

 KevinCerqueira / SiMCov2 Private

O SiMCov2 é um sistema de monitoramento de COVID-19 utilizando sensores por meio do protocolo MQTT e uma API REST utilizando o modelo cliente-servidor com socket TCP.

☆ 0 stars 🍴 0 forks

☆ Star

👁 Unwatch ▾

Code

Issues

Pull requests

Actions

Projects

Security

Insights

Settings

🔗 main ▾

...



KevinCerqueira ...

yesterday

[View code](#)

☰ README.md



🔗 SiMCov2 - Sistema de Monitoramento de COVID-19 (v2)

O intuito deste problema foi a criação de uma comunicação utilizando MQTT (além de uma API REST utilizando socket puro), onde sensores como oxímetro, termômetro, esfigmomanômetro e frequencímetro (hardwares) pudessem enviar dados via MQTT para o servidor, e o servidor enviasse via API REST (socket TCP) os dados para o cliente, que nesse caso seria um médico (a), e o mesmo visse os dados de forma ordenada e prioritária as informações e fosse alertado em casos de emergências. SiMCov2 é uma versão melhorada do SiMCov (<https://github.com/KevinCerqueira/SiMCov>)

Tecnologias utilizadas:

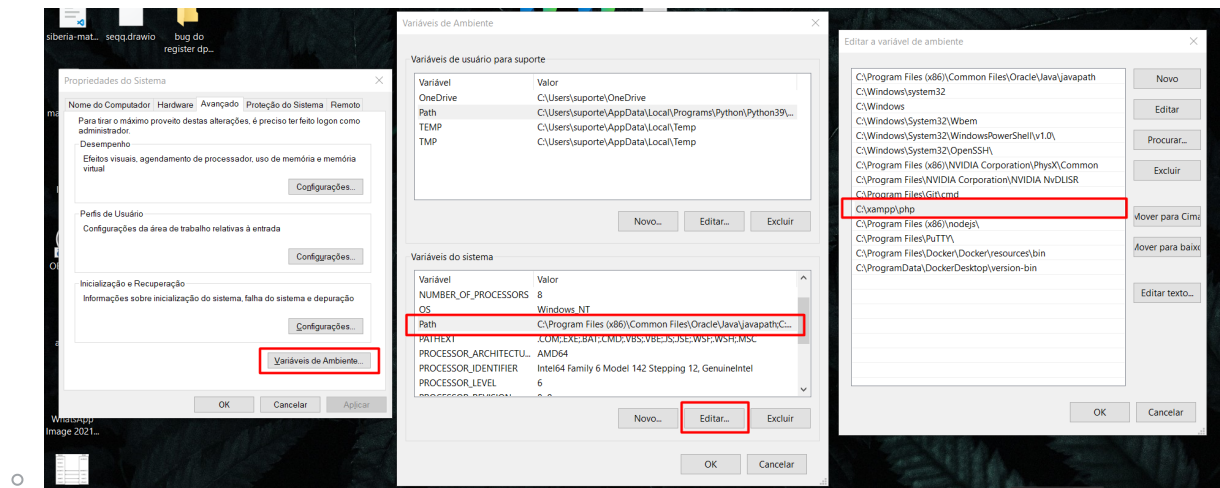
- Python 3.8 (Necessário ser essa versão)
- PHP 7.4.16
- Javascript
- HTML5 & CSS3

Bibliotecas utilizadas

- Python:
 - paho.mqtt.client (comunicação MQTT)
 - pymongo (comunicação com banco de dados MongoDB)
 - socket (servidor com socket puro)
 - TCP e UDP
 - threading (threads)
 - base64 (codificação)
 - re (regex)
 - sys (comandos do sistema)
 - os (pastas e rotas do sistema)
- PHP
 - MqttClient (comunicação MQTT)
 - ConnectionSettings (configuração da comunicação MQTT)
 - funções:
 - socket_create (criar cliente com socket puro)
 - socket_connect (conectar com o servidor)
 - socket_write (enviar dados ao servidor)
 - socket_recv (receber dados do servidor)
 - socket_close (fechar conexão com o servidor)
- Javascript
 - JQuery 3.6.0
 - Ajax
 - SweetAlert2
 - Select2
- HTML5 & CSS3
 - Bootstrap 5.1
 - FontAwesome

Como rodar:

1. Antes de tudo é necessário ter instalado o Python (versão 3.8) e o XAMPP (versão 3.3.0) + PHP (v7.4)
 - Python (v3.8.0): <https://www.python.org/downloads/>
 - XAMPP (v3.3.0) + PHP 7.4: https://www.apachefriends.org/pt_br/download.html
 - **ATENÇÃO:** Baixar o que consta a versão **7.4** do PHP. Não é necessário instalar o php à parte, pois o o mesmo já vem pré configurado no xampp
2. Caso utilize windows, é necessário verificar se o PHP do XAMPP está setado nas variáveis ambiente do seu computador.



3. Para utilizar o socket do PHP é necessário habilitá-lo no php.ini

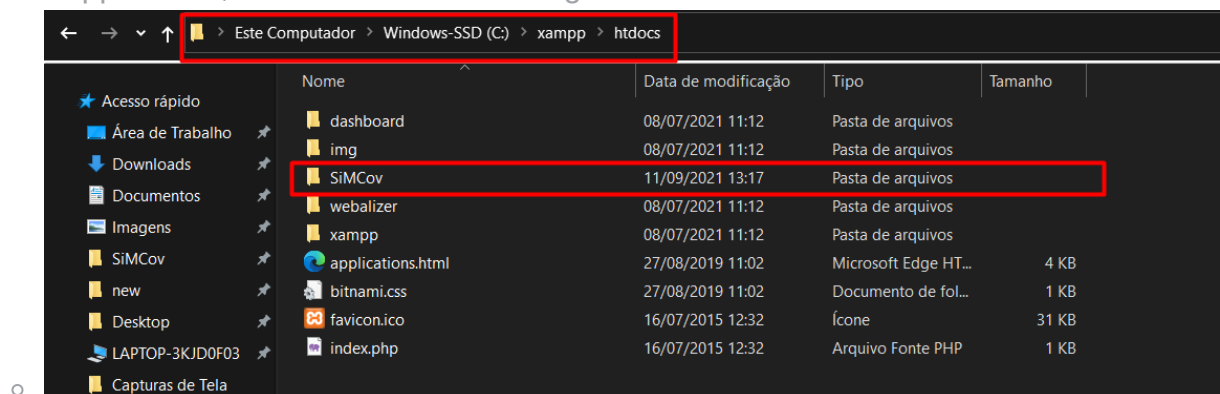
(C:\xampp\php\php.ini):

- Basta pesquisar dentro do arquivo o nome 'sockets' e apagar o ponto e vírgula (;) que fica na frente do mesmo.

