

**Universidade Federal do Maranhão  
Centro de Ciências Exatas e Tecnologia  
Ciência da Computação  
Laboratório de Engenharia de Software  
Geraldo Braz Junior**

**Eduardo Silva Vieira**

**Coordenação Fácil  
Especificação da Arquitetura Proposta**

**Este trabalho tem como objetivo a  
complementação da segunda nota de  
Laboratório de Engenharia de  
Software.**

**São Luís - MA  
2018**

## 1. Introdução

Este trabalho tem como objetivo especificar e descrever a arquitetura usada para desenvolver a Coordenação Fácil, um sistema de gerenciamento de processos acadêmicos baseado em uma arquitetura orientada a serviços (REST).

## 2. Arquitetura

Este trabalho usou arquitetura orientada a serviços REST para modelar os três módulos pretendidos (Plano de Estudos, Aproveitamento de Cadeiras e Monitoria).

### 2.1 Diagrama de Classes

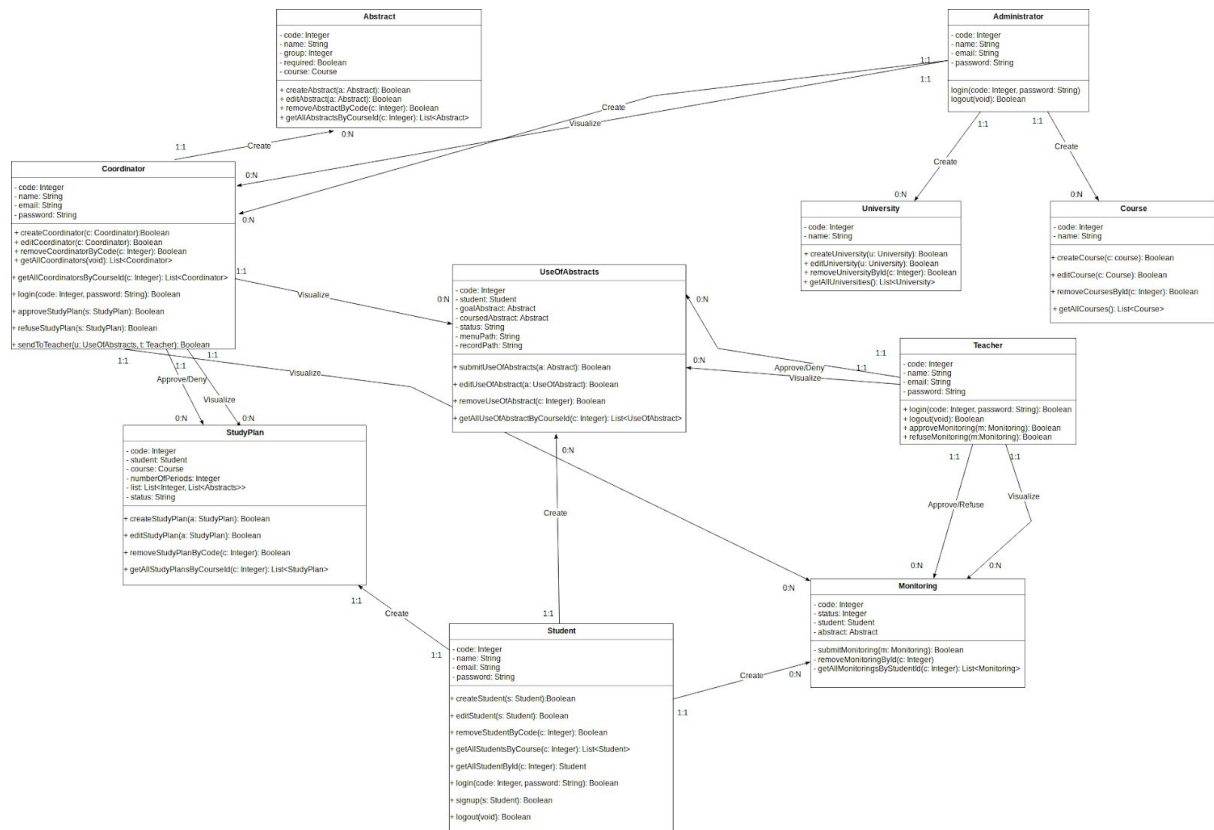


Figura 1 - Diagrama de Classes

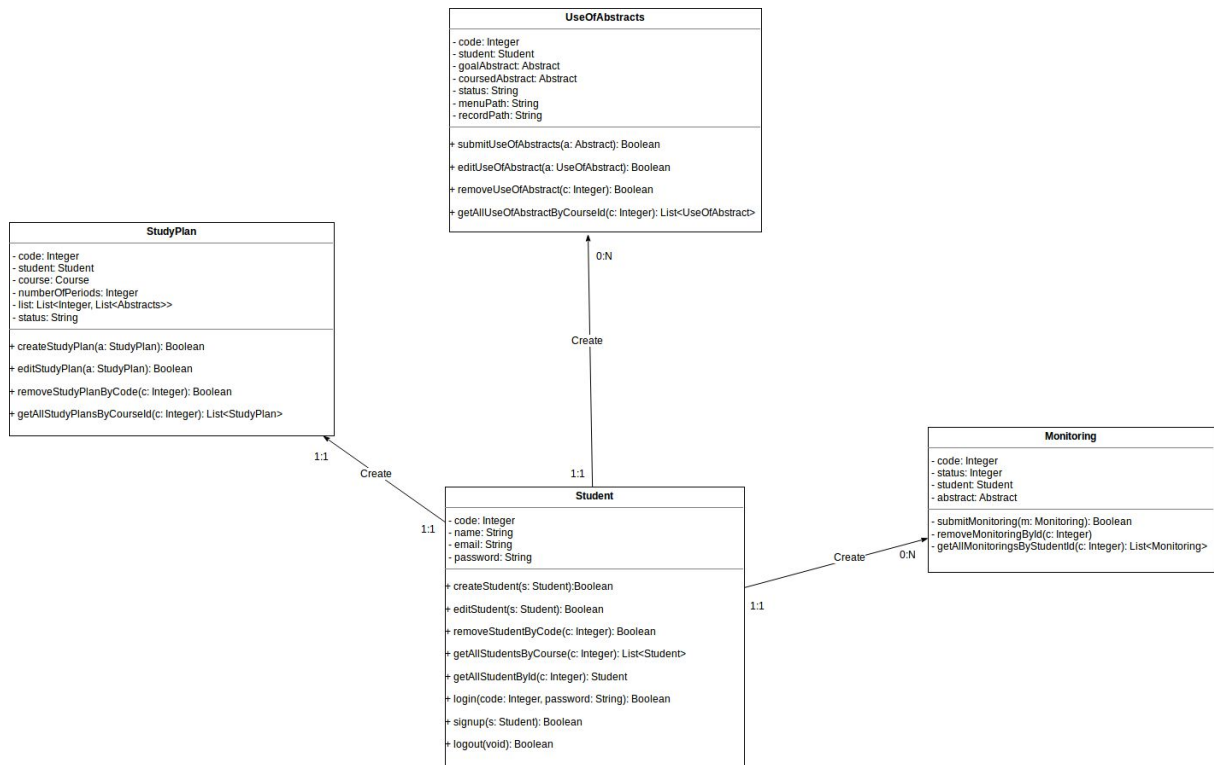


Figura 2 - Diagrama de Classes de estudante

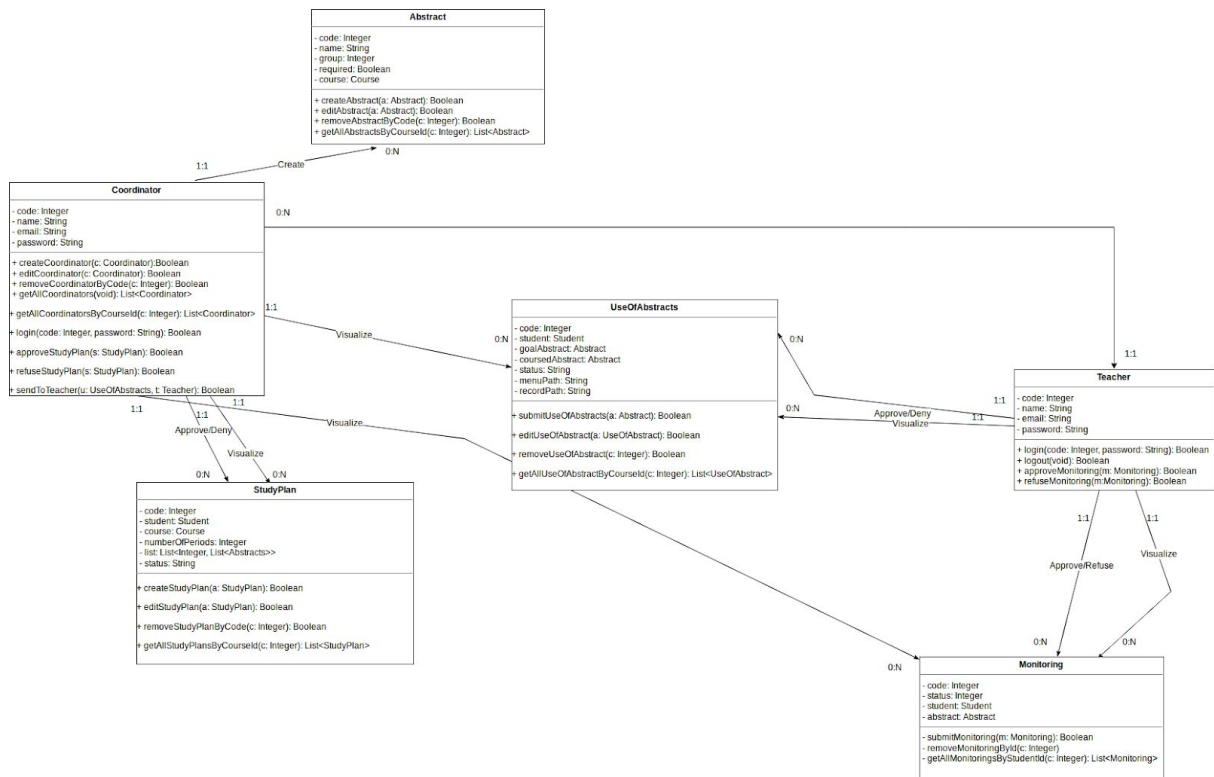


Figura 3 - Diagrama de Classes de Coordenador

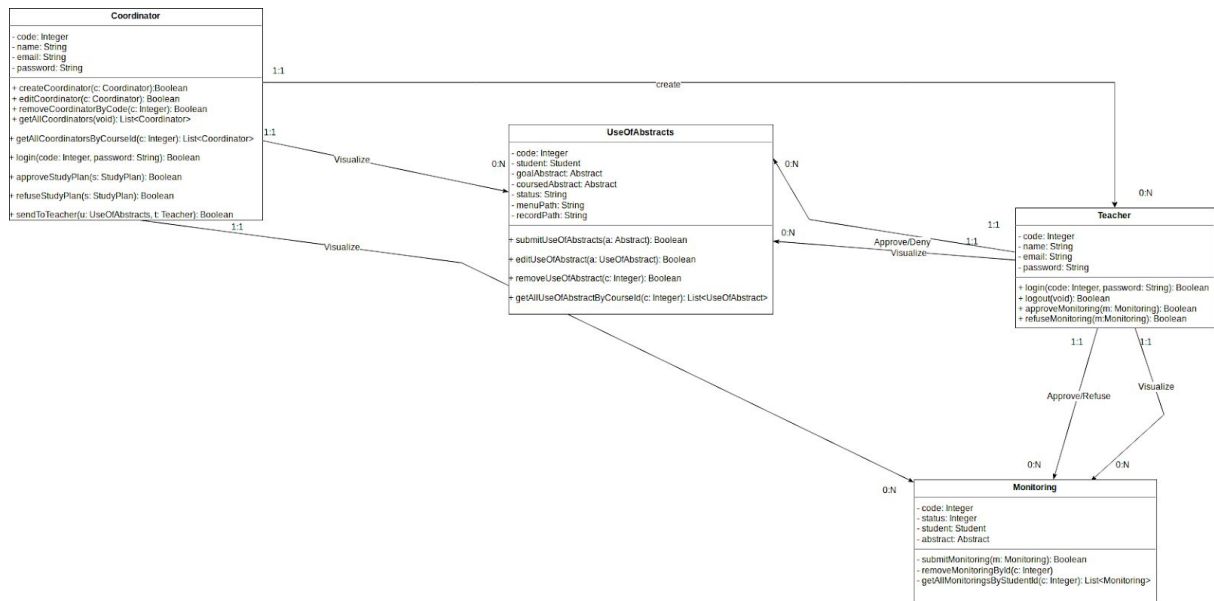


Figura 4 - Diagrama de Classes de Professor

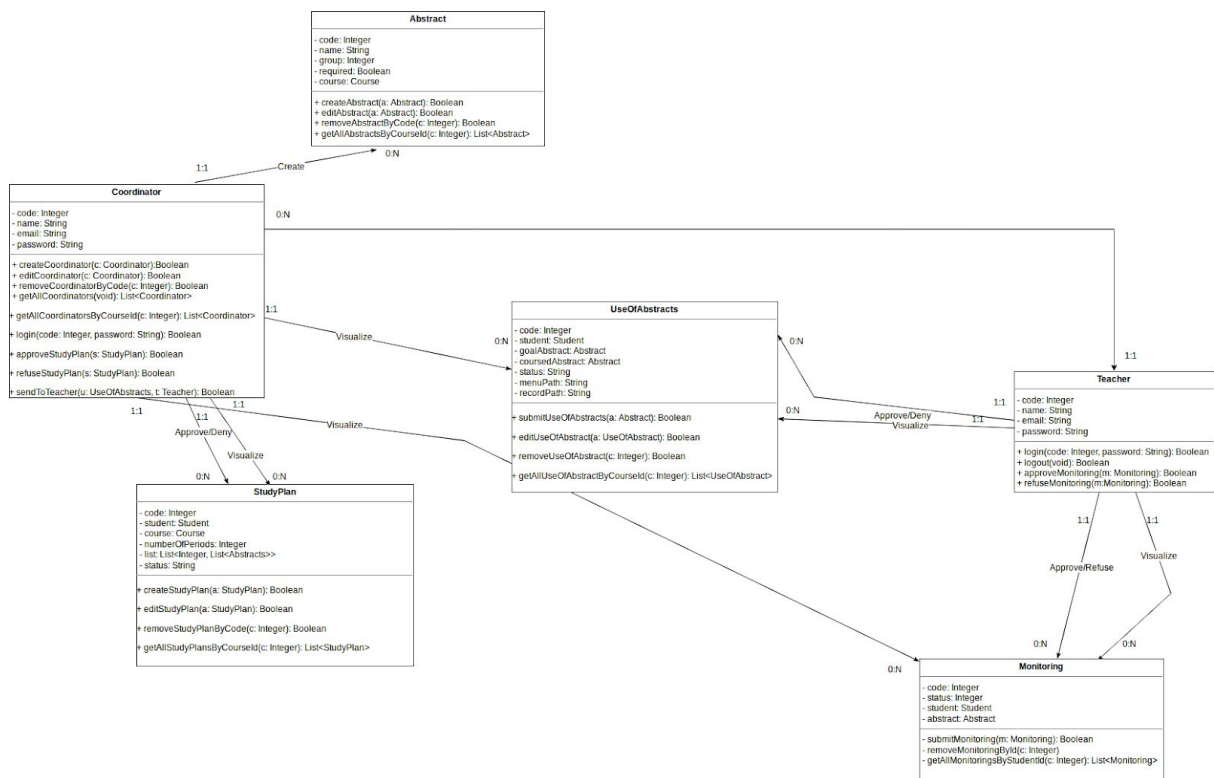


Figura 5 - Diagrama de Classes de Administrador

## 2.2 Diagrama de Entidade Relacionamento

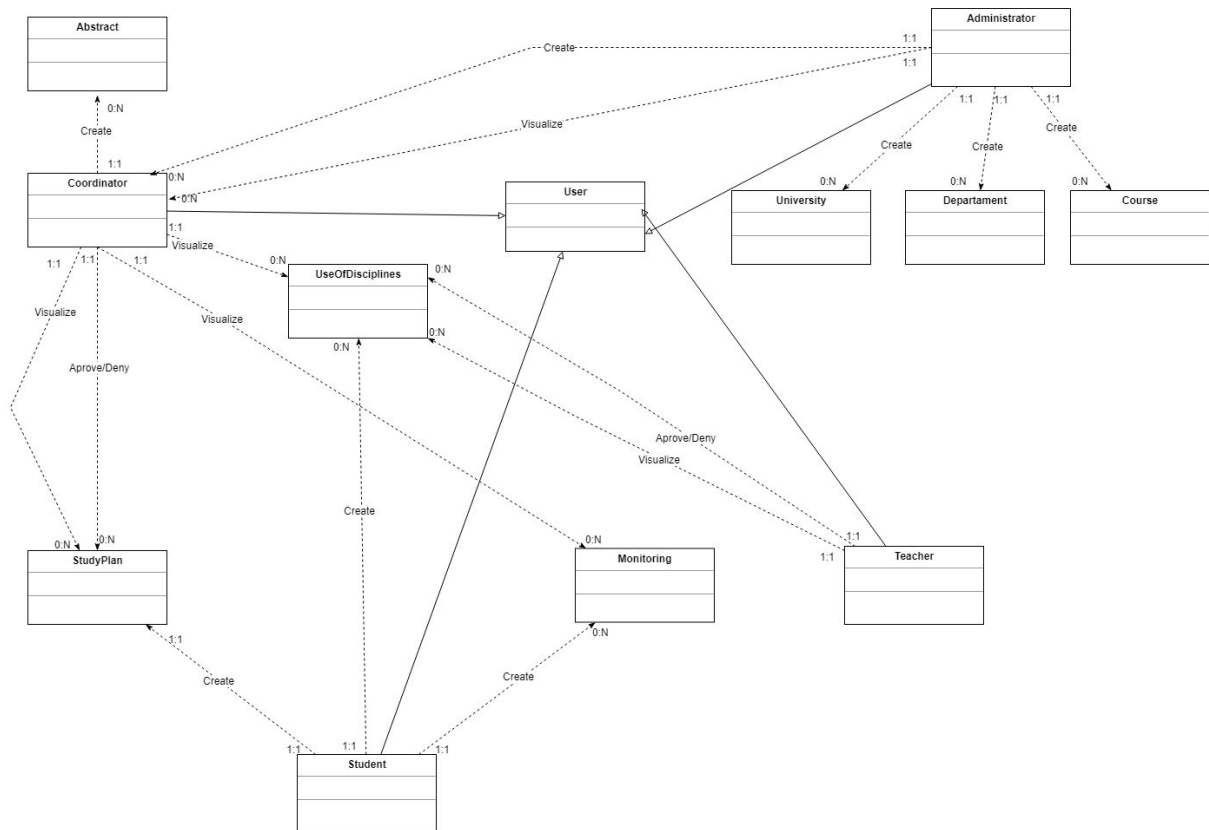


Figura 6 - Diagrama de Entidade Relacionamento

## 2.3 Diagramas MoLIC

MoLIC (*Modeling Language for Interaction as Conversation*) é uma linguagem desenvolvida por Paula et al. (2015). Criada para representar a interação, ou conversa, que o usuário tem com o sistema para atingir seus objetivos. Ela é composta de quatro artefatos: diagrama de objetivos, esquema de sinais conceituais, diagrama de interação e a especificação de sinais situacionais. De forma simplificada, o diagrama de objetivos indica o que o usuário deve fazer com a aplicação, o esquema de sinais conceituais define e organiza os conceitos envolvidos no sistema, o diagrama de interação representa como os objetivos serão alcançados durante a interação e a especificação de sinais situacionais apresenta detalhes dos sinais usados no diagrama de interação. O principal artefato da linguagem MoLIC é o diagrama de interação, onde é diretamente representado a comunicação do usuário com o sistema, ou preposto do designer.

