

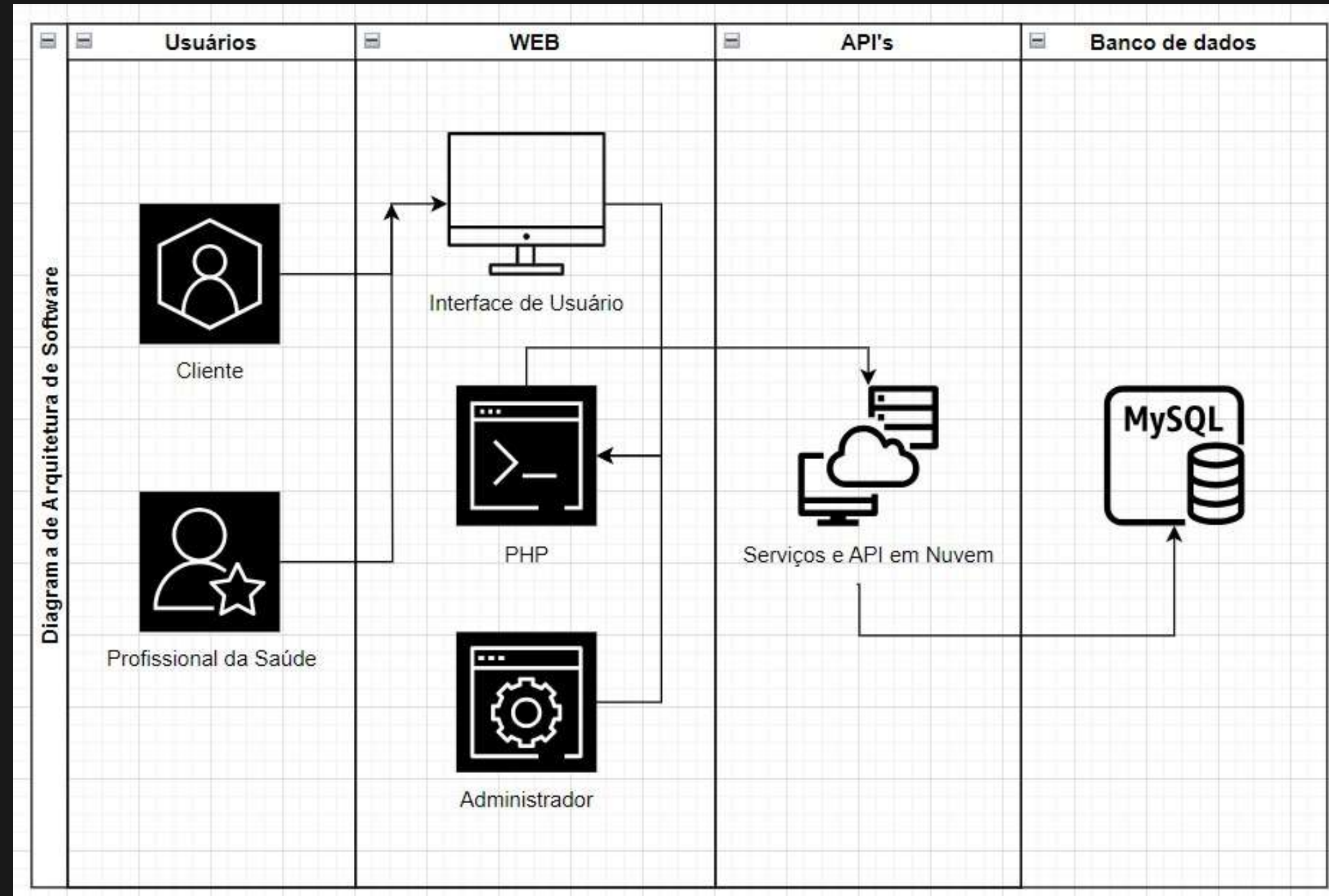
Sistema de Gerenciamento de Vacinas – SGV



Introdução

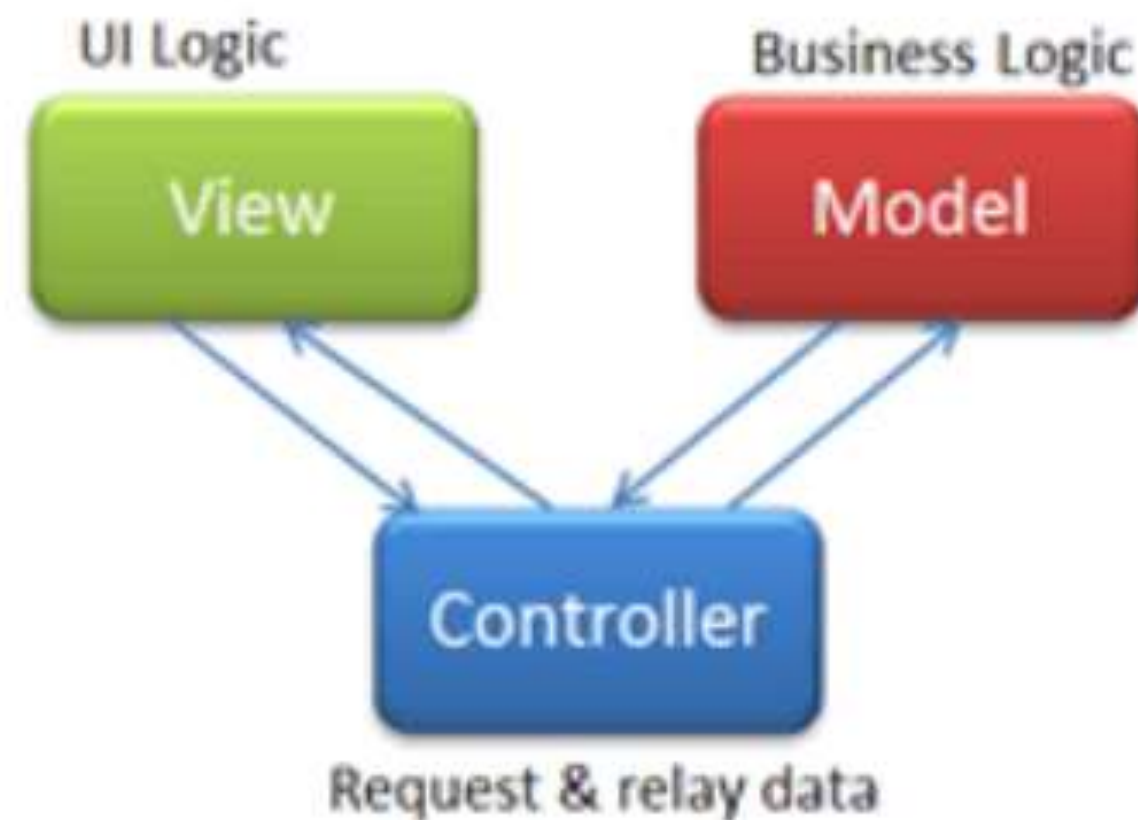
○ **Sistema de Gerenciamento de Vacinas**
é uma solução projetada para facilitar o controle e o acompanhamento das vacinas administradas às pessoas. Visando simplificar o acesso às informações relacionadas à saúde, promovendo a prevenção e a conscientização sobre a importância das vacinas. Com uma abordagem intuitiva e funcional, o sistema oferece diversos recursos: Registro Centralizado de Vacinas e Informações sobre Reações Adversas.

Diagrama de arquitetura de software



Estruturação do Padrão MVC

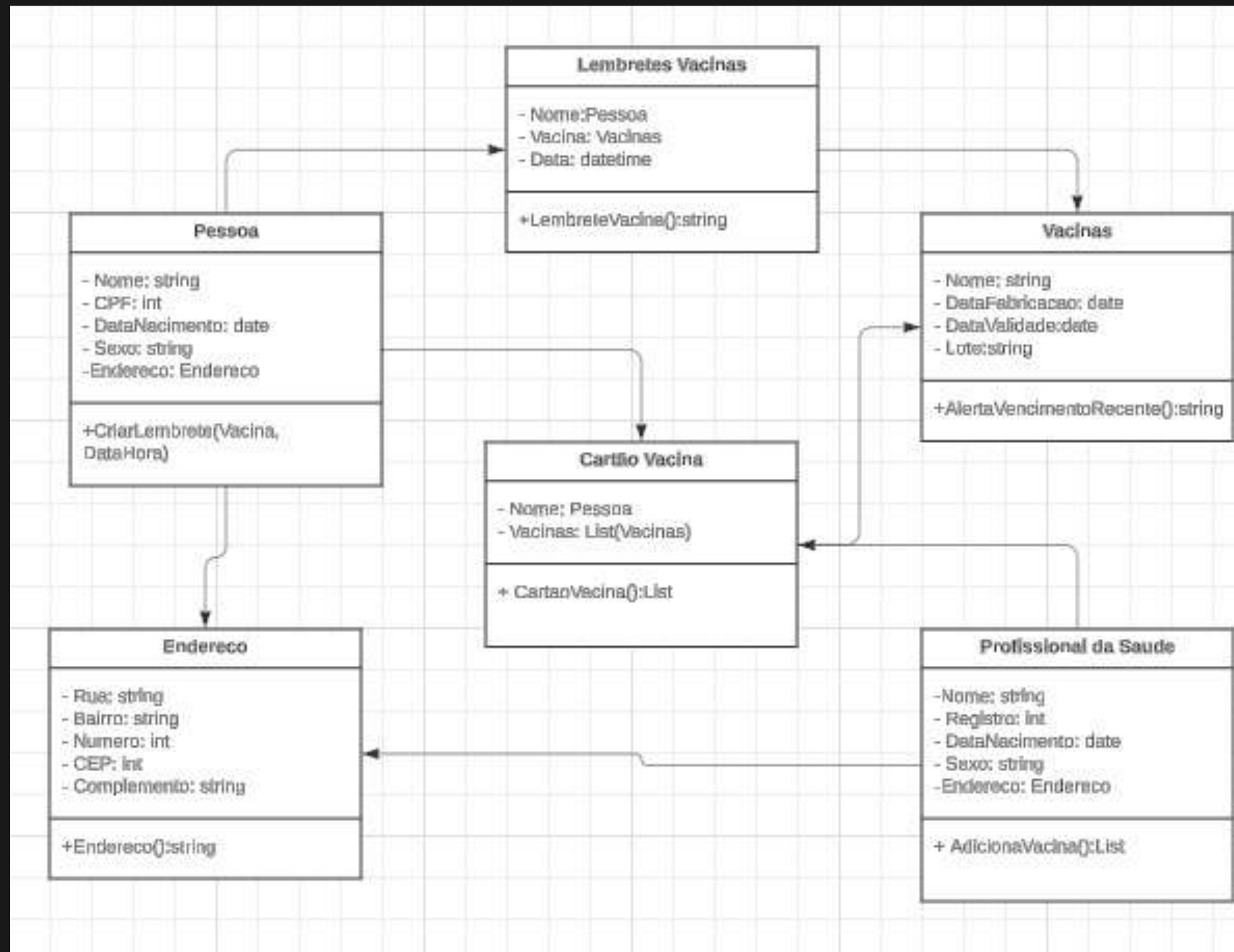
MVC - Model View Controller



Utilizando o modelo MVC, o usuário irá interagir com os controllers, nesse caso sendo as entradas de vacinas, datas e demais dados que o usuário irá utilizar.

O controller irá interagir e receber a resposta da view e model, assim trocando dados entre si para a exibição correta seguindo o modelo de negócio.

Diagramas de classes



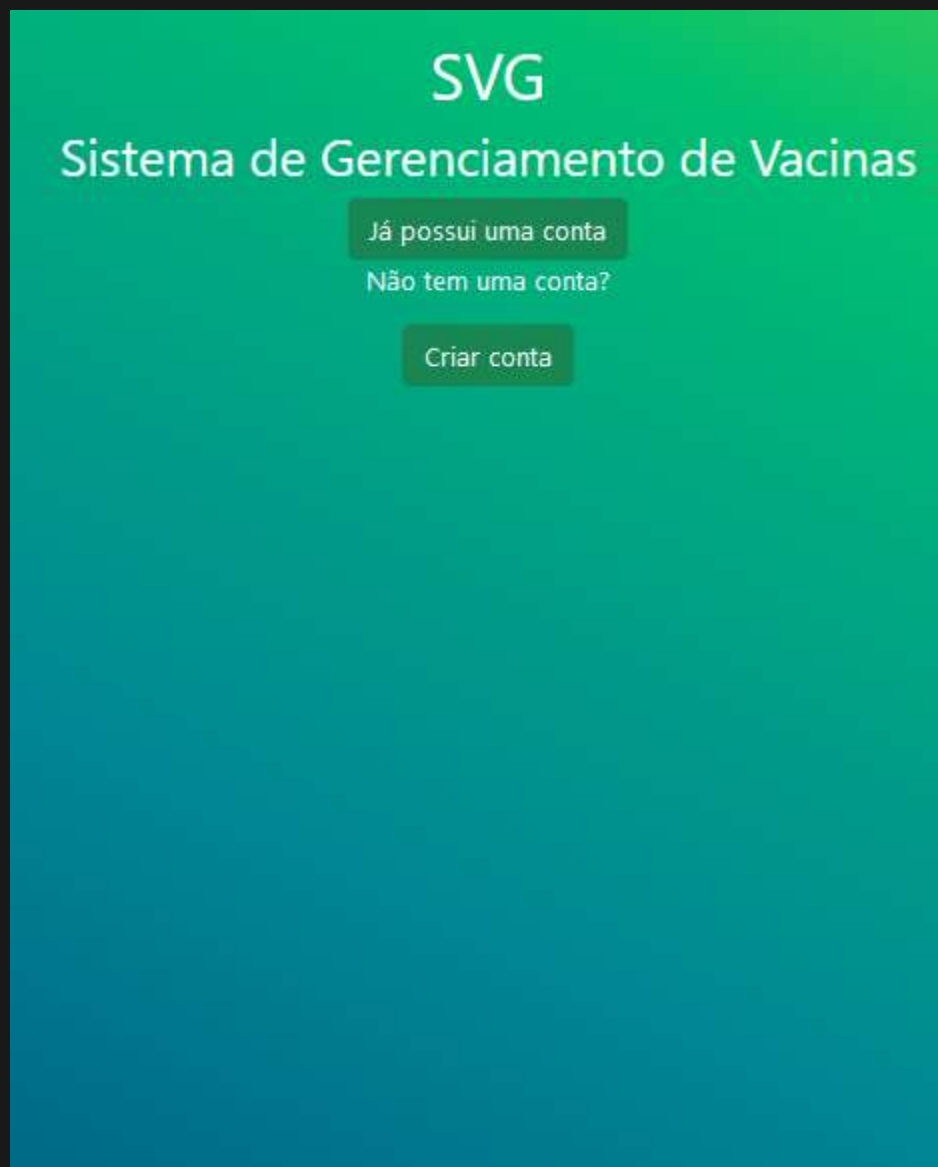
Funcionalidades Implementadas: model

```
1  import { Schema, model } from 'mongoose'
2
3  const UserSchema = new Schema({
4    nome: String,
5    CPF: String,
6    DataNascimento: String,
7    Sexo: String,
8    Endereco: String,
9    Senha: String
10 })
11
12 UserSchema.methods.matchPassword = async function (password) {
13   try {
14     return await password === this.password;
15   } catch (error) {
16     throw new Error(error);
17   }
18 };
19
20 export default model('User', UserSchema)
```

Modelo de usuário desenvolvido

Funcionalidades Implementadas view

```
2  <!doctype html>
3  <html lang="en">
4
5  <head>
6    <meta charset="utf-8">
7    <meta name="viewport" content="width=device-width, initial-scale=1">
8    <title>WhattSaúde</title>
9    <link rel="stylesheet" href="Style.css">
10   <link rel="icon" href="img\wpp.png" type="image/png">
11   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css" rel="stylesheet"
12     integrity="sha384-rbsA2VBKQhggwzxH7pPCaAqO46MgnOM80zW1RWuH61DGLwZJEdK2Kadq2F9CU665" crossorigin="anonymous">
13
14   <script src="MetodosVisuais.js"> </script>
15   <script src="MetodosConfirmacao.js"> </script>
16
17 </head>
18
19 <body>
20
21   <div class="content">
22
23     <div class="logo">
24       
25     </div>
26     <div class="textoPrincipal">
27       <h1>WhattSaúde
28       <h2>Conecte-se com seu médico</h2>
29
30     </h1>
31   </div>
32
33   <div class="formularios">
34     <div id="CriarConta">
35       <h3>Crie sua conta:</h3>
36       <div class="mb-3">
37         <label for="e-mail" class="form-label">E-mail</label>
38         <input type="email" class="form-control" id="emailCriar1" placeholder="Insira seu e-mail">
39       </div>
40       <div class="mb-3">
41         <label for="e-mail" id="labelConfirmEmail" class="form-label">Confirmação do E-mail</label>
42         <input type="email" class="form-control" id="emailCriar2" placeholder="Insira seu e-mail novamente">
```



Visualização tela de
login

Funcionalidades Implementadas: controller

```
1  import User from '../models/User'
2
3  class UserController {
4    async store(req, res) {
5      const {email, name, password, type} = req.body;
6      const user = await User.create({
7        nome,
8        CPF,
9        DataNascimento,
10       Sexo,
11       Endereco,
12       Senha
13     })
14     return res.json(user)
15   }
16
17   async findOne(req, res) {
18     const {email, password} = req.body
19     const query = User.where({email: email})
20     const user = await query.findOne()
21     if(user.password == password){
22       return res.status(200).json(user)
23     }
24     else{
25       return res.status(404).json({
26         message: 'Usuário não encontrado'
27       })
28     }
29   }
30 }
31
32 export default new UserController
```

Controlador do usuário





Trabalho Interdisciplinar: Aplicações Inovadoras

Participantes

- **Lucas Soares Magalhães**
- **Rafael Henrique Freitas de Lima**
- **Robert Leandro Messias**
- **Sérgio da Paz Valladão Júnior**