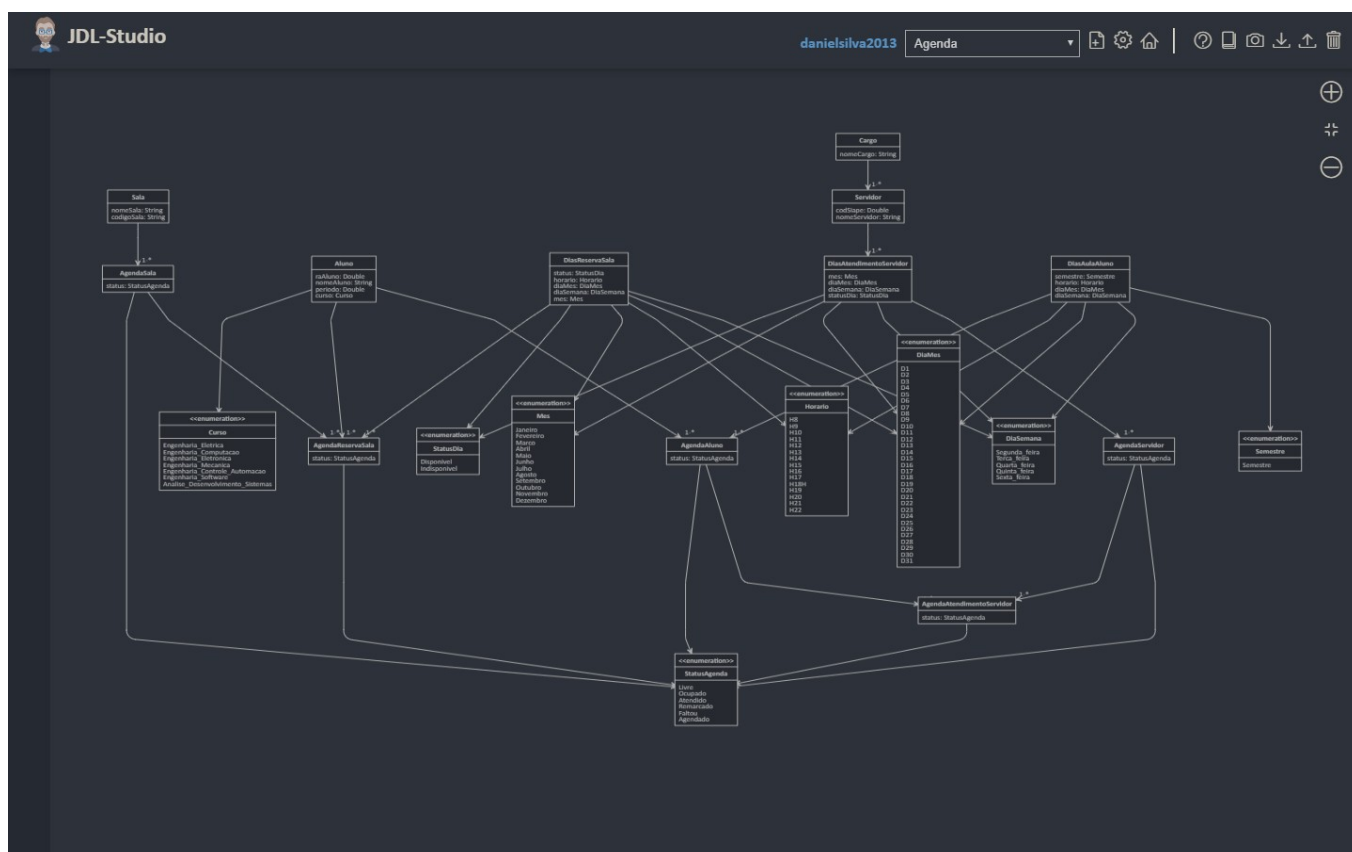


Figura 4. Diagrama de Classes

Dicionário de dados: Apresentar o dicionário de dados do banco de dados. Documentar cada tabela com seus atributos mostrando nome do atributo, tipo, tamanho, descrição, se é obrigatório ou não, e o que mais for necessário para descrever os dados. Documentar também usuários, stored procedures, funções e qualquer outra implementação ligada ao banco de dados.

