

Shit-Checkin

1. Introdução

Este arquivo apresenta o documento de projeto do sistema de checkin de registro do uso de banheiros da Shit-Checkin. Começando pelo projeto da arquitetura do sistema, passando ao detalhamento dos componentes da arquitetura, até chegar ao projeto detalhado das classes.

Este documento está organizado da seguinte forma: a seção 2 apresenta a plataforma de software a ser utilizada na implementação do sistema; a seção 3 discute aspectos do projeto da arquitetura do sistema; as seções 4 e 5 apresentamos modelos relativos aos subsistemas identificados; finalmente, a seção 6 discute aspectos do projeto.

2. Plataforma de Implementação

O sistema se baseia na seguinte arquitetura para seu funcionamento:

- O gerenciamento de dados é feito através de um banco de dados SQL;
- A interação do usuário será apenas para registrar seu check-in ou ver a lista de check-ins;
- Sistema de cadastro simples;

3. Arquitetura de Software

Poucas são às restrições a usabilidade do sistema, visto a sua simplicidade e aplicabilidade. A descrição dos requisitos propostos são os seguintes:

- **Usabilidade:** Será possível registrar e ler check-ins. Cada objeto deverá possuir nome do usuário, horário de entrada, imagem da entrada e da saída;
- **Desempenho:** É um sistema leve, com banco de dados simples, capaz de realizar a função sem complicações;
- **Disponibilidade:** Será disponibilizado dentro da rede da empresa e em todas as suas filiais;
- **Portabilidade:** Pode ser utilizado em qualquer dispositivo que tenha acesso a um navegador.

O sistema foi feito com um banco de dados MySQL, dentro de um servidor local, utilizando a linguagem de programação JAVA para o desenvolvimento do backend(lado do sistema que trabalha diretamente com os dados), a partir do framework *Spring* e foram utilizados os frameworks *Bootstrap* e *AngularJS* para o desenvolvimento da interface em HTML e CSS. O fluxo do processo está descrito na Figura 1.

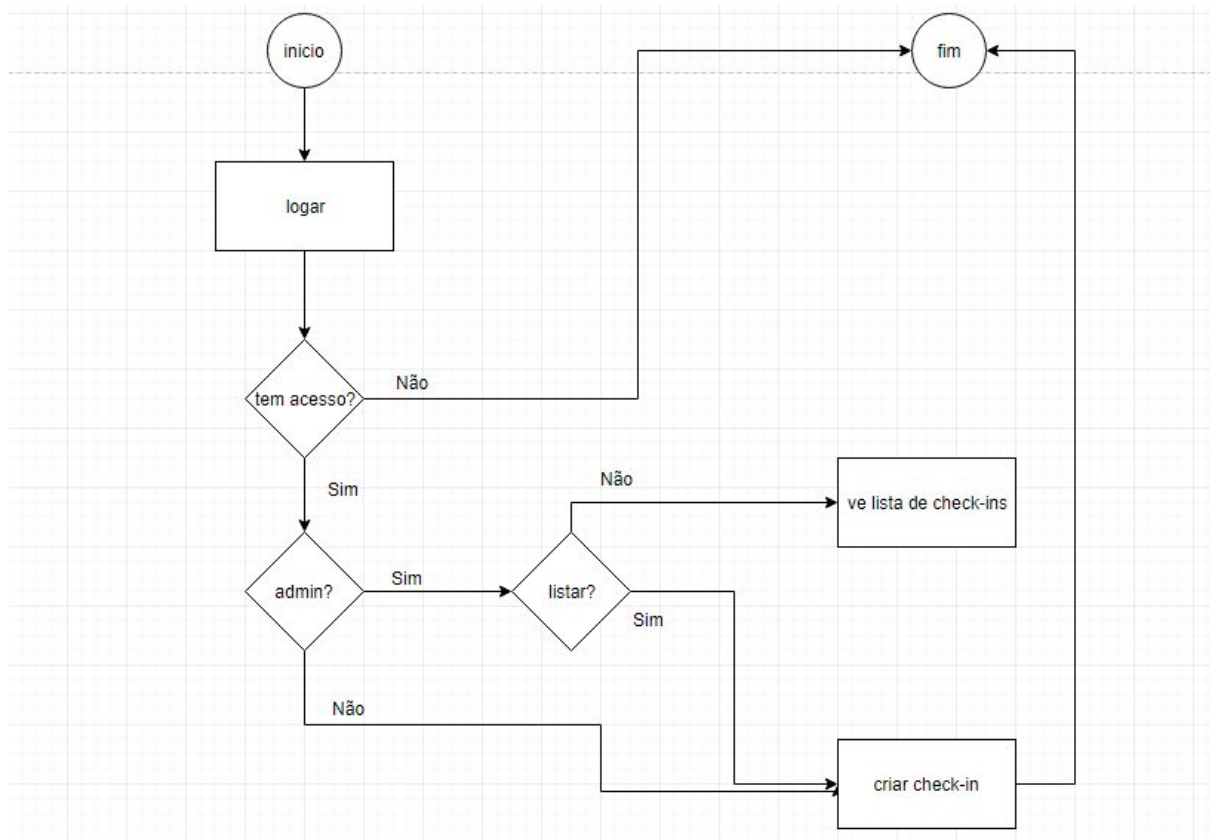


Figura 1 - Diagrama do fluxo do processo.

O sistema conta com as funcionalidades de um CRUD (Create, Read, Update, Delete). As ações de cada ator desse sistema estão descritas no diagrama de casos de uso, conforme mostrado na figura 2.

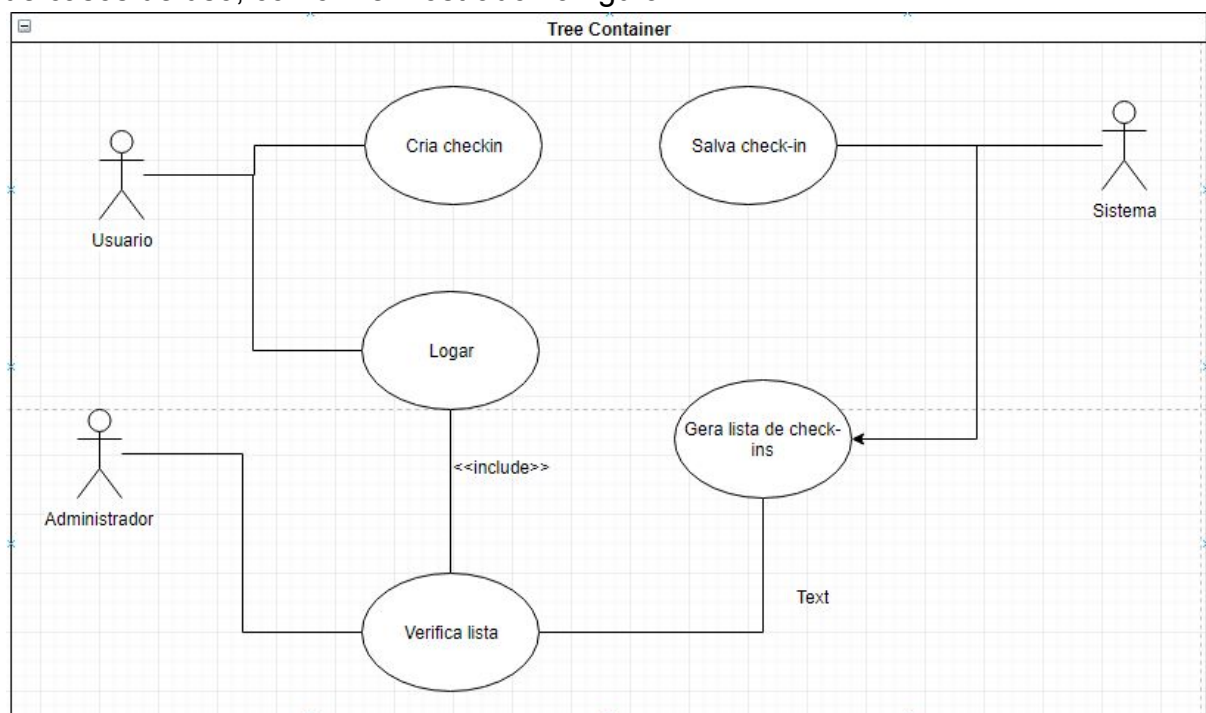


Figura 7 - Diagrama de casos de uso.

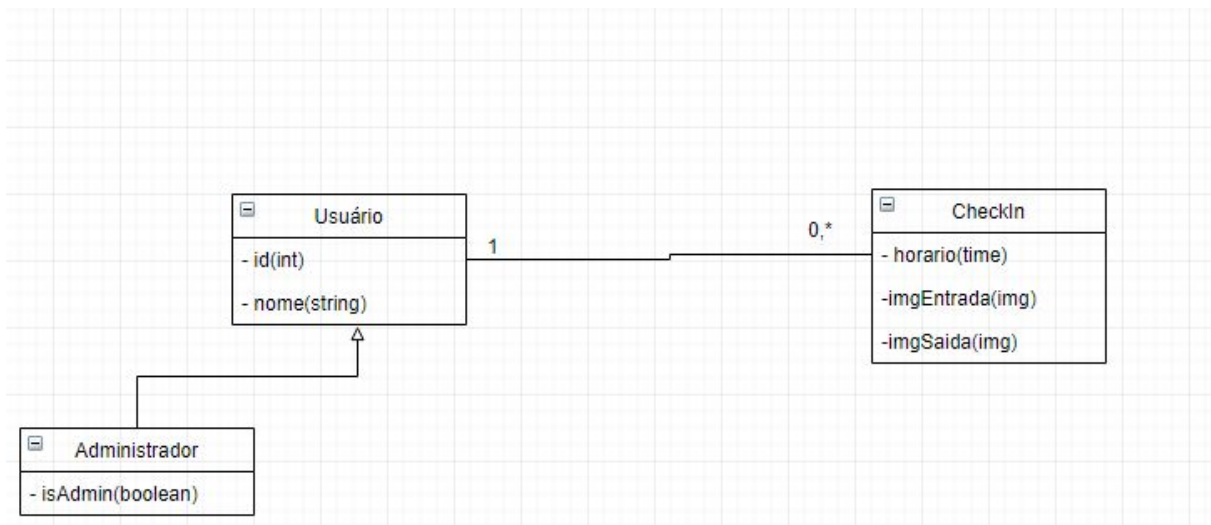


Figura 5 - Diagrama de classe do projeto.