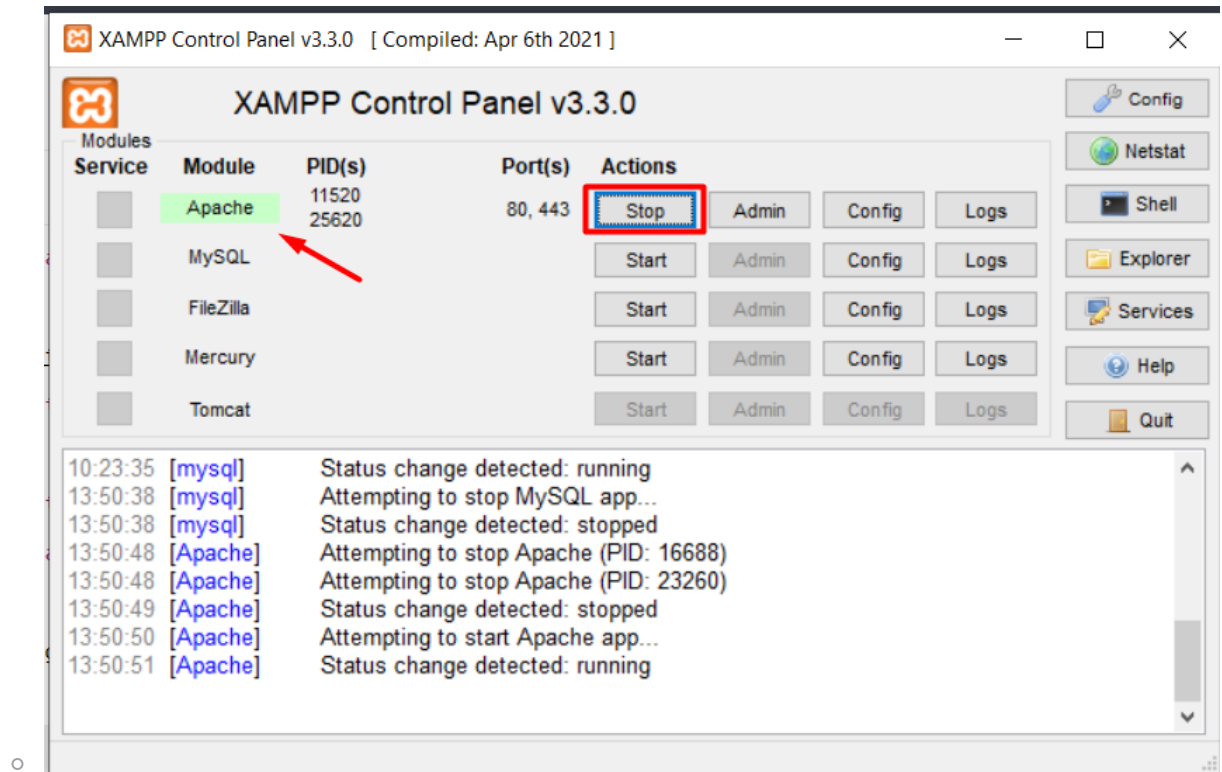


4. Feito tudo isso, confirme que está tudo funcionando, basta abrir o terminal e digitar 'python --version' para verificar se o Python foi instalado corretamente, e para verificar o PHP basta digitar no mesmo terminal 'php -v'. Feito isso confirme se as versões aparecem devidamente.

5. Após configurado, está na hora de mover o repositório para dentro da pasta C:\xampp\htdocs, como mostrado na imagem:



6. Agora abra o XAMPP e dê start na opção Apache (o mesmo deve ficar verde):



7. É necessário instalar as dependências do PHP com o 'composer' (instale ele via <https://getcomposer.org/>), certificado de que esteja instalado, rode o seguinte comando na pasta 'SiMCov/client/Controllers':

```
composer install
```

8. Agora instale as importações necessárias para o python:

```
pip install pymongo
```

```
pip install "pymongo[srv]"
```

```
pip install paho-mqtt
```

9. Após isso, certifique-se que você tem a variável de ambiente CLUSTER no arquivo 'SiMCov/server/.env', como no exemplo do .env.example:

```
CLUSTER=mongodb+srv://<username>:
<password>@kccluster.meacr.mongodb.net/test
```

- caso não tenha acesso ao banco de dados, será necessário nos solicitar, ou criar um no site do MongoDB.

10. Pronto, agora o front-end da aplicação está rodando, agora é a hora de rodar os servidores. Vá no terminal e execute o arquivo 'servertcp.py' e deixe-o rodando, abra um novo terminal para executar 'servermqtt.py' da mesma

```
python servertcp.py
```

```
python servermqtt.py
```

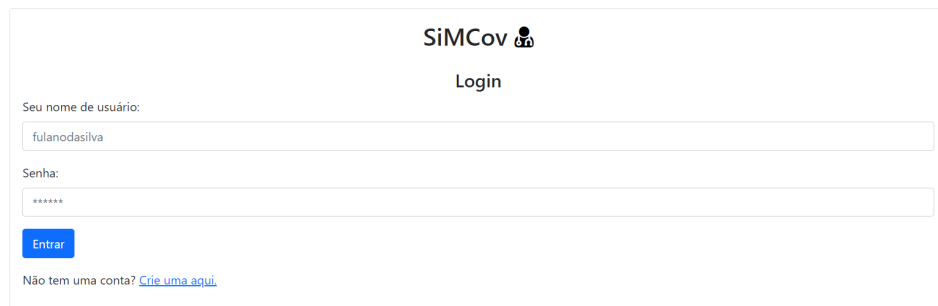
- E caso queira que os sensores fiquem mudando os valores automaticamente, execute o 'simulator.py'

```
python simulator.py
```

11. Pronto! agora é só acessar a tela inicial do sistema, basta acessar o link '<http://localhost/SiMCov2/client>' em um navegador

Telas:

- Entrar: <http://localhost/SiMCov2/client/pages/auth/login.php>



The screenshot shows the login interface for SiMCov2. At the top, it says 'SiMCov' with a user icon. Below that is the word 'Login'. There are two input fields: 'Seu nome de usuário:' with the text 'fulanodasilva' and 'Senha:' with masked characters '*****'. A blue 'Entrar' button is below the password field. At the bottom, it says 'Não tem uma conta? [Crie uma aqui.](#)'

o

- Cadastro: <http://localhost/SiMCov2/client/pages/auth/register.php>



SiMCov

Cadastro

Escolha um nome de usuário:

fulanodasilva

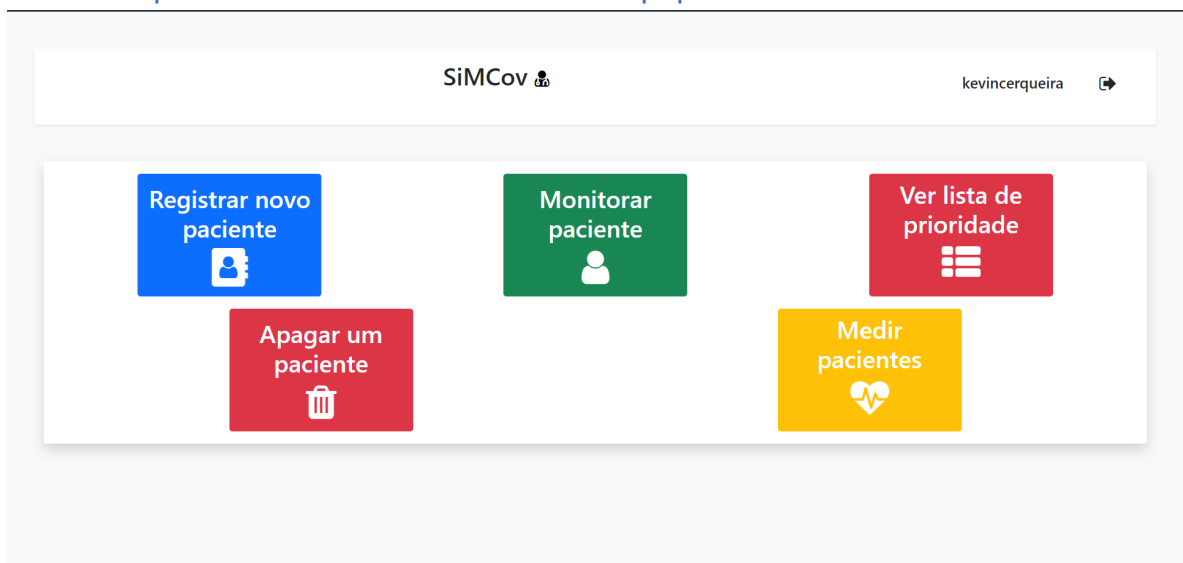
Senha:

Digite novamente a senha:

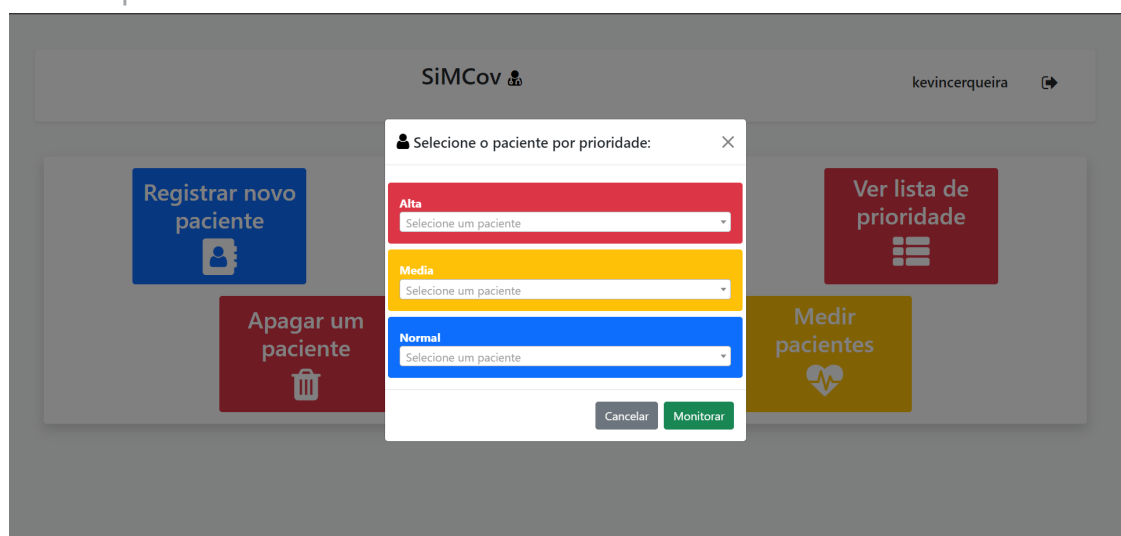
Cadastrar

Já tem uma conta? [Faça login](#)

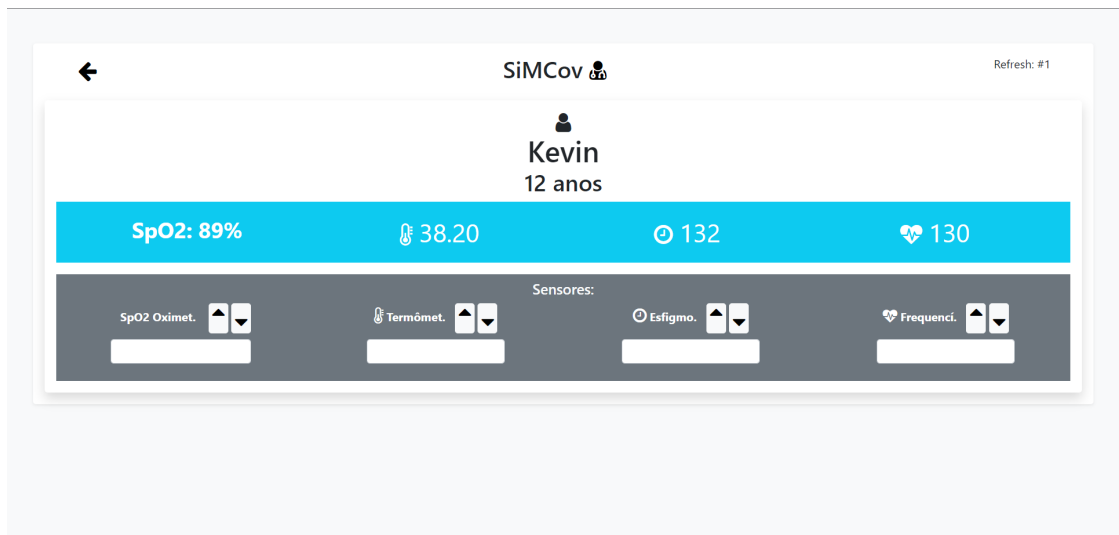
- Dashboard: <http://localhost/SiMCov2/client/index.php>



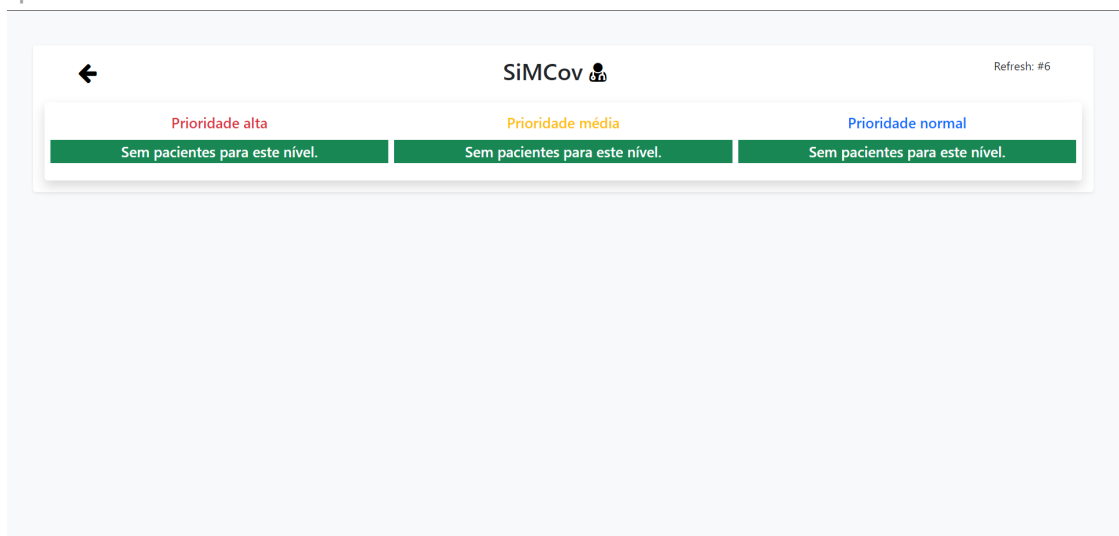
- Monitorar Paciente: Selecione o paciente via modal no dashboard
 - Selecione o paciente no Modal:



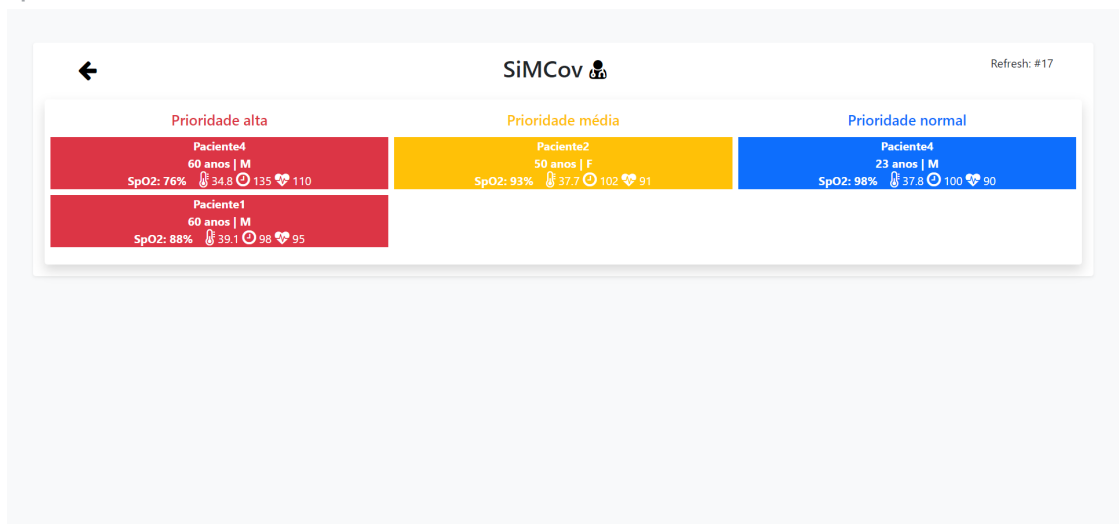
- Após selecionado será redirecionado pra tela de monitoração:



