

IMUNIZA FÁCIL

SISTEMA DE GERENCIAMENTO DA FILA DE VACINAÇÃO

Jonas Neres Da Silva

Luan Costa Santos

Lucas Da Silva Guiron

Lucas Marchesoni Silvério Da Silva

Vinicius Takedi Souza Brunelli

Thiago Da Silva Barbosa

SÃO PAULO

2021

Jonas Neres Da Silva

Luan Costa Santos

Lucas Da Silva Guiron

Lucas Marchesoni Silvério Da Silva

Vinicius Takedi Souza Brunelli

Thiago Da Silva Barbosa

IMUNIZA FÁCIL

SISTEMA DE GERENCIAMENTO DA FILA DE VACINAÇÃO

Trabalho composição de nota, apresentado ao curso de Ciências da Computação da Universidade São Judas como complemento das "UCs" Modelagem de Software e Programação de Soluções Computacionais.

SÃO PAULO

2021

Resumo

Vivemos hoje em um mundo globalizado e cercado de tecnologias, que nos auxiliam a fazer tudo a todo momento, e devido ao surgimento da pandemia causada pelo Coronavírus SARS-CoV 2, causador da doença Covid-19 e como a vacinação é o melhor e mais eficaz método de prevenção contra a Covid-19.

E observando que no Brasil não existe um sistema centralizado para gerenciar a fila de vacinação. Foi identificado a necessidade de um sistema computacional capaz de cadastrar e gerenciar os pacientes e a ordem de vacinação deles.

Nesse contexto, o presente projeto propõe o “IMUNIZA FÁCIL - SISTEMA GERENCIADOR DA FILA DE VACINAÇÃO”, que irá gerenciar e centralizar os dados da fila de vacinação em um banco de dados informatizado, permitindo consultas e registros de pacientes e da ordem que serão vacinados de acordo com as regras divulgadas pelo Ministério da Saúde.

Abstract

We live today in a globalized world surrounded by technologies, which help us to do everything at all times, and due to the emergence of the pandemic caused by the Coronavirus SARS-CoV2, which causes Covid-19 disease and how vaccination is the best and most effective prevention method against Covid-19.

And noting that in Brazil there is no centralized system to manage the vaccination queue. The need for a computer system capable of registering and managing patients and their vaccination order was identified.

In this context, the present project proposes the “IMUNIZA FÀCIL - VACCINATION QUEUE MANAGEMENT SYSTEM”, which will manage and centralize the vaccination queue data in a computerized database, allowing consultations and records of patients and the order to be vaccinated according to the rules released by the Ministry of Health.

Sumário

Sumário

1. Propósito do Documento de Requisitos
2. [Requisitos](#_gjdgxs)

2.1 [Funcionais](#_3znysh7)

2.2 [Não Funcionais](#_3rdcrjn)

1. [Diagramas](#_3dy6vkm)

3[.1 Diagrama de Caso de Uso](#_1t3h5sf)

3.2 Diagrama de Classes

3.3 Diagrama de atividade

3.4 Diagrama de Componentes

3.5 Diagrama de estado

3.6 Diagrama de implantação

3.7 Diagrama de Sequência

1. Modelo Entidade e relacionamento
2. Diagrama Entidade e relacionamento
3. Análise do banco de dados
4. Análise orientada a objeto

7.1 Pacientes

7.2 Funcionário

1. Prototipação

# Propósito do documento

Este documento possui as especificações de requisitos para o software desktop com interface gráfica e banco de dados, que tem como objetivo facilitar às pessoas a se imunizarem contra o vírus SARS-CoV2/Corona-Vírus/Covid . Diante desta triste realidade que é a pandemia, o desafio será unir a disponibilidade de vagas nas filas de vacinas, de forma rápida, simples e de fácil gestão.

# 

# 2. Requisitos

## 2.1 Funcionais

|  |  |  |
| --- | --- | --- |
| ID | Descrição | Plataforma |
| RF001 | O sistema deverá ter o usuário administrador cadastrado previamente, esse usuário deverá colocar suas credenciais para o sistema autenticar antes de realizar o login. Diferenciando a entrada como administrador e como atendentes, aparecem as opções referentes a cada cargo, já que os administradores poderão visualizar tudo, e os atendentes as opções específicas. | Desktop |
| RF002 | O sistema vai ter um menu de opções onde o funcionário administrador do sistema onde ele pode selecionar a opção para cadastrar administradores e atendentes. | Desktop |
| RF003 | O administrador também poderá selecionar uma opção para cadastrar pessoas a serem vacinadas onde já virão características definidas no sistema para ele apenas preencher os dados, essas características são nome, idade, profissão relacionada a área da saúde, endereço e data de vacinação. | Desktop |
| RF004 | Deverá ter uma opção para o administrador selecionar para remover administrador e atendentes. | Desktop |
| RF005 | Uma opção para o sistema gerar um relatório com um período selecionado pelo usuário, o relatório vai trazer a média de vacinação diária para as faixas etárias que estarão disponíveis. | Desktop |
| RF006 | Os atendentes devem ter acesso a opção de visualizar a fila de vacinação seguindo uma ordem de prioridades. | Desktop |
| RF007 | Ele deve ter uma opção para confirmar a vacina, visualizando quem é o próximo da fila e aplicando a vacina, após isso ele deve marcar o checkbox que aquela pessoa foi vacinada, o sistema já aparecerá automaticamente a data de vacina após isso | Desktop |

## 

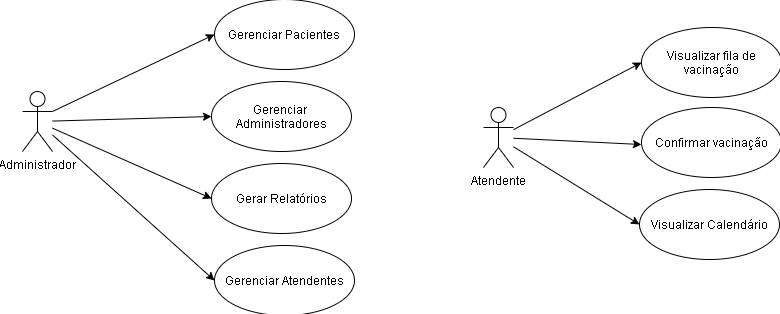
## 2.2 Não Funcionais

|  |  |  |
| --- | --- | --- |
| ID | Descrição | Categoria |
| RNF001 | Somente usuários cadastrados terão acesso ao sistema | Segurança |
| RNF002 | O Usuário Administrador terá acesso total a todas as funcionalidades. | Segurança |
| RNF003 | Os Usuários Pacientes só terão acesso ao seu próprio cadastro e aos dados de sua vacinação. | Segurança |
| RNF004 | Os Usuários agentes de saúde terão acesso em modo consulta, aos cadastros, à relação de unidades sanitárias, ao calendário de vacina e a lista de vacinados. | Segurança |
| RNF005 | Os Usuários gerentes terão os mesmos acessos dos usuários agentes, porém com permissão de criação e edição de cadastro e alterar a relação de unidades sanitárias. | Segurança |
| RNF006 | O tempo de retorno das consultas (isto é, o intervalo de tempo entre qualquer consulta e seu resultado) não pode ser maior do que 5 segundos. | Performance |
| RNF007 | O sistema deve possuir navegação simplificada, de modo a tornar o sistema produtivo e fácil de usar. | Usabilidade |
| RNF008 | O sistema deve possuir botões de ação com imagens relacionadas à funcionalidade dele. | Usabilidade |
| RNF009 | O sistema deve possuir telas de confirmação para ações importantes. | Usabilidade |
| RNF010 | O sistema será dividido em módulos, de modo que as atualizações serão feitas mais rapidamente e sem a necessidade de longos períodos de atualização onde o sistema ficaria impossibilitado de operar. | Manutenibilidade |
| RNF011 | O sistema será implementado na Linguagem de Programação JAVA, sendo assim portável nas plataformas mais utilizadas (Windows, Linux e MacOS). | Processo |
| RNF012 | O sistema possuirá um manual de uso, a fim de auxiliar os diferentes tipos de usuário. Ele explicará detalhadamente como proceder na realização das funções requisitadas para a aplicação. | Documentação |

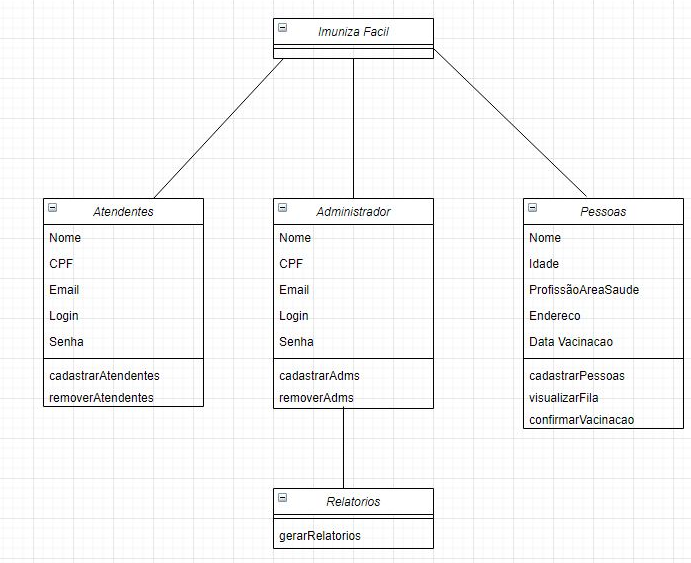
# 3. Diagramas

obs: Caso algum diagrama fique de qualidade ruim segue o link para o repositório no github que contém todas as imagens separadas por diagramas: <https://github.com/viniciustakedi/USTJ-CC-A3/tree/main/Diagramas>

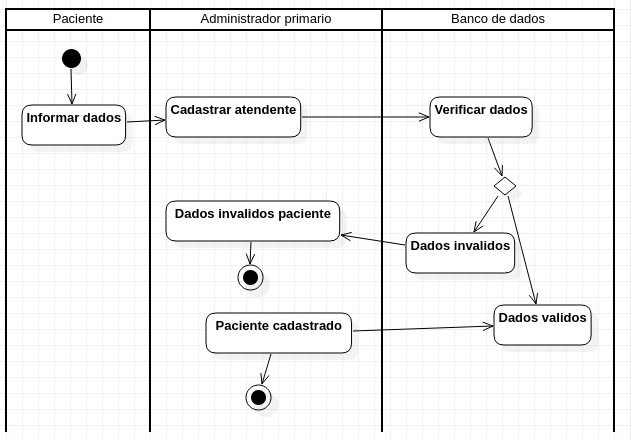
## 3.1 Diagrama de Caso de Uso

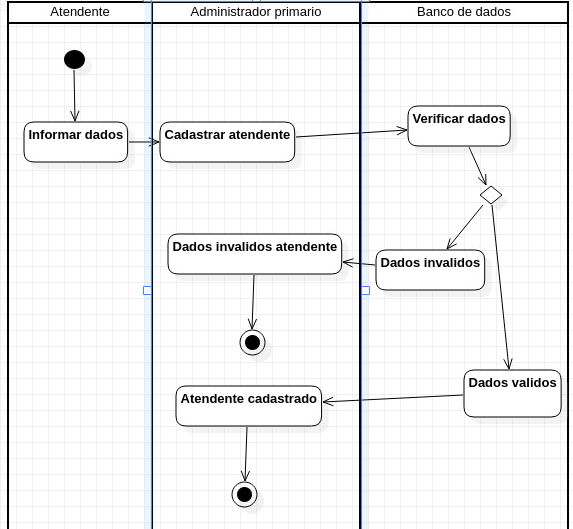


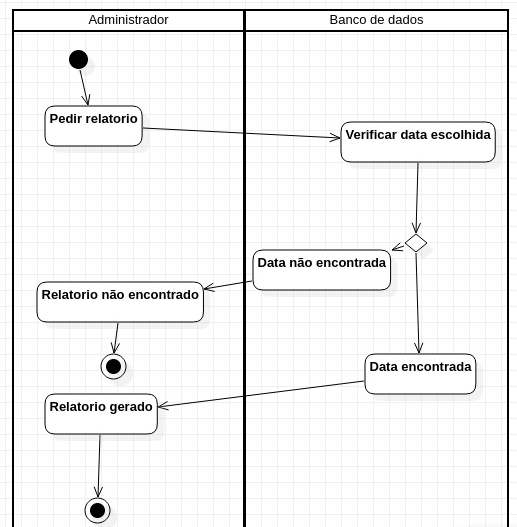
## 3.2 Diagrama de Classes



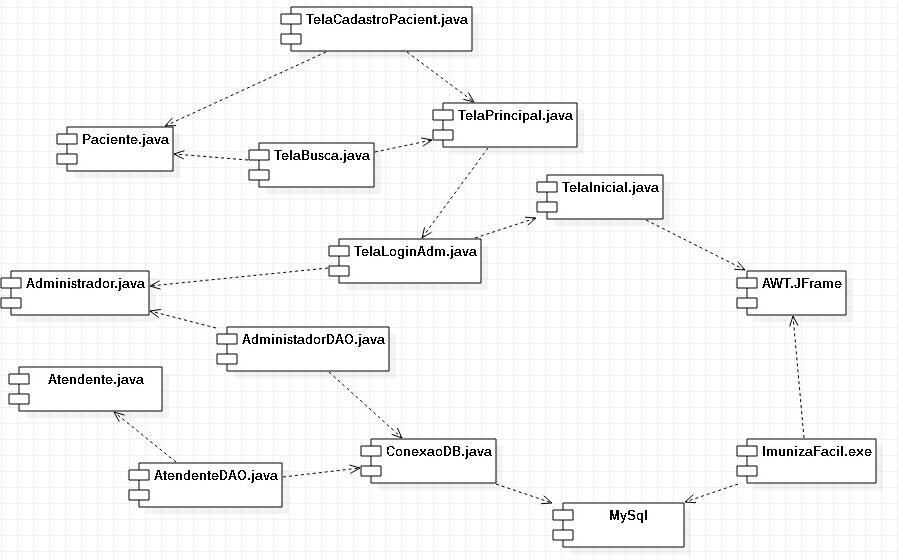
## 3.3 Diagramas de Atividades



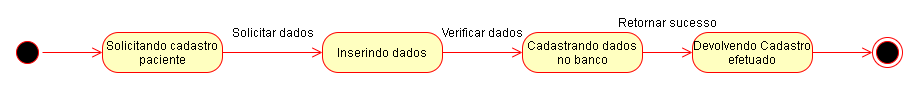


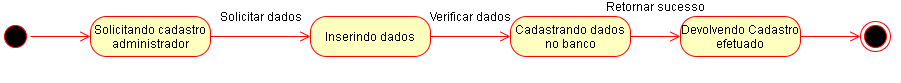


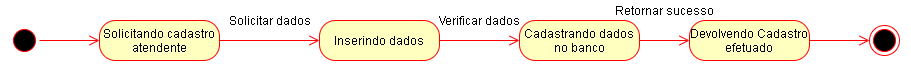
## 3.4 Diagrama de Componentes

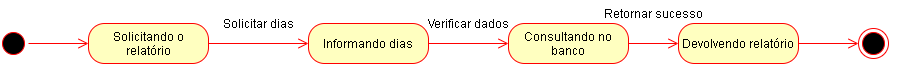


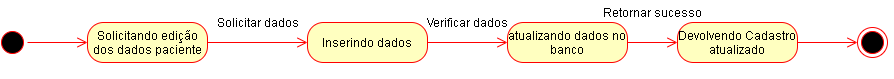
3.5 Diagrama de estado

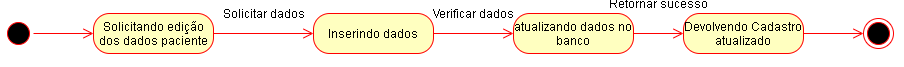


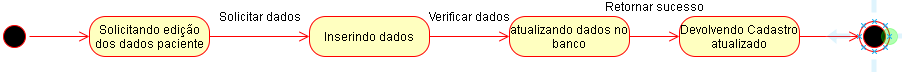




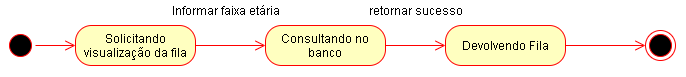


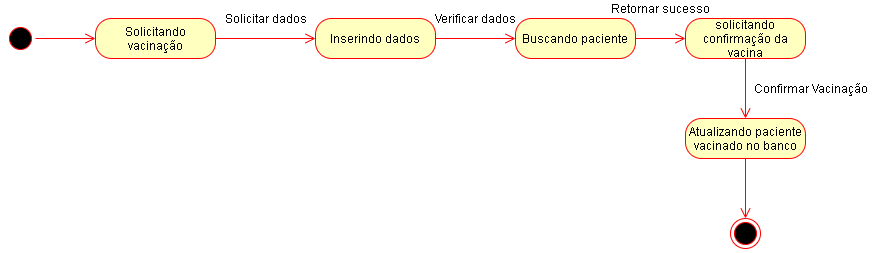




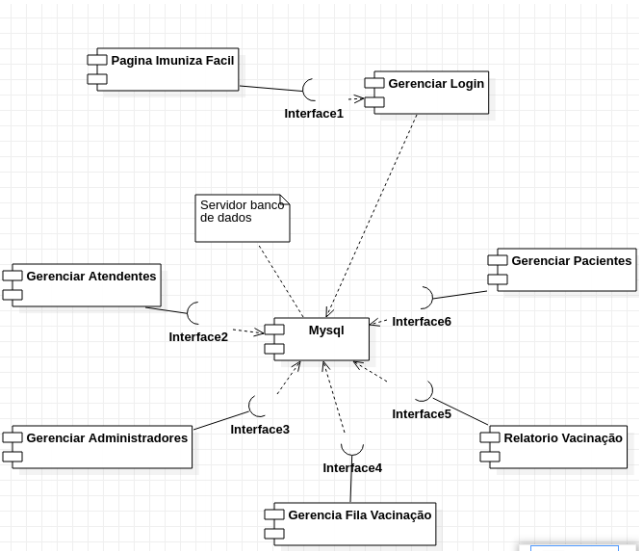




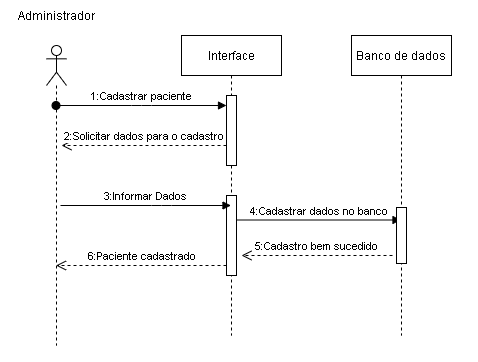


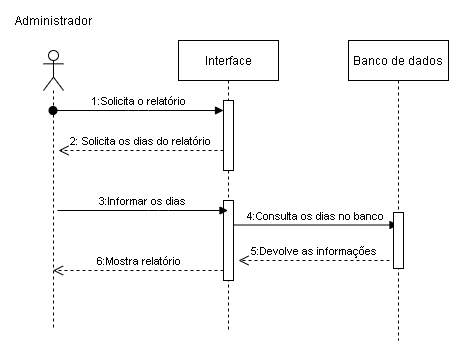