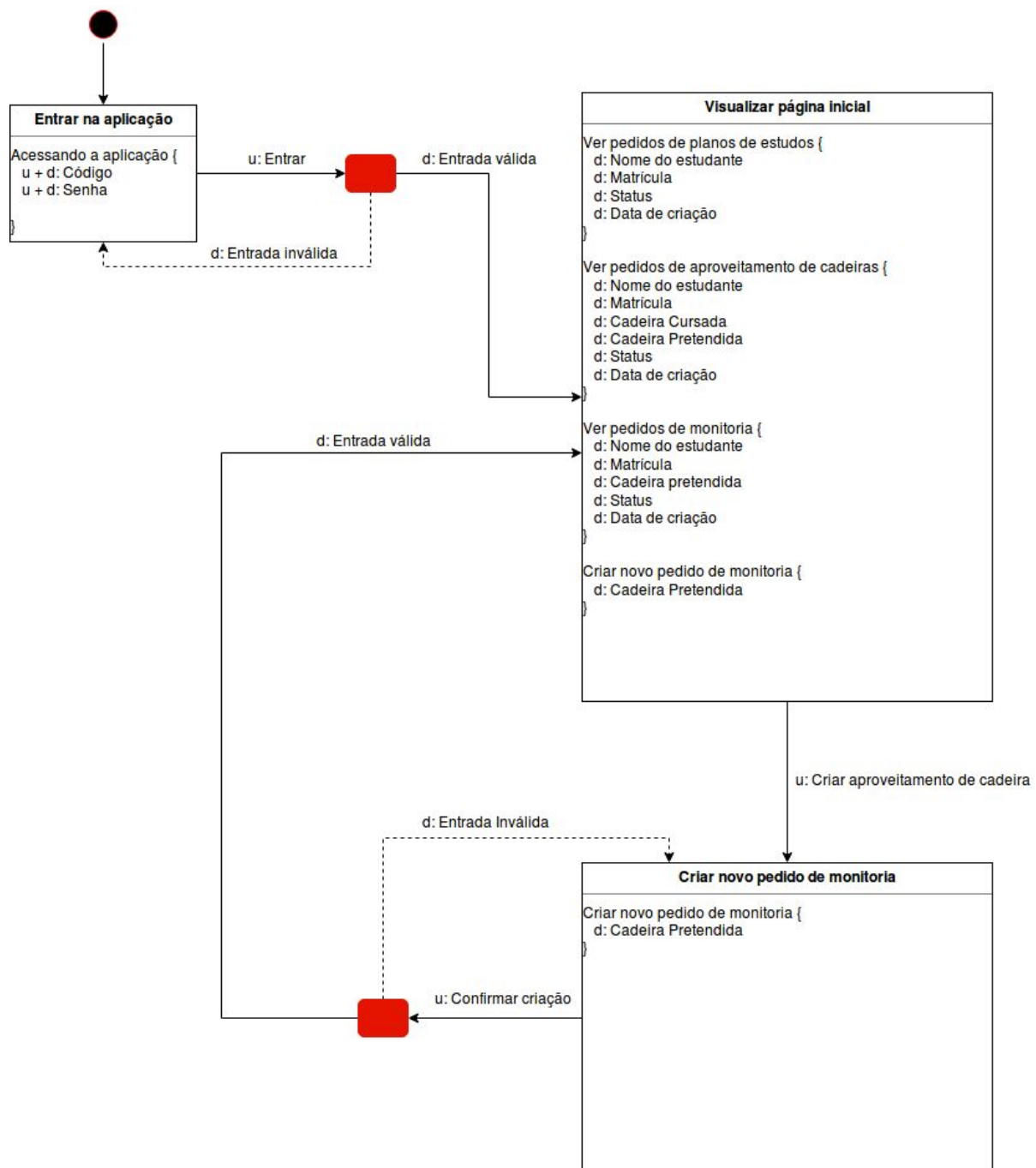


Figura 7 - Diagrama MoLIC - Monitoria

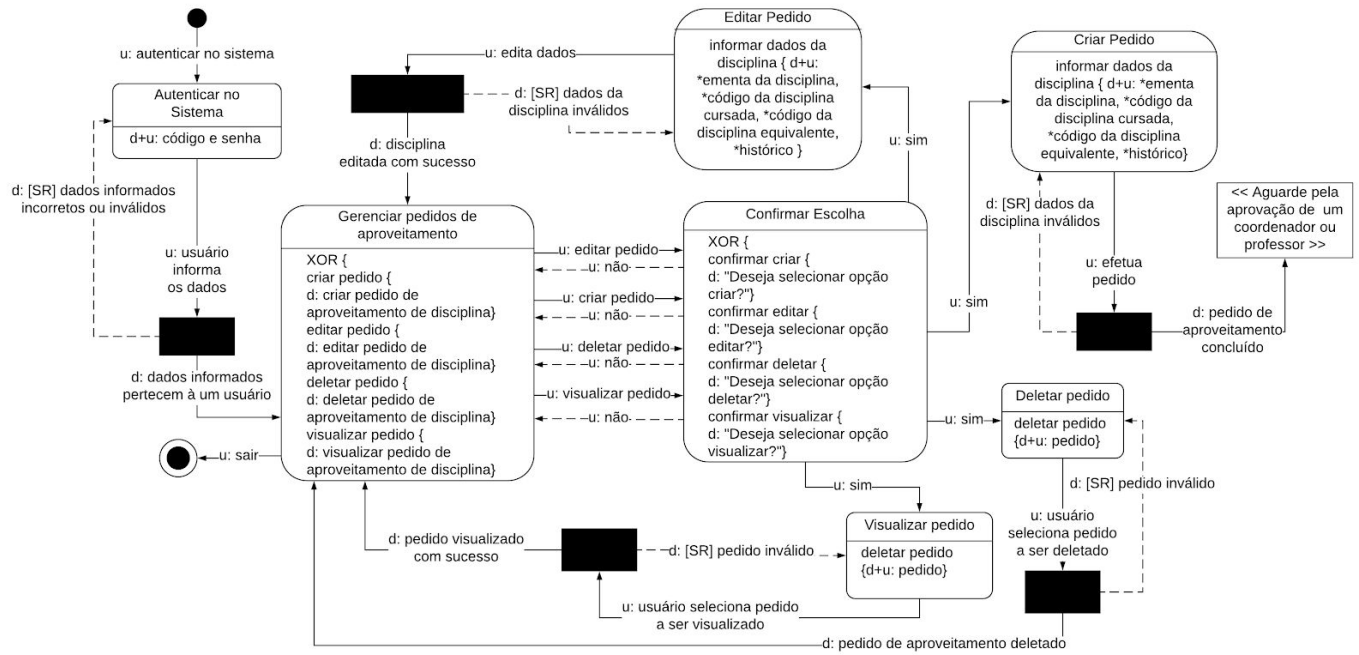


Figura 8 - Diagrama MoLIC - Plano de Estudos

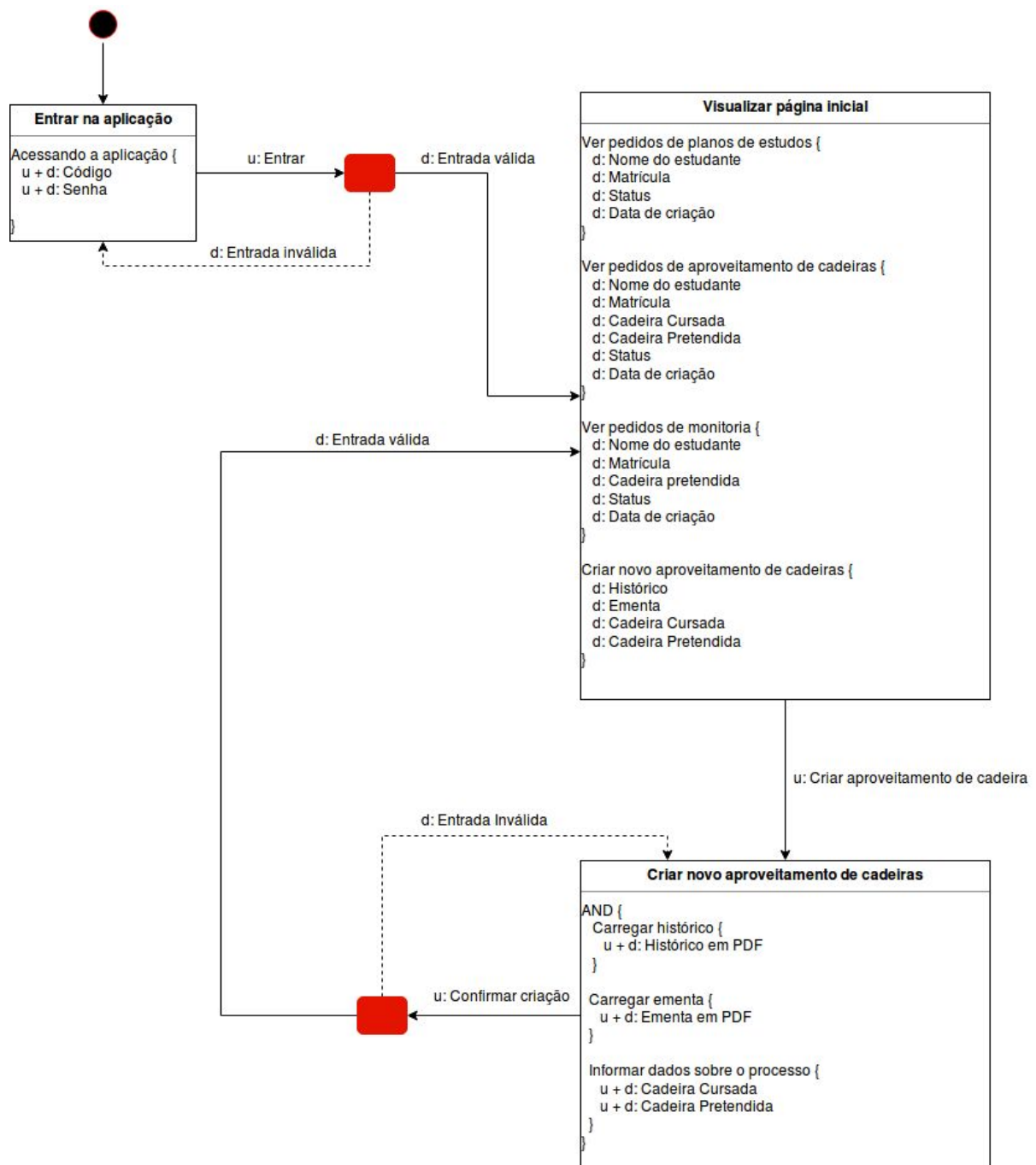


Figura 9 - Diagrama MoLIC - Aproveitamento de Cadeiras

## 2.4 - Diagrama de Atividade



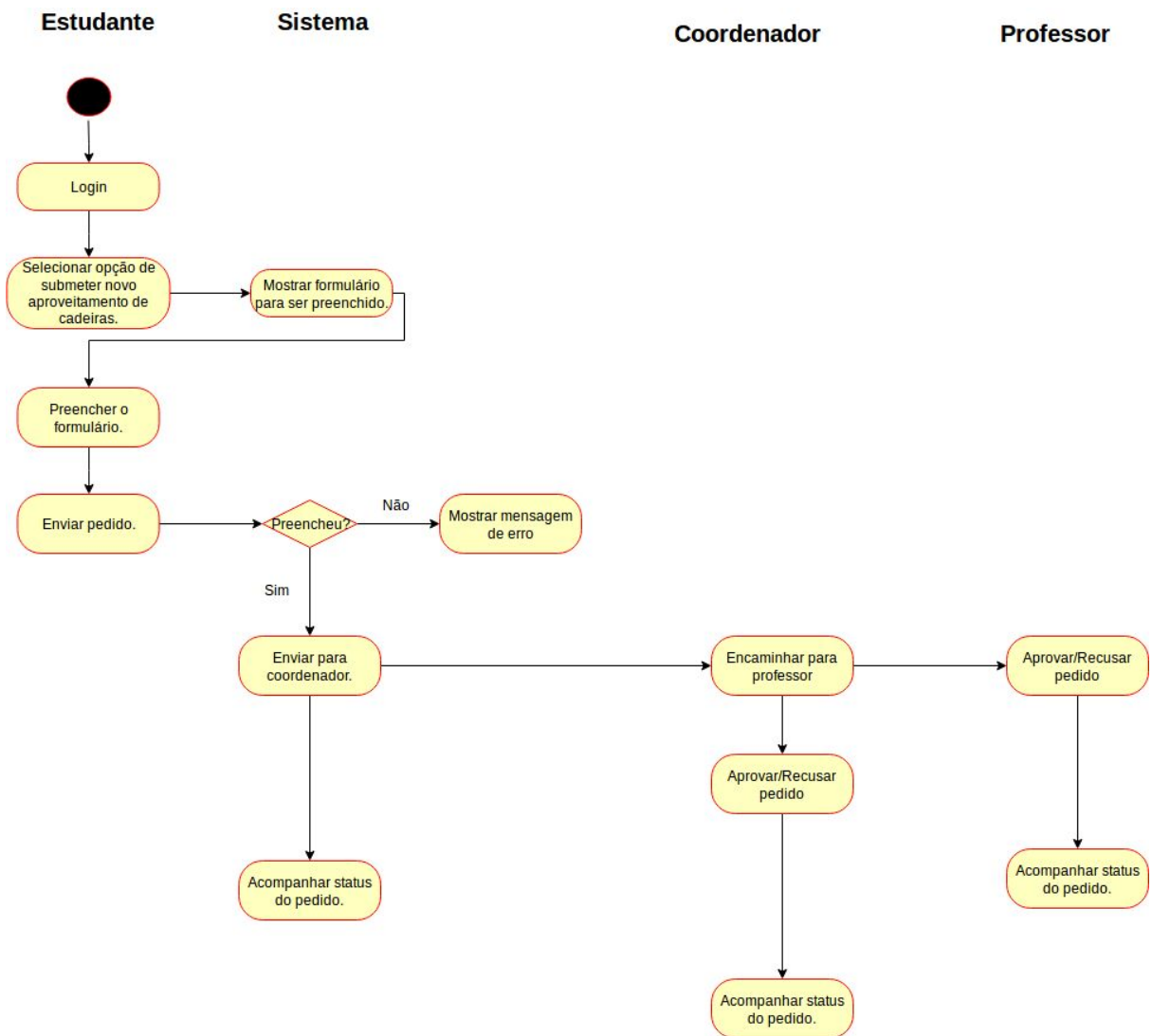


Figura 10 - Diagrama de Atividades de Aproveitamento de Cadeiras

## 2.5 - Diagrama de Sequência

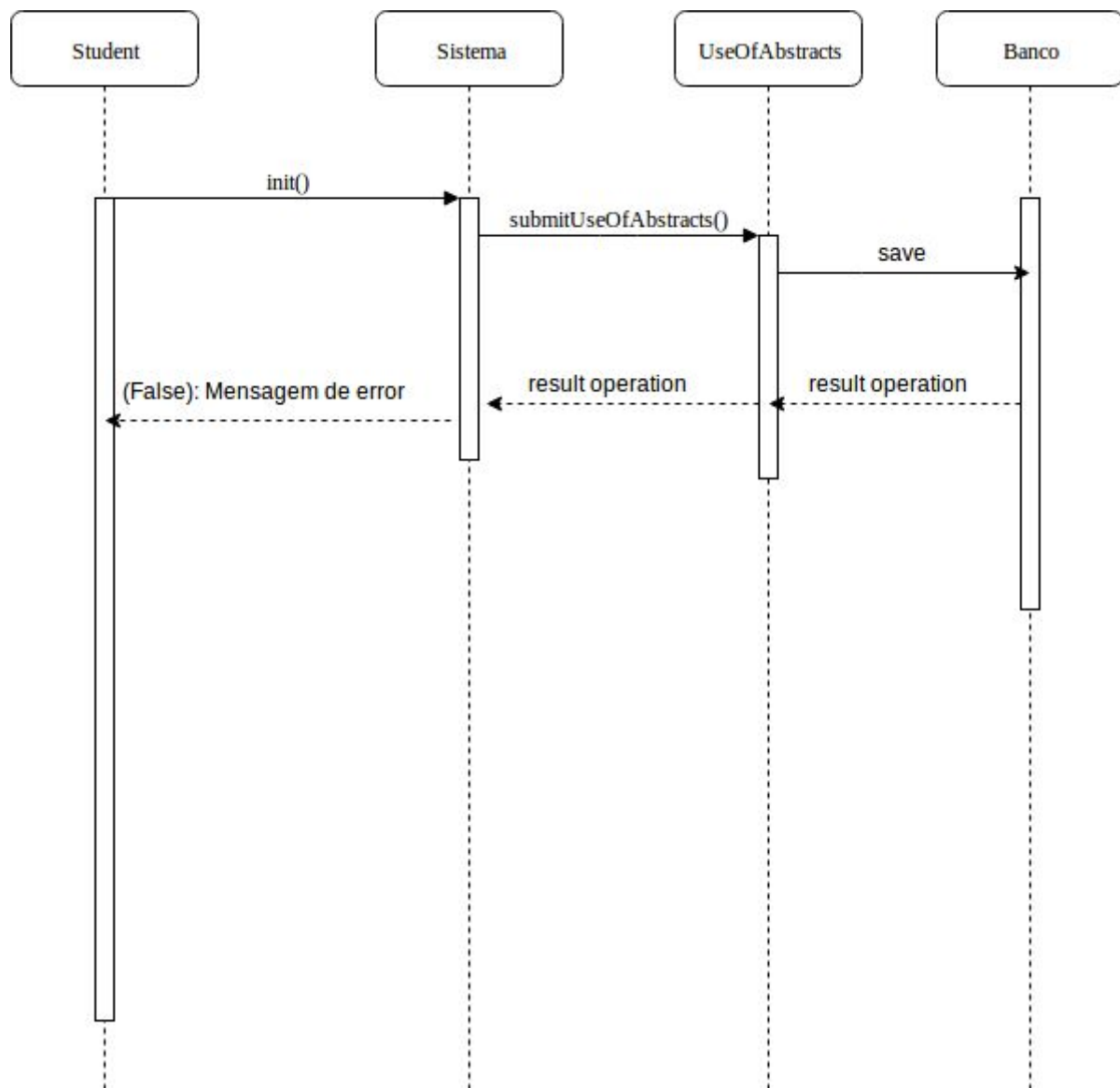


Figura 11 - Diagrama de Sequência de Estudante no módulo de aproveitamento de cadeiras

## References

- Ackerman, A. F., Buchwald, L. S., and Lewski, F. H. (1989). Software inspections: an effective verification process. *IEEE Software*, 6(3):31–36.
- Damian, A., Marques, A., Diniz, S., and Diniz Junqueira Barbosa, S. (2015). Estudo de viabilidade da molveric cards: Uma técnica para a inspeção de diagramas MoLIC.
- DAMIAN, Adriana Lopes. Técnicas para inspeção de diagramas MOLIC. 2016. 174 f. Dissertação (Mestrado em Informática) - Universidade Federal do Amazonas, Manaus, 2016.
- de Paula, M. G., da Silva, B. S., and Barbosa, S. D. J. (2005). Using an interaction model as a resource for communication in design. In *CHI '05 Extended Abstracts on Human Factors in Computing Systems, CHI EA '05*, pages 1713–1716, New York, NY, USA. ACM.
- Fagan, M. E. (1976). Design and code inspections to reduce errors in program development. volume 15, pages 182–211.
- Qazi, A., Shahzadi, S., and Humayun, M. (2016). A comparative study of software inspection

techniques for quality perspective. 10:9–16.