

## Trabalho final

O trabalho deverá ser feito em trio

Fazer a implementação do software para acesso de usuários ao sistema, conforme especificado adiante. O diagrama de casos de uso do sistema é apresentado na Figura 1.

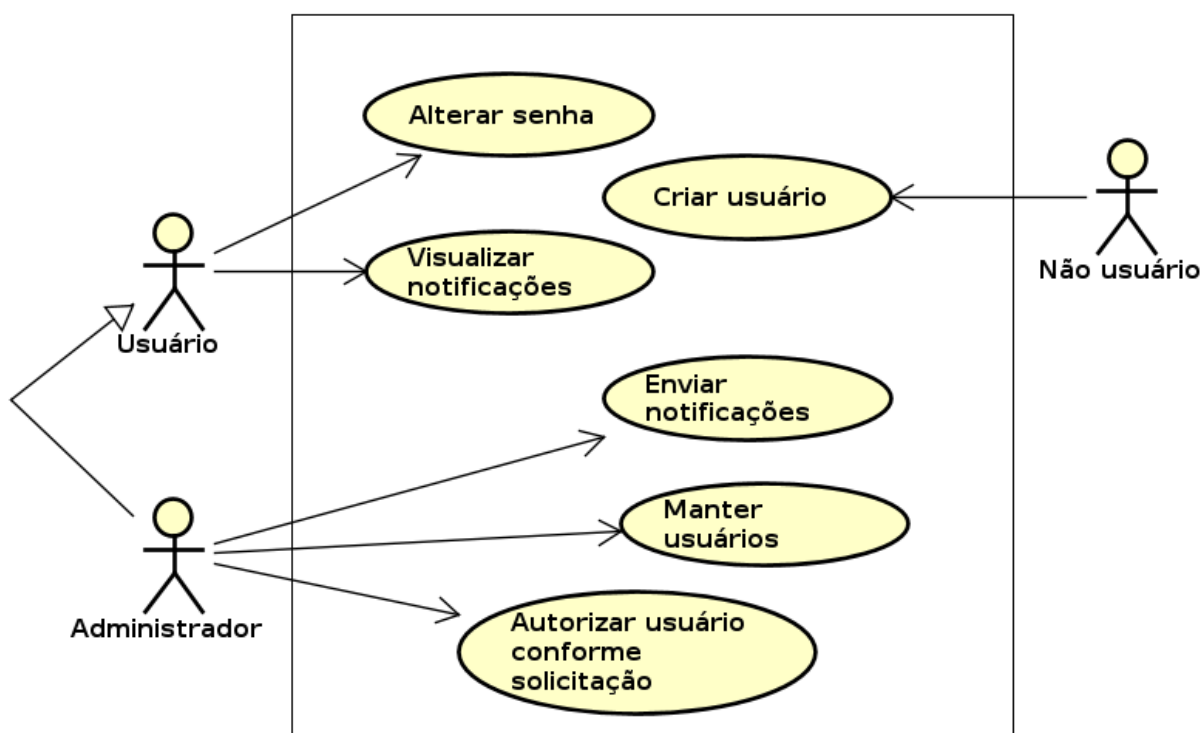


Figura 1 - Diagrama de Casos de Uso do sistema que será desenvolvido

### Descrição do minimundo

Ao fazer acesso o primeiro usuário realizará seu cadastro e torna-se o administrador. Ele poderá cadastrar outros usuários. Outros usuários não cadastrados, aqui chamados “*Não usuários*” podem fazer seu cadastro (Criar usuário), entretanto, precisam aguardar a autorização do administrador para fazer acesso ao sistema.

Todos os usuários devem fazer acesso ao sistema com nome de usuário e senha (autenticação).

Os administradores podem enviar mensagens ou notificações para um, ou mais usuários cadastrados no sistema. Ao acessar o sistema os usuários devem ter um meio de visualizar as notificações que devem ser marcadas como “lida”.

Ao listar os usuários o administrador deve visualizar, além do nome e data de cadastro, o número de notificações enviadas e o número de notificações lidas.

A qualquer momento, desde que autenticados, cada usuário poderá realizar a alteração da sua senha.