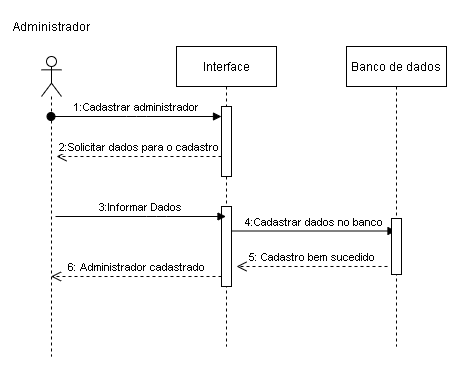
3.6 Diagrama de implantação

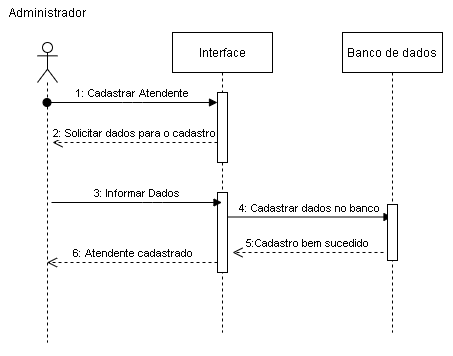


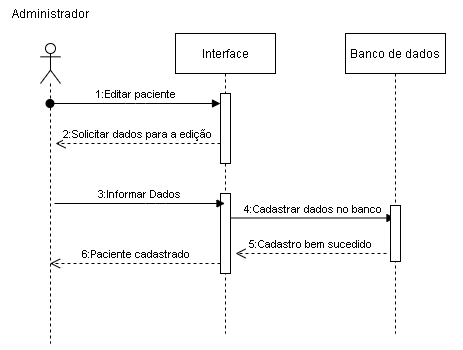
3.7 Diagrama de sequência

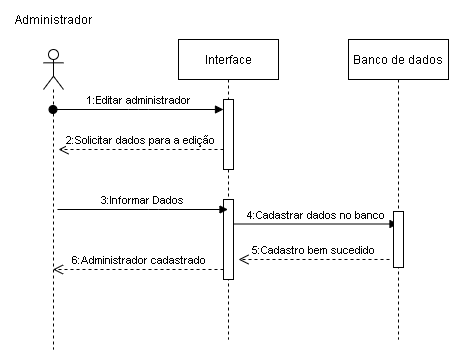


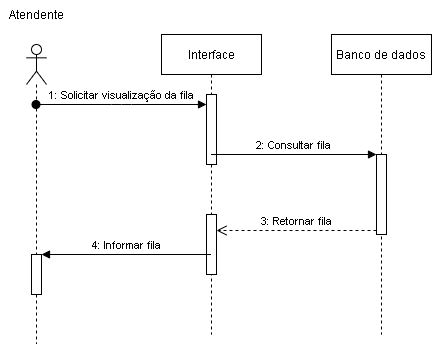


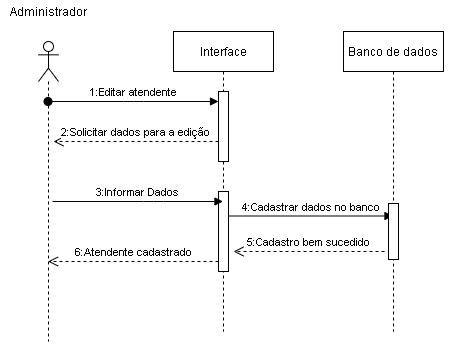


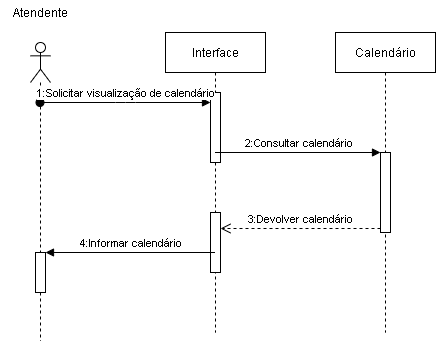


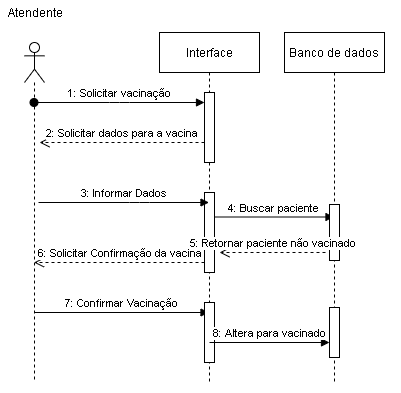




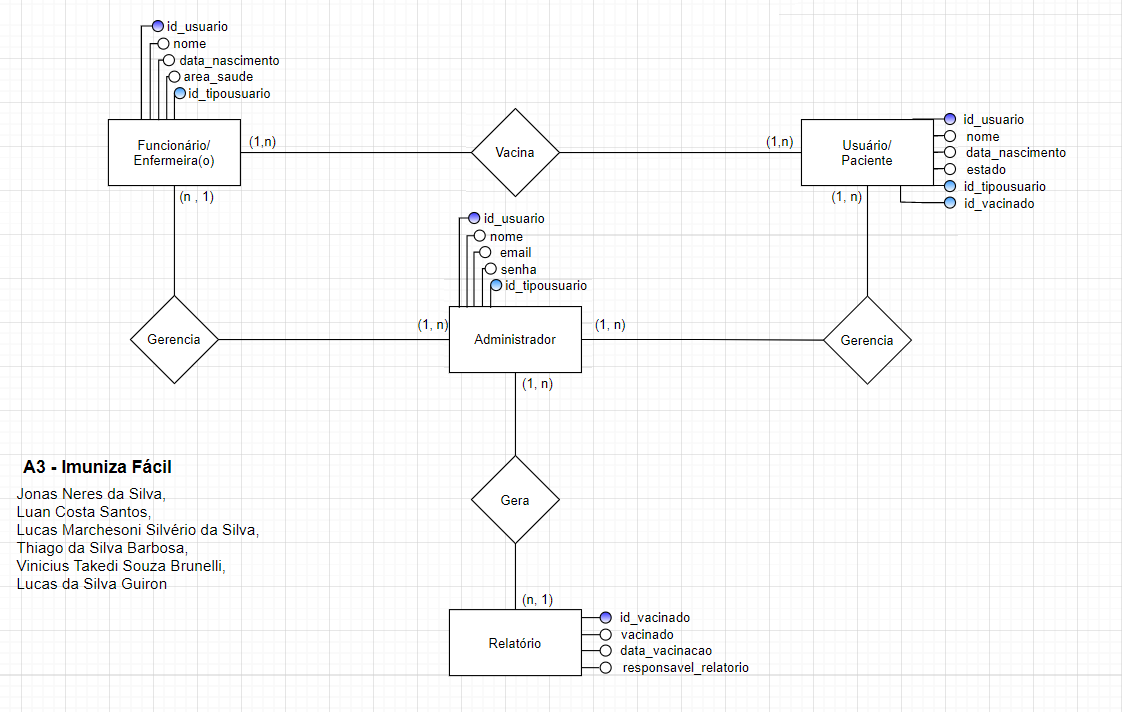




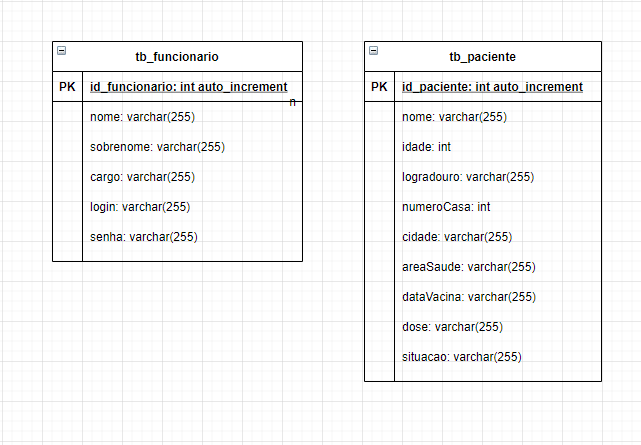




4. Modelo Entidade e relacionamento



5. Diagrama Entidade e relacionamento



6. Análise do banco de dados

O banco de dados do sistema é bem simples, possui apenas duas tabelas. A primeira é a tb\_paciente que vai receber os dados de cadastro dos pacientes a serem vacinados, tendo como primary key e **auto increment** o campo id, e recebendo os dados nome, sobrenome, logradouro, cidade e área da saúde como varchar e como int os campos idade e número da casa. A outra tabela é a tb\_funcionario que vai receber os dados de cadastro dos funcionários e adms, sendo estes diferenciados no campo cargo, existe a primary key e auto incremente que é o campo id, e recebe dados como nome, sobrenome, cargo, login e senha, todos varchar.

7. Análise orientada a objeto

7.1 Pacientes

É composto pelos atributos nome, sobrenome, idade, logradouro, número da casa, cidade, área da saúde, data vacina, dose e status. Tendo seus métodos como salvarPaciente(), editarPaciente(), apagarPaciente() e visualizarPaciente(). Sua função no sistema é receber os dados e salvar no sistema, para saber qual sua posição na fila, quando será vacinado, quantas doses recebeu e etc.

7.2 Funcionário

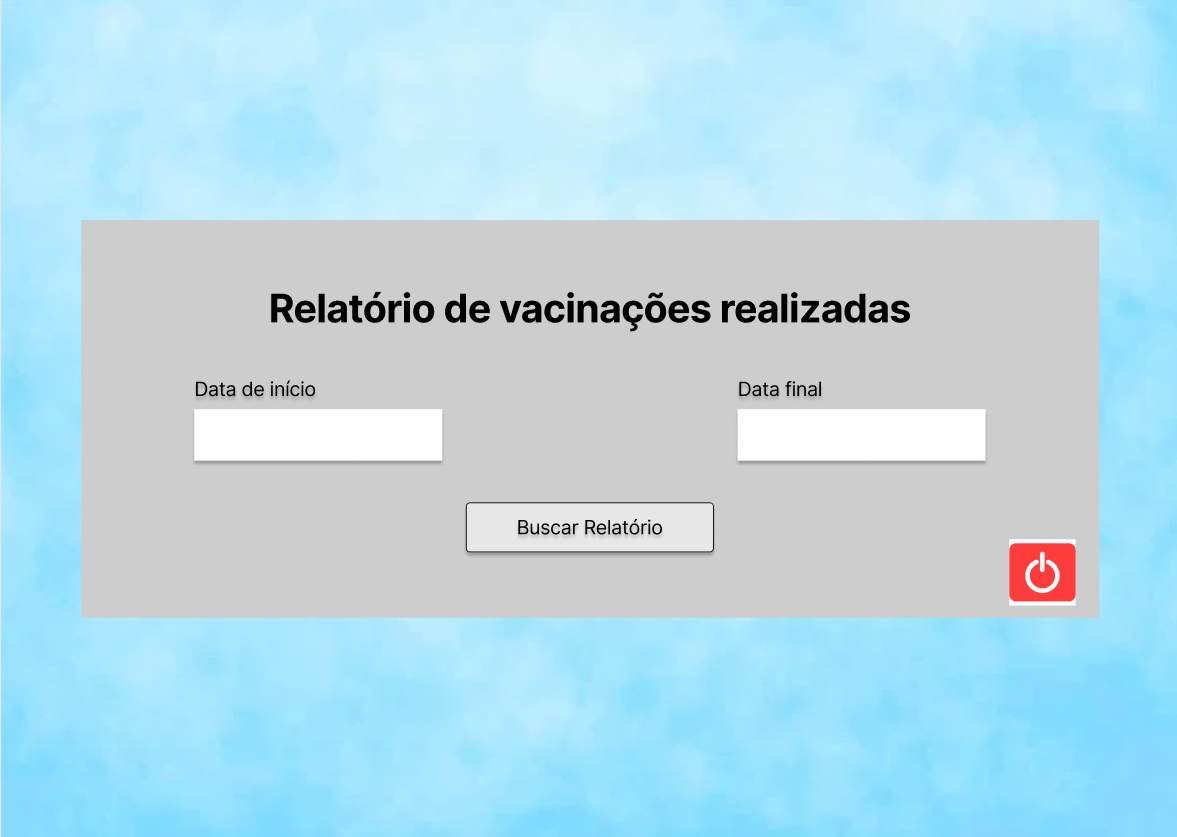
É composto pelos atributos nome, sobrenome, cargo, login e senha. Tendo seus métodos como salvarFuncionario, editarFuncionario(), apagarFuncionario(), visualizarFuncionario(). Sua função no sistema vai variar de acordo com o atributo cargo, caso ele seja definido como administrador, certas funções irão aparecer como cadastrar funcionários, cadastrar pacientes, remover funcionários e gerar relatório. Já se for definido como atendente ele deverá visualizar as funções visualização da fila de vacinação e confirmação de vacinação, ou seja, o objeto funcionário terá sua funcionalidade diferenciada diretamente pelo atributo cargo.

8. Prototipação

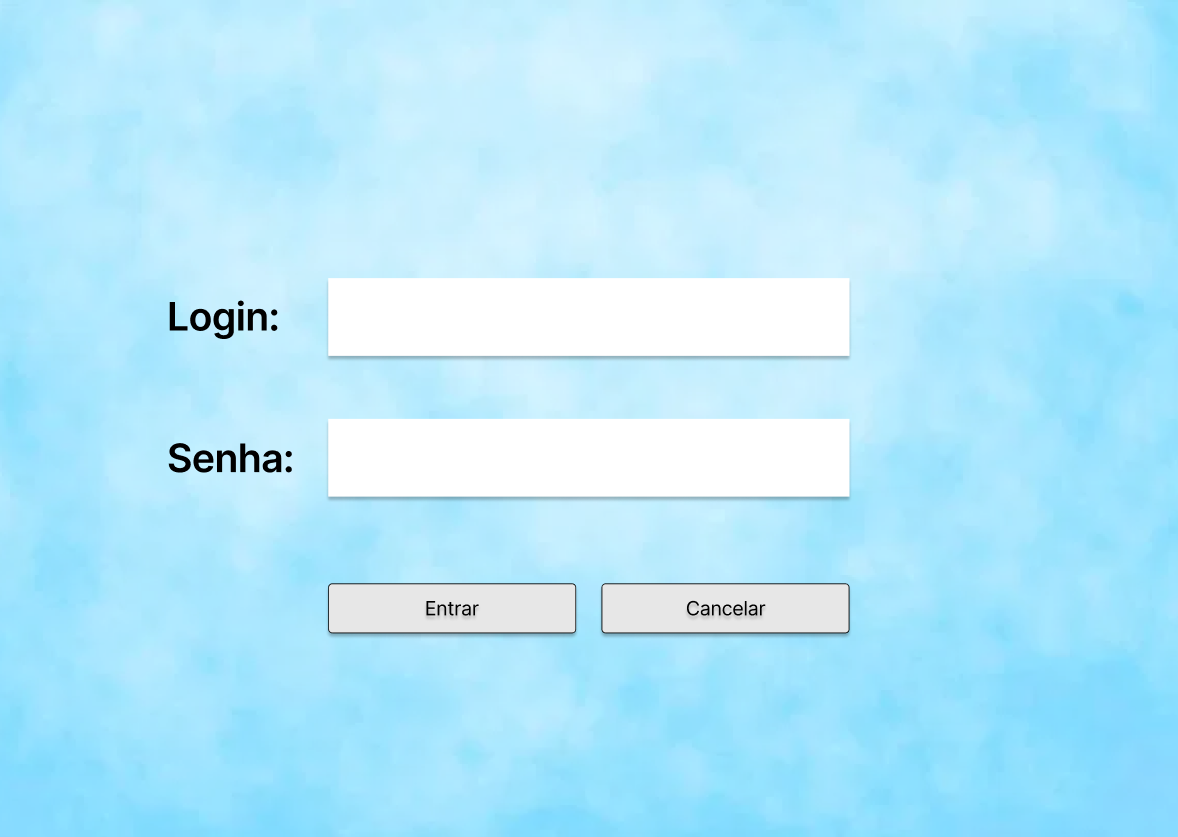
Tela inicial:



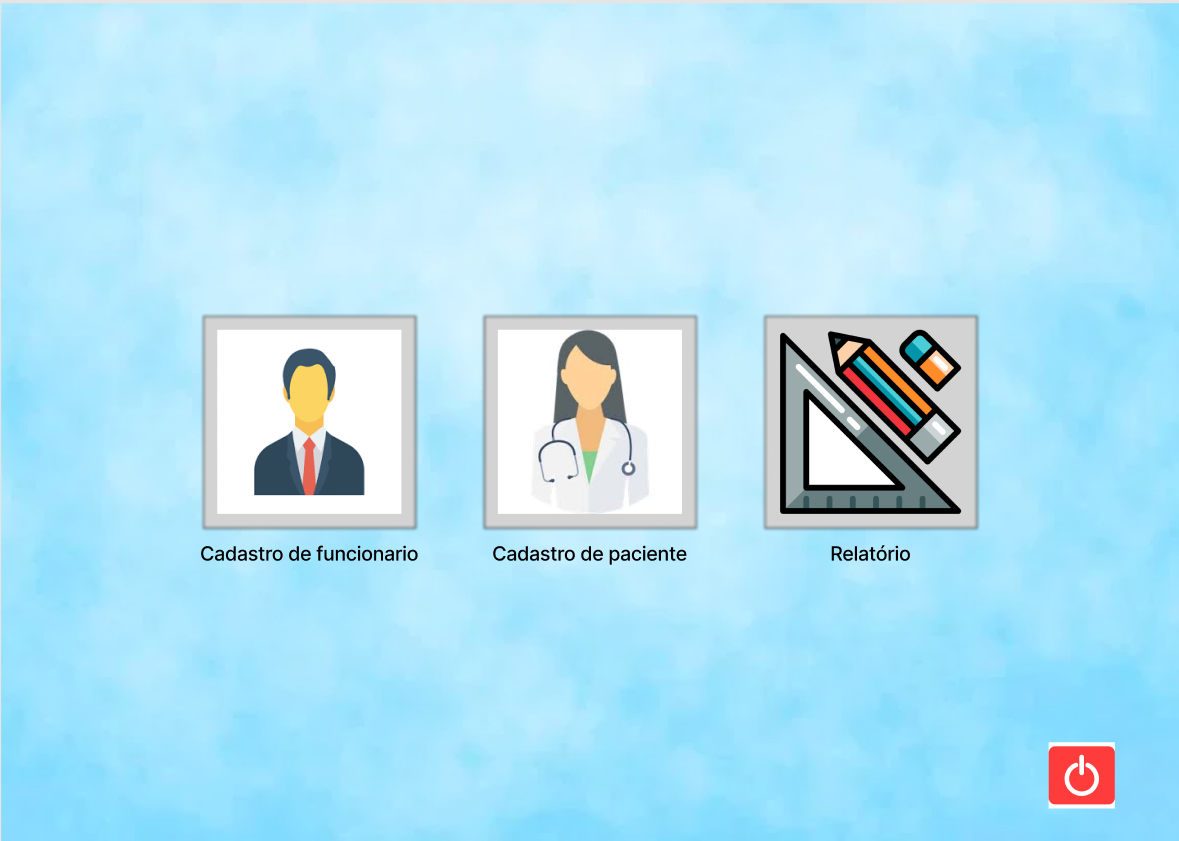
Tela para informar data para vacinação:



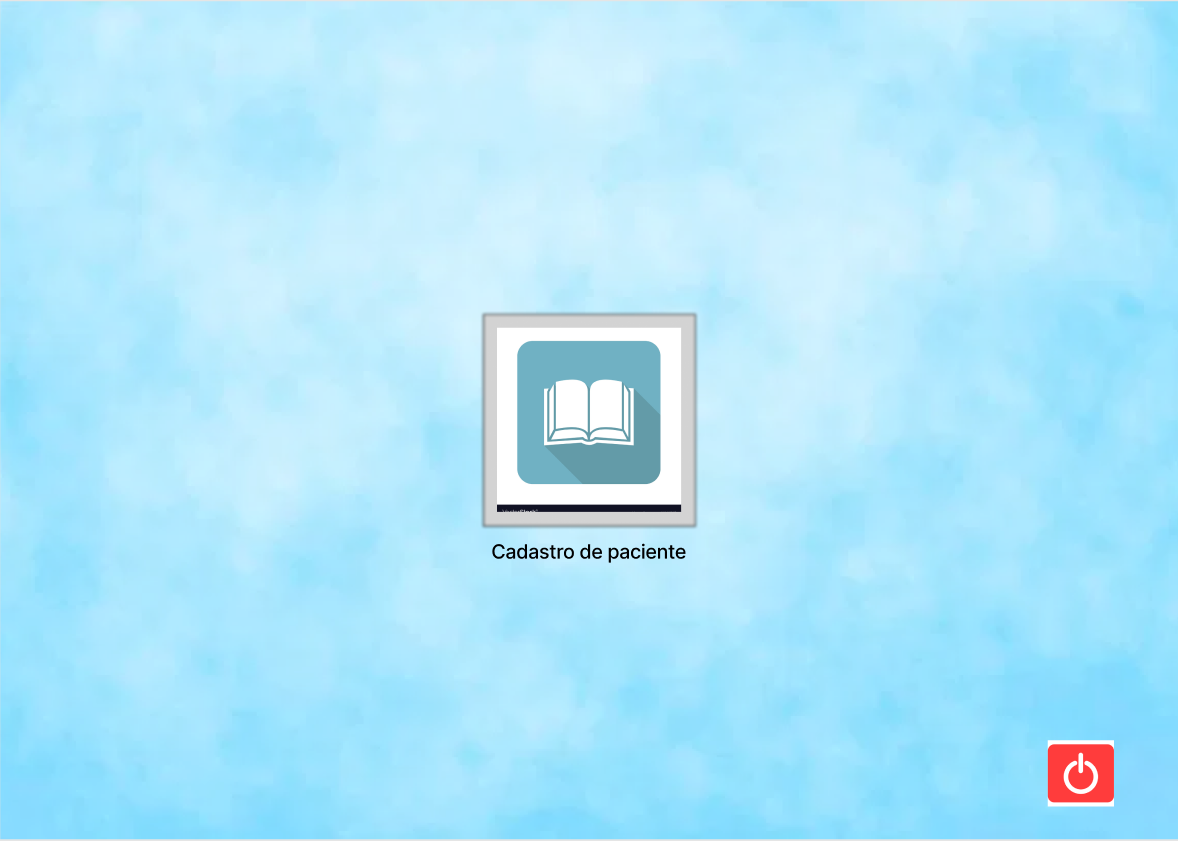
Tela para realizar o login, administrador e funcionário:



Tela administrador para realizar as operações mostradas abaixo:



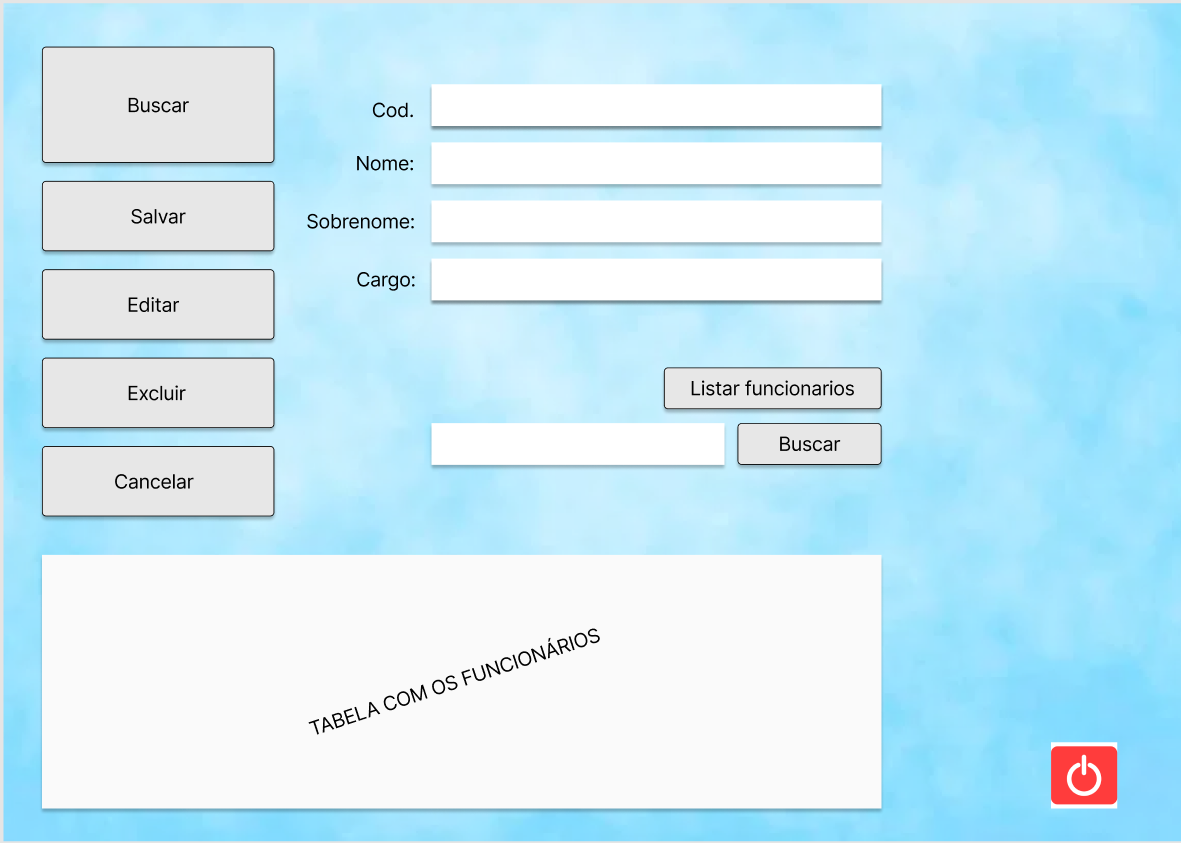
Tela das funções que o funcionário pode realizar:



Tela para verificar a fila de vacinação:



Tela para cadastro de funcionarios, e listagem:



Tela de cadastro de pacientes a serem vacinados,e listagem:

