

DOCUMENTO DE SOFTWARE

<SalaMaps>

VERSÃO <1.0.01>

SUMÁRIO

1. Introdução	2
1.1. Objetivos deste documento	2
1.2. Definições e siglas	
1.3. Problema	
1.4. Objetivo	
1.5. Definições e siglas	2

2. Formulário de Entrevista	2
3. Requisitos	3
4. Protótipo	3
4.1. Tela Principal	3
4.2. Tela de <<outras telas>>	4
5. Cenários de Casos de Uso	4
6. Diagrama de Casos de Uso	6
7. Diagrama de Classes	6
8. Diagrama de Atividades	6
9. Diagrama de Sequência	6
10. Diagrama Entidade Relacionamento	6
11. Diagrama Modelo Relacional	6
12. Modelo Físico SQL	19

1. Introdução

1.1. Tema

Sistema de direcionamento de salas, voltado para o auxílio na distribuição de salas para novos alunos que iniciarão suas atividades no campus UNA Cristiano Machado, podendo futuramente ser implementado nos demais campi.

1.2. Justificativa

A dificuldade dos alunos de localizar suas salas de aula e a presença de problemas e mudanças de última hora dificultam que os alunos se localizem corretamente e que as aulas possam ser iniciadas no horário correto, o sistema será implementado de maneira que quaisquer mudanças no ensalamento serão notificadas automaticamente no momento da alteração.

1.3. Problema

Com o grande fluxo de aulas presentes no campus o ensalamento pode sofrer mudanças de última hora e causar transtornos na movimentação e localização do corpo discente.

1.4. Objetivo

Informar o ensalamento para os alunos em cada dia de aula específica de maneira que os mesmos não tenham dificuldade e consigam se movimentar de maneira objetiva de modo que os corredores não fiquem lotados e os professores possam iniciar a aula no horário correto.

1.5. Definições e siglas

Caso o seu trabalho utilize alguma abreviação ou sigla, especificar aqui.

Número	Sigla	Definição

2. Formulário de Entrevista

Objetivo da Entrevista

- Entender a dificuldade dos alunos e professores para localizar as salas

Entrevistado(s)

- Rômulo Cintra e João Henrique

Perguntas e respostas

01) O que você precisa para localizar as salas?

Para localizar as salas de aula precisamos saber a numeração e disposição das salas de aula e se a informação que possuímos está correta.

02)Qual a dificuldade de encontrar as salas?

A disposição das salas do campus CM é de numeração em bloco e isso pode confundir facilmente pessoas desavisadas.

03)O que facilitaria o seu uso do aplicativo?

Facilitaria se eu pudesse adaptar o aplicativo para meu uso e que ele não fosse totalmente dependente de internet, para poder o acessar em lugares sem sinal.

04)Qual solução você enxerga para uma possível falha de localização das salas no aplicativo?

A solução que vem à mente é a criação de um canal de comunicação entre os alunos e a faculdade, onde os professores poderiam mandar mensagens diretamente aos alunos ou onde a coordenação pudesse informar as localizações e horários de atendimento dos setores de ajuda.

Observações

3. Requisitos

Requisitos Funcionais

RF01 - O usuário deve se conectar com seu login e senha

RF02 - O coordenador de curso deve conseguir cadastrar e alterar salas de aula

RF03 - Professores devem conseguir fazer comunicados através do sistema

RF04 - Deve possuir um mapa do campus com a disposição das salas de aula

RF05 - Deve diferenciar salas de aula normais de laboratórios e salas “especiais”(Reunião, GELP, CAC, NSI, etc...)

RF06 - Deve emitir uma notificação sempre que a sala marcada para o dia for alterada

RF07 - Deve ser personalizável o programa

RF08 - Deve disponibilizar o horário de atendimento dos setores (Ex: CAC - 09:00~19:00)

RF09 - Parte do sistema deve operar de maneira offline

RF10 - Pesquisa por voz para usuários com dificuldade de visão/leitura

Requisitos Não Funcionais

RNF01 - Cada um tem o seu registro (Requisito de produto)

RNF02 - Dois modos de edição (professor\aluno) (Requisito de produto)


RNF03 - Capacidade para uma quantidade elevada de usuários (Requisito de eficiência)

RNF04 - Deve ser capaz de emitir o mapa do campus com fluidez e oferecer recurso como zoom e rotação sem perda de qualidade (Requisito externo)

RNF05 - Ele deve ser multiplataforma como IOS, android e PC (Requisito de portabilidade)

4. Protótipo

4.1. Tela Principal



☐

Aluno Professor

RA ou CPF

SENHA [Show](#)

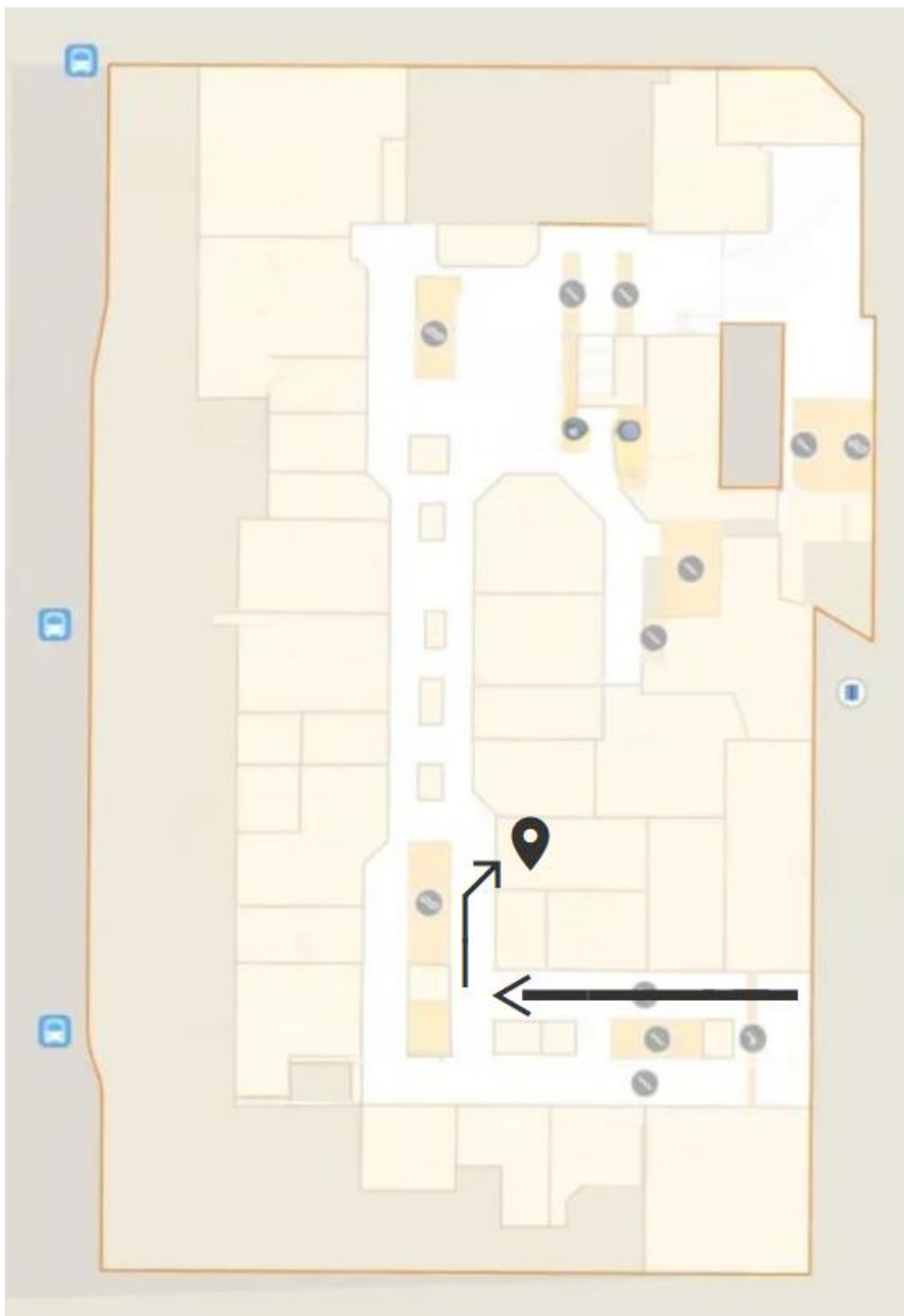
[Esqueceu a senha?](#)

Entrar

[Precisa de Ajuda?](#)

4.2. Tela de <<outras telas>>

1. Mapa



2. Usuário logado



1 Nova Mensagem
do professor



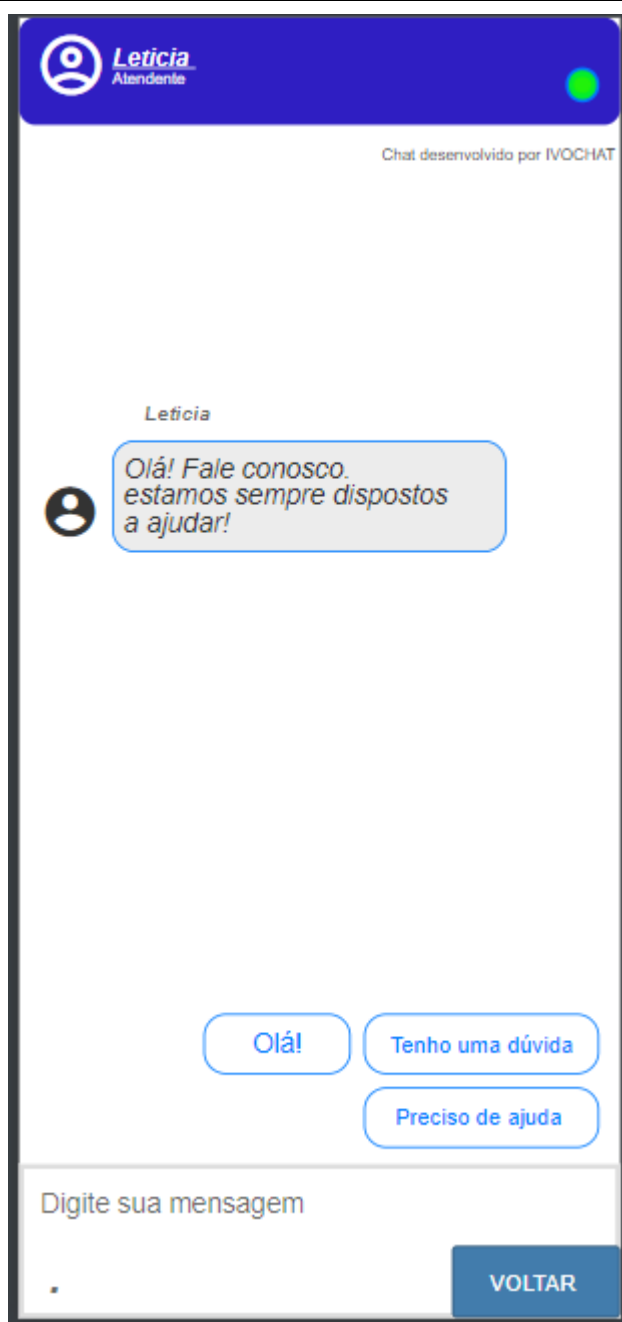
Quinta 30 de Março de 2023

A aula de hoje será na sala:

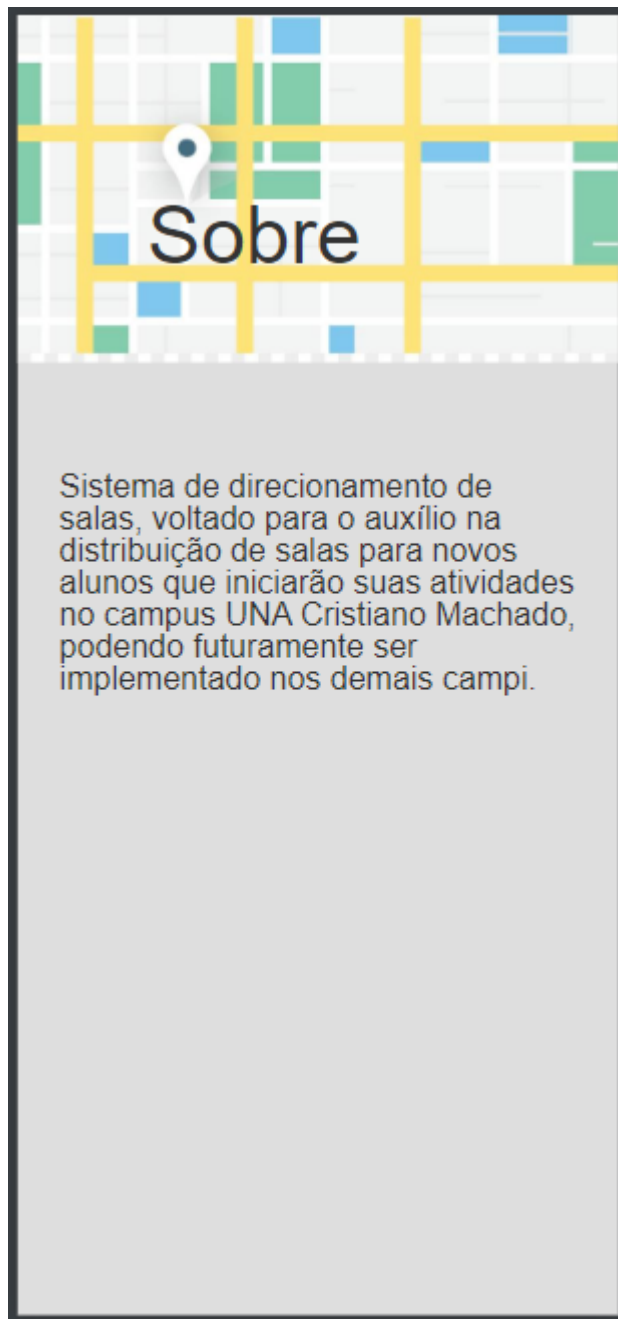
Cristiano Machado - UNA -
Laboratório De Informática -
Único - 4º Andar - 407

[Veja no mapa](#)

3. Ajuda



4. Sobre



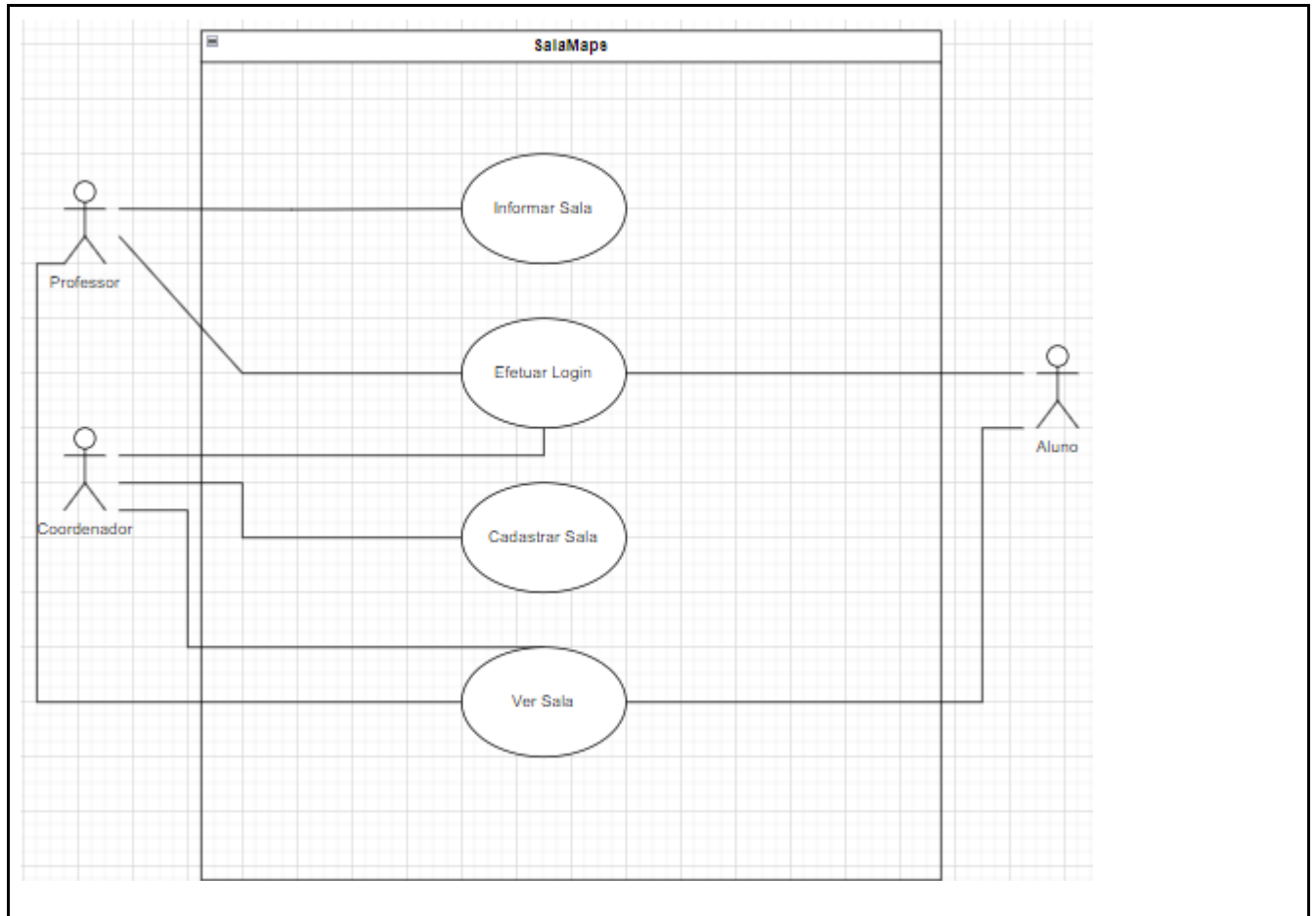
5. Cenários de Casos de Uso

Nome	SalaMaps
Identificador	Identificar salas
Sumário	Auxílio de alunos e professores para designação de salas
Ator primário	Alunos e professores
Ator secundário	Sistema da Una
Pré-condição	Registro
Fluxo Principal	1 - Abrir aplicativo. 2 - Informar o registro do aluno ou professor. 3- Consultar disponibilidade de sala (Professor). 4 - Informar no aplicativo a sala disponível para os alunos. (Professor).
Fluxos Alternativos	1 - Se não achar sala disponível consultar coordenação.
Fluxos de Exceções	Sala inválida
Pós- condição	

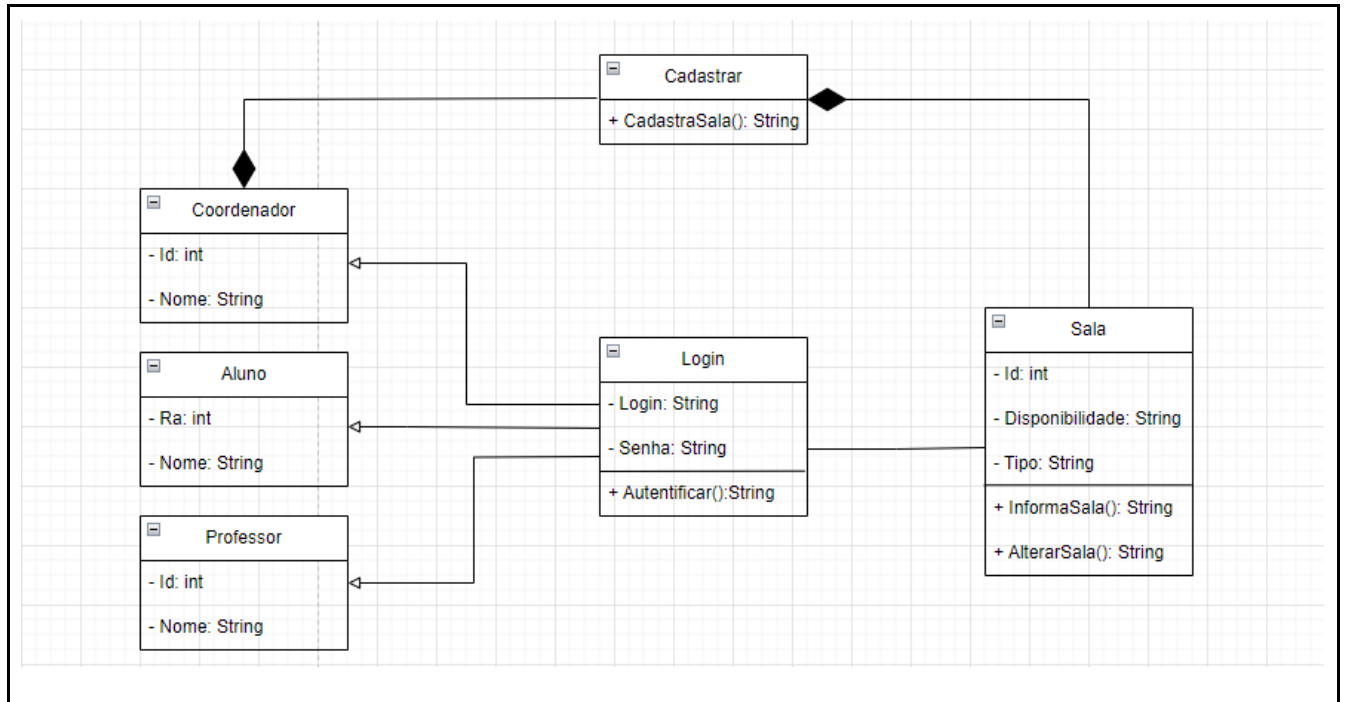
Nome	Salas
Identificador	Registrar sala de aula
Sumário	Lugar onde será registrado a sala de aula

Ator primário	Sistema da Una
Ator secundário	Professores
Pré-condição	Número de registro
Fluxo Principal	1 - Consultar o sistema 2 - Atribuir o número de registro a uma sala 3 - Finalizar registro
Fluxos Alternativos	1 - Sala com registro já existente
Fluxos de Exceções	Registro invalido
Pós- condição	