O sistema deve realizar o registro em arquivo (log de sistema) para as seguintes operações:

1. Inclusão, alteração ou exclusão de usuários;
2. Envio de notificações;
3. Leitura de notificações;
4. Alteração de senha;
5. Autorização de usuário.

As mensagens gravadas no arquivo de log devem ser desta forma:

Exemplos:

"<OPERACAO>: <<NOME>>, (<<DATA>>, <<HORA>>, e <<USUARIO>>";

Caso as operações falhem: Ocorreu a falha <<MENSAGEM DA FALHA>> ao realizar a "<OPERACAO> do contato <<NOME>>, (<<DATA>>, <<HORA>>, e <<USUARIO>>."

Substituir as variáveis entre pelos respectivos dados.

<<NOME>> = nome do contato sob o qual a operação está sendo realizada

<<DATA>> = data atual do sistema operacional, no formato DD/MM/AAAA, onde D é dia, M – mês, e A – ano.

<<HORA>> = hora atual do sistema operacional, no formato HH:MM:SS, onde HH é a hora, MM – minuto e S segundos.

<<USUARIO>> = usuário autenticado no sistema

<<MENSAGEM>> = mensagem de falha no sistema

<<OPERAÇÃO>> = Operações listadas acima

Deve ser disponibilizada uma tela de "Configuração" que permita escolher dentre os formatos de arquivos de log, o que será usado na aplicação. O usuário poderá mudar o tipo de log em tempo de execução. A escolha feita pelo usuário deverá ser persistida.

Exibir no rodapé do sistema:

- Usuário logado
- Tipo {Usuário, Administrador}
- Botão - quanto existirem notificações, cujo texto deve ser o total de notificações. Ao clicar deve ser aberta uma janela onde o usuário visualiza e acessa o que for indicado nas notificações

## Requisitos não funcionais

### RNF01 – Casos de uso do tipo CRUD com State

Os casos de uso do tipo CRUD devem seguir o que foi ministrado e debatido em aula, utilizando o padrão de navegação entre janelas disponíveis em <https://whimsical.com/crud-manter-HcNY9D491ajiNZ7j8EA7Y9>

### RNF02 – Persistência de dados

Utilize como persistência qualquer banco de dados, relacional ou não, que não exija instalações ou configurações extras no ambiente de implementação. Prefira utilizar o SQLite. Não utilizar o Docker para manter o banco de dados.

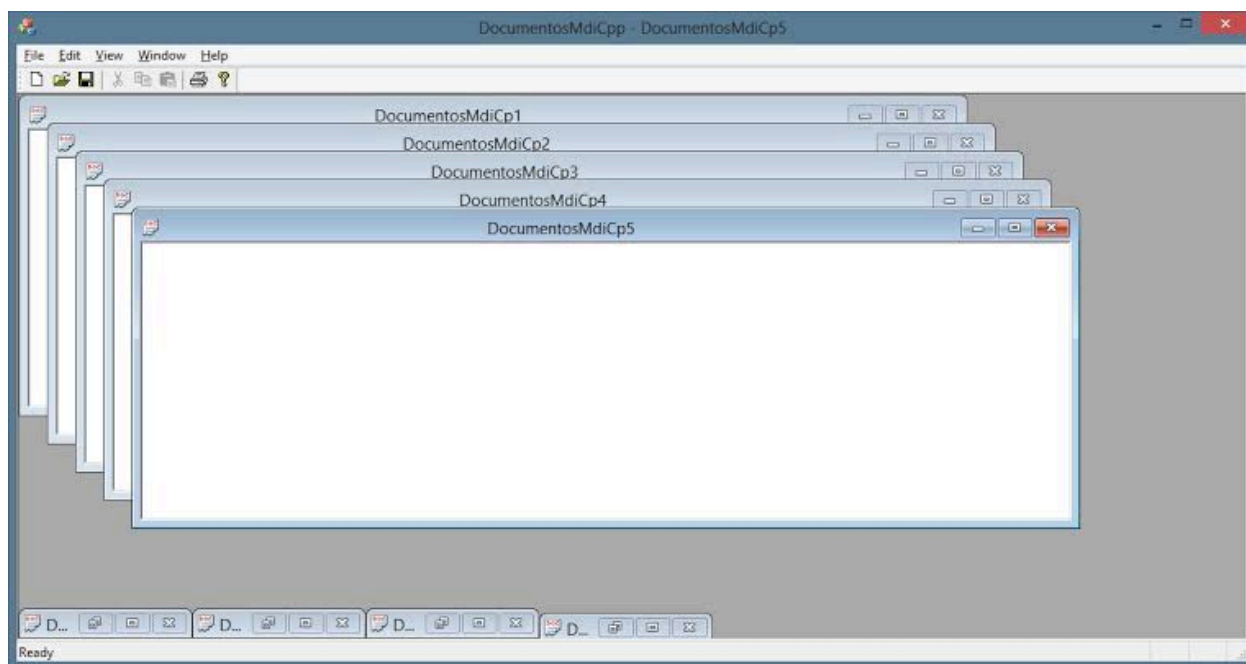
### RNF03 – Padrão de projeto DAO

Utilize o padrão de projeto DAO, sabendo que não é permitido utilizar Framework de persistência como Hibernate ou similares.

## RNF04 – Interface gráfica

O sistema deve ser desenvolvido utilizando o kit de componentes gráficos do Java Swing.

O sistema deve permitir que usuários naveguem entre telas que estejam abertas para realizar as funções desejadas. Ou seja, utilizar o MDI (Multiple Document Interface<sup>1</sup>). Um exemplo de sistema com o padrão MDI é apresentado na figura abaixo. A barra de botões é opcional.



## RNF05 – Utilizar o padrão de projeto Command

Ao utilizar o State, utilizar o padrão Command integrado a ele.

## RNF 06 – Padrão de projeto Observer

Implemente o padrão Observer de modo que caso qualquer usuário altere algum dado em uma janela aberta, as demais janelas abertas referentes àquele dado sejam atualizadas.

## RNF07 – Formatos de arquivos de Log

O sistema deve permitir os seguintes formatos para arquivos de log: CSV (com separação de campos usando ponto e vírgula “;”), JSON.

<sup>1</sup> [https://pt.wikipedia.org/wiki/Interface\\_de\\_documentos\\_m%C3%BAtiplos](https://pt.wikipedia.org/wiki/Interface_de_documentos_m%C3%BAtiplos)

### **RNF08 – Padrão de Projeto Adapter para Log**

O módulo de log deve adotar o padrão de projeto Adapter. Além disso, deve ser um projeto Maven independente, utilizado pelo sistema proposto nesta especificação.

### **RNF09 – Criação de senhas**

Ao criar senhas do sistema utilizar adaptar o Validador de senhas, disponível em <https://github.com/claytonfraga/validadorsenha>, que deve ser incluído como JAR<sup>2</sup> no projeto Maven, utilizando o JitPack (<https://jitpack.io/>)

### **Demais requisitos que fazem parte da avaliação:**

1. A dupla deve criar um documento no Google Docs que descreva de forma explícita quais requisitos foram implementados. Serão considerados apenas os requisitos que foram implementados e totalmente atendidos.
2. Os softwares com erros de código que impeçam a compilação não serão corrigidos e receberão nota zero.
3. O software que não executar não será corrigido e receberá nota zero
4. A aplicação dos conceitos e técnicas estudados na disciplina;
5. Utilização dos padrões de projeto adequados e a criatividade e aplicabilidade desses padrões;
6. O sistema deve ser desenvolvido utilizando linguagem Java 17;
7. O projeto deve ser criado utilizando o Maven;
8. O padrão MVP (Passive View) foi debatido em aulas e correções realizadas durante o semestre;
9. O tratamento de exceções deve ser feito em todo o projeto;
10. A entrega do projeto deverá ser feita pelo Gitlab ou Github, não usar como backup com um arquivo compactado, utilizar como repositório para controle de versões e trabalho

---

<sup>2</sup> <https://www.reviversoft.com/pt/file-extensions/jar>



em equipe. A entrega deverá ser feita já com o comando clone: `git clone <link do repositório>`.