### - Diagrama de Atividades

Os diagramas de atividade foram criados com o auxílio da ferramenta Astah

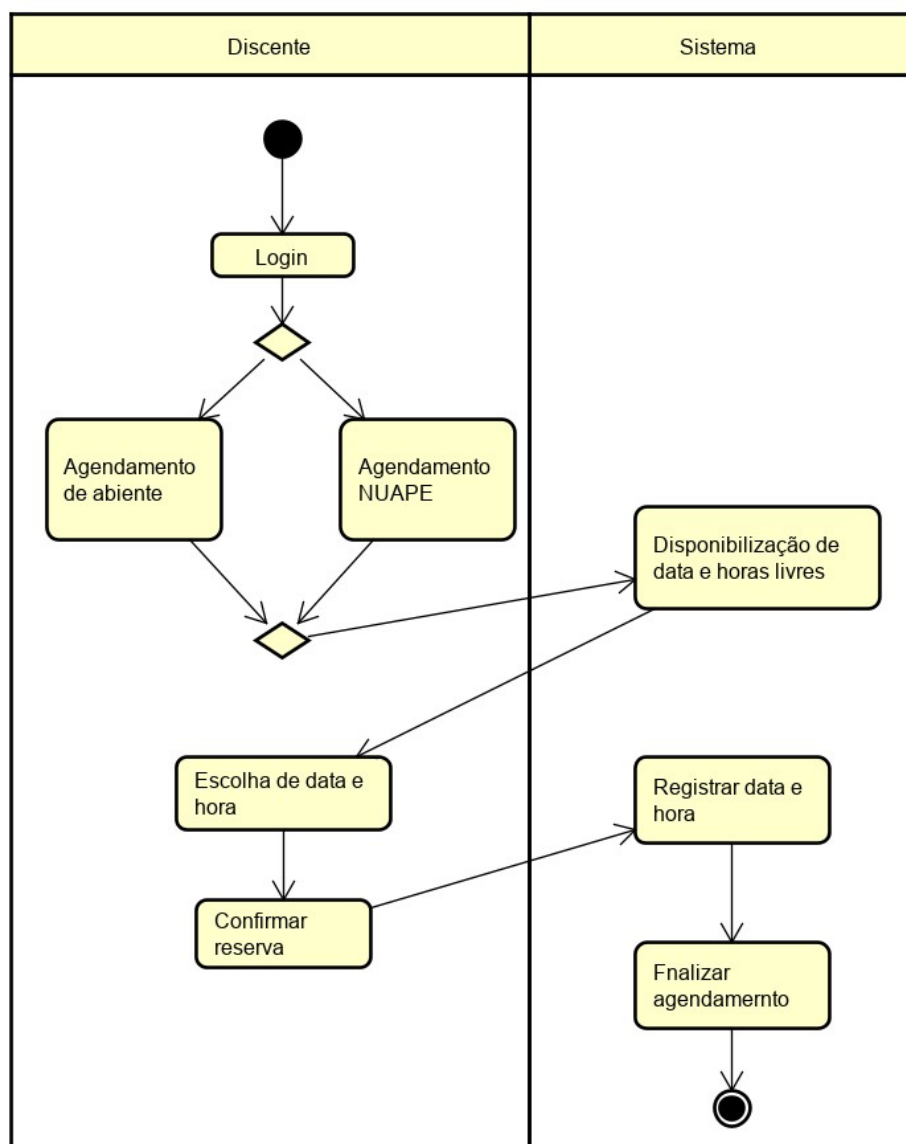


figura 5. Diagrama de Sequência geral para o usuario Discente

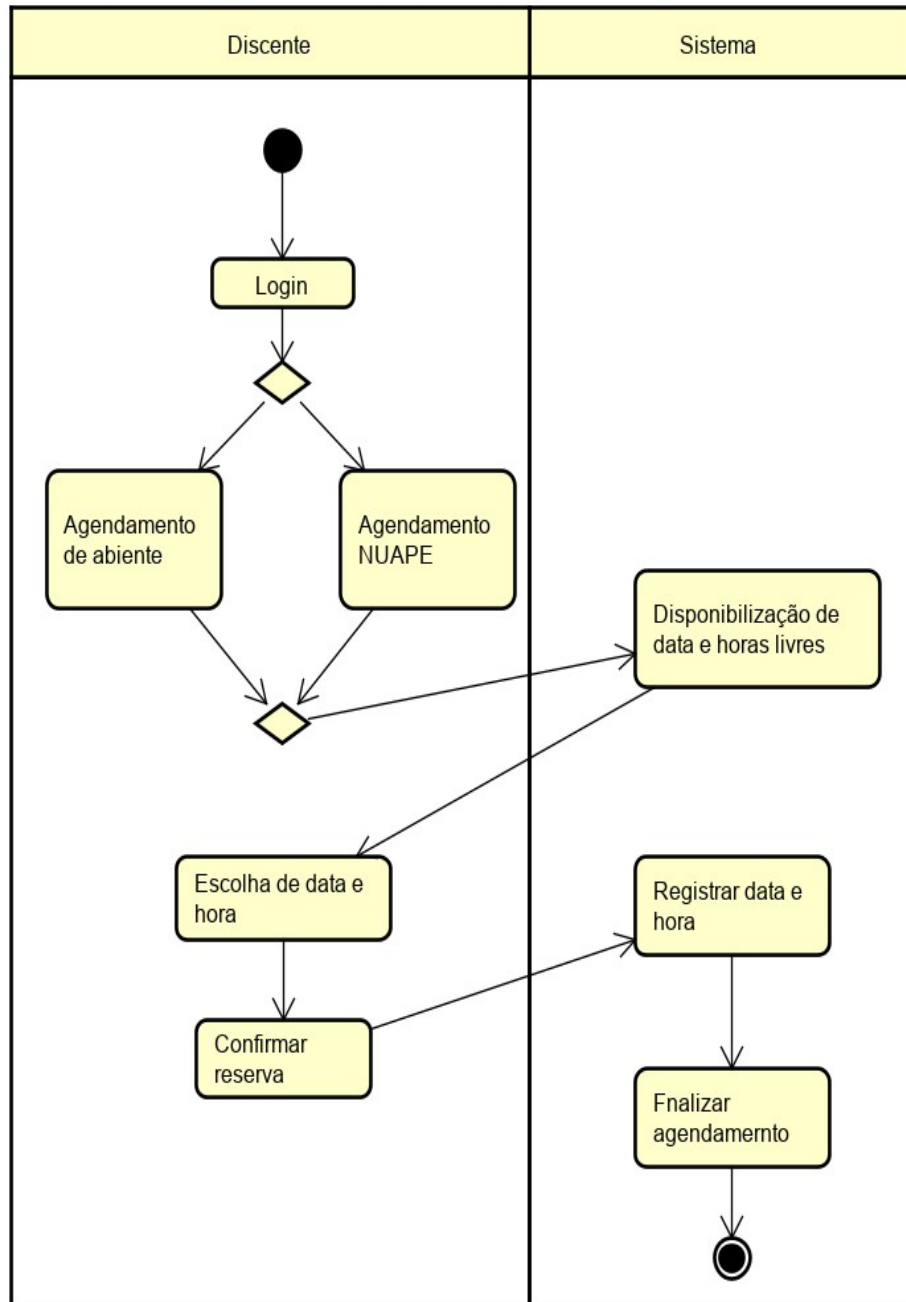




figura 6. Diagrama de Sequência geral para o usuário Docente.

## **6 Implementação**

### **6.1 Protótipos de Telas**

Apresentar o protótipo do sistema, que consiste na interface preliminar contendo um conjunto de funcionalidades e telas. O protótipo é um recurso que deve ser adotado como estratégia para levantamento, detalhamento, validação de requisitos e modelagem de interface com o usuário (usabilidade).

As telas do sistema podem ser criadas na própria linguagem de desenvolvimento ou em qualquer outra ferramenta de desenho. Cada tela deve possuir uma descrição do seu funcionamento, constando pelo menos o objetivo da tela e dinâmica de navegação (de onde é chamada e que outras telas pode chamar). A descrição das telas deve registrar informações que possam ser consultadas para facilitar a implementação e a execução de testes, assim como a que requisitos funcionais se referem.

Se os protótipos de tela foram usados nas fases anteriores, esse item deve ser preenchido conforme os protótipos forem desenvolvidos e concluído com a versão final dos protótipos.

### **6.2 Descrição do código**

Descrever o sistema quanto ao código gerado. Explicar a organização dos arquivos, pacotes, classes ou quaisquer estruturas utilizadas no desenvolvimento do projeto, listando os componentes criados e sua estrutura. Use diagramas (Diagrama de Componentes, Diagrama de Pacotes) para ilustrar a implementação.

Descrever também convenções e padronizações para comentários no código, nomenclatura de classes, objetos, funções, etc. Se necessário, use exemplos.

### **6.3 Testes**

Descrever os procedimentos e métricas empregados para a avaliação e validação do software desenvolvido.



## 7 Considerações Finais

Apresentar e discutir os resultados obtidos e sua aplicabilidade. Abordar o que foi atingido e o que não foi, as limitações, possíveis integrações com outros projetos e continuação do sistema em trabalhos futuros.

## 8 Bibliografia

Apresentar todas as obras (livros, artigos, Internet, revistas, etc...) utilizadas na elaboração da documentação e na implementação do projeto.

[https://www.camara.leg.br/proposicoesWeb/prop\\_mostrarintegra?codteor=867726](https://www.camara.leg.br/proposicoesWeb/prop_mostrarintegra?codteor=867726)

[https://www.google.com/search?rlz=1C1AVFC\\_enBR794BR798&ei=ZthnXY71O-O65OUPzd-niAo&q=significado+de+universidade&oq=significado+de+universidade&gs\\_l=psy-ab.3..0i70i249j0l2j0i22i30l7.33871.44979..45264...1.2..4.184.4678.0j32.....0....1..gws-wiz.....10..0i71j0i10j35i39j0i67j0i131j35i39i70i249j0i22i10i30.GjI\\_p9tb7mk&ved=0ahUKewiOwKXunKjKAhVjHbkGHc3vCaEQ4dUDCAo&uact=5](https://www.google.com/search?rlz=1C1AVFC_enBR794BR798&ei=ZthnXY71O-O65OUPzd-niAo&q=significado+de+universidade&oq=significado+de+universidade&gs_l=psy-ab.3..0i70i249j0l2j0i22i30l7.33871.44979..45264...1.2..4.184.4678.0j32.....0....1..gws-wiz.....10..0i71j0i10j35i39j0i67j0i131j35i39i70i249j0i22i10i30.GjI_p9tb7mk&ved=0ahUKewiOwKXunKjKAhVjHbkGHc3vCaEQ4dUDCAo&uact=5)

exemplo1

[http://moodle.utfpr.edu.br/pluginfile.php/682882/mod\\_resource/content/2/01%20-%20Exemplo\\_01\\_Documento\\_de\\_Projeto.pdf](http://moodle.utfpr.edu.br/pluginfile.php/682882/mod_resource/content/2/01%20-%20Exemplo_01_Documento_de_Projeto.pdf)

exemplo 2

[http://moodle.utfpr.edu.br/pluginfile.php/682885/mod\\_resource/content/1/01%20-%20Exemplo\\_02\\_Documento\\_de\\_Projeto.pdf](http://moodle.utfpr.edu.br/pluginfile.php/682885/mod_resource/content/1/01%20-%20Exemplo_02_Documento_de_Projeto.pdf)

Apresentação ( n colocar todos os requisito, só os principais. Precisam ser objetivos). Ele disse q vai ser rigoroso, mas pra gente melhorar