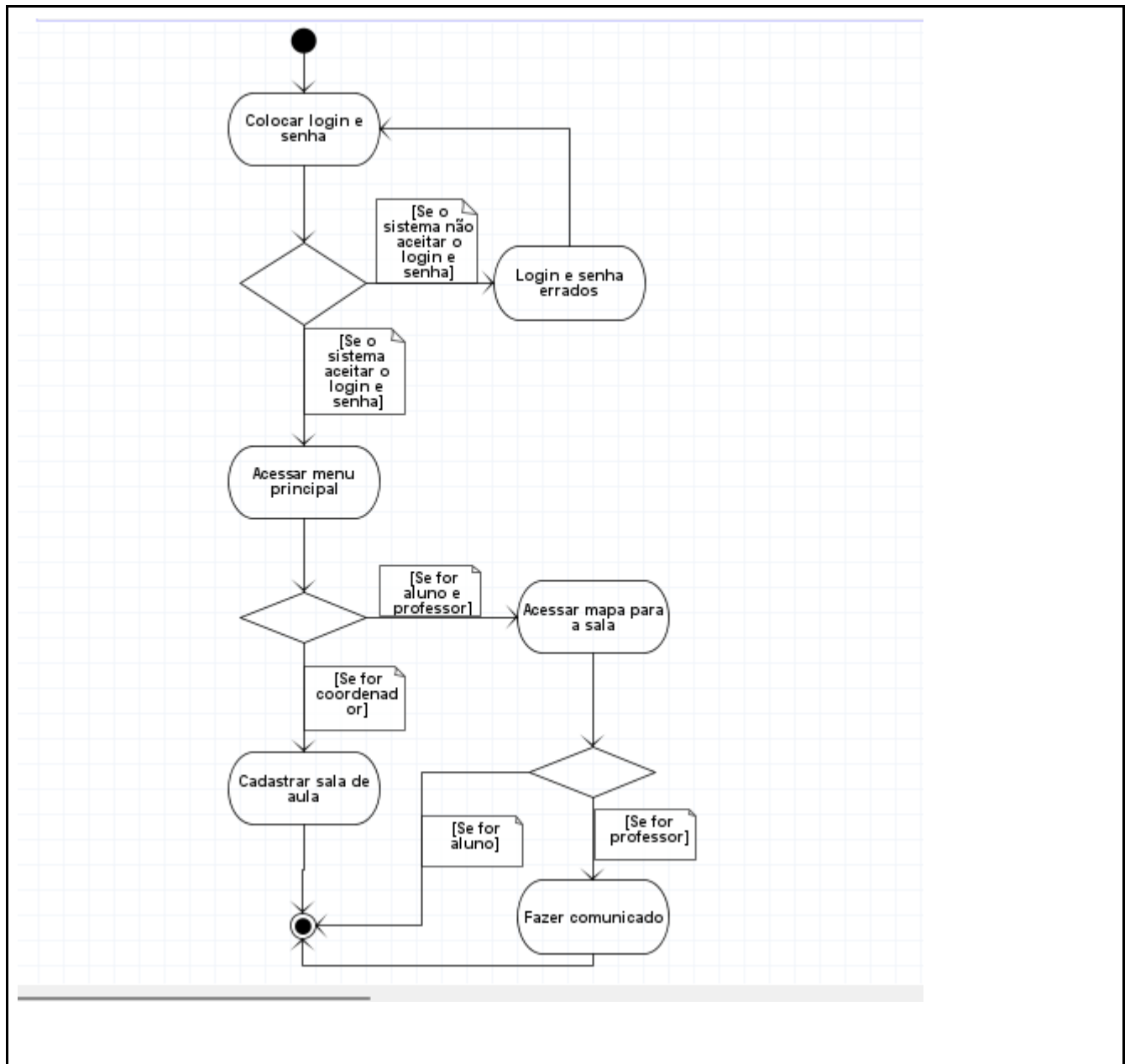
6. Diagrama de Casos De Uso



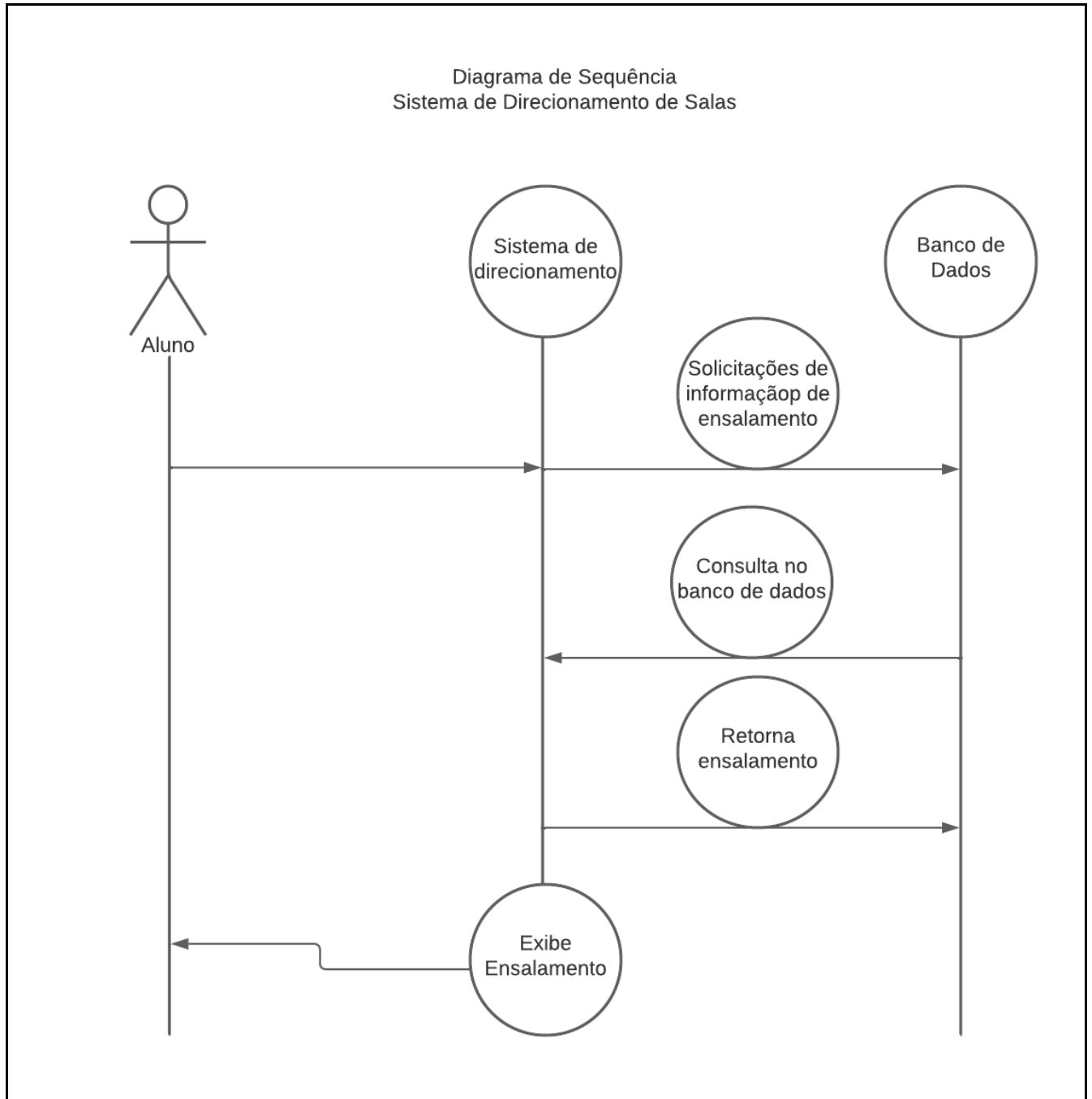
7. Diagrama de Classes



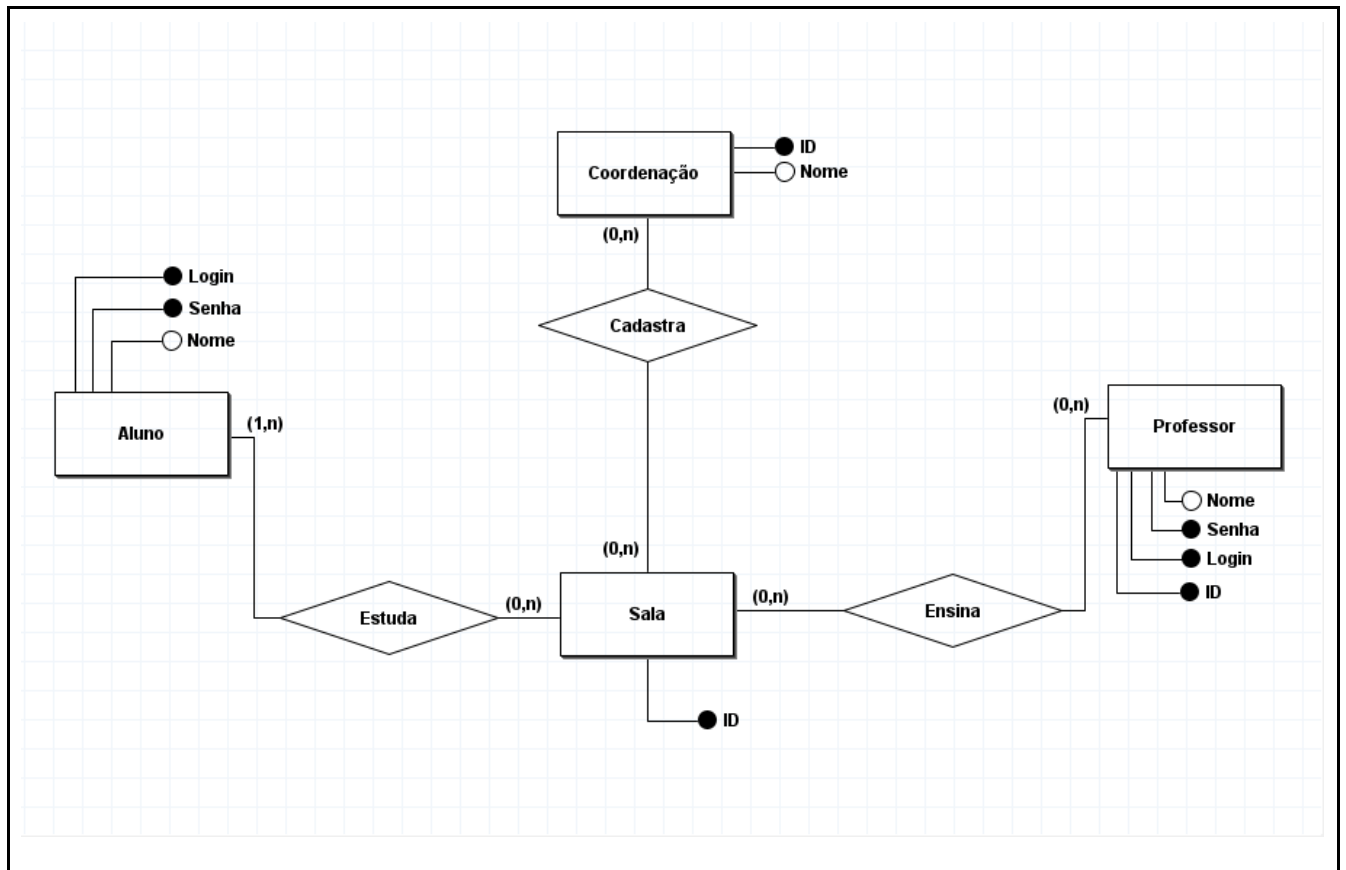
8. Diagrama de Atividades



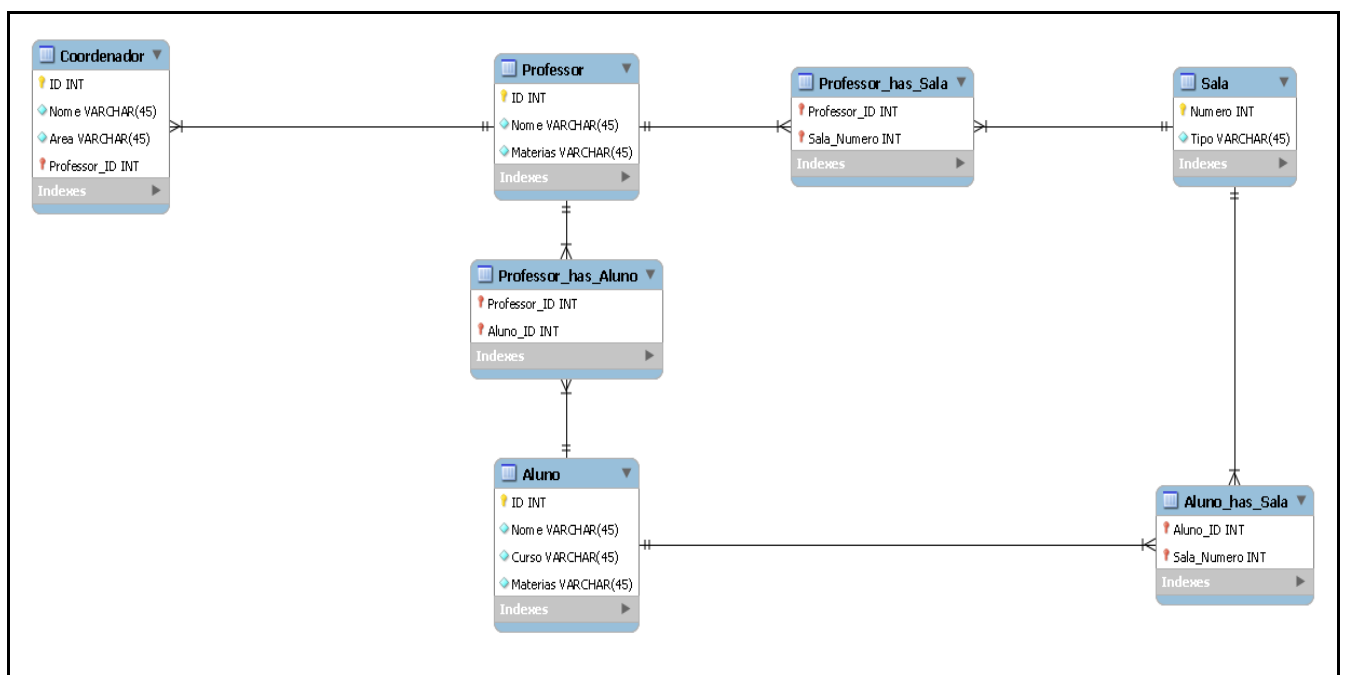
9. Diagrama de Sequência



10. Diagrama Entidade Relacionamento



11. Diagrama Modelo Relacional



12. Modelo Físico SQL

```
create database if not exists campus;  
use campus;
```

```
create table if not exists Aluno(  
id INTEGER NOT NULL,  
nome VARCHAR(50) NOT NULL,  
curso VARCHAR(50) NOT NULL,  
materias VARCHAR(50) NOT NULL,  
PRIMARY KEY(id)  
);
```

```
create table if not exists professor(  
id    INTEGER NOT NULL,  
nome VARCHAR(50) NOT NULL,  
curso VARCHAR(50) NOT NULL,  
materias VARCHAR(50) NOT NULL,  
PRIMARY KEY(id)  
);
```

```
create table if not exists sala(  
numero INTEGER NOT NULL,  
PRIMARY KEY(numero)
```

```
);
```

```
create table if not exists aula(  
id_aula INTEGER NOT NULL,  
nome VARCHAR(50) NOT NULL,  
horario VARCHAR(50) NOT NULL,  
dia DATE NOT NULL,  
PRIMARY KEY(ID)
```

```
);
```

```
create table if not exists da_aula(  
PRIMARY KEY (id_professor, id_aluno),  
FOREIGN KEY (id_professor) REFERENCES professor(id),  
FOREIGN KEY (id_aluno) REFERENCES aluno(id)  
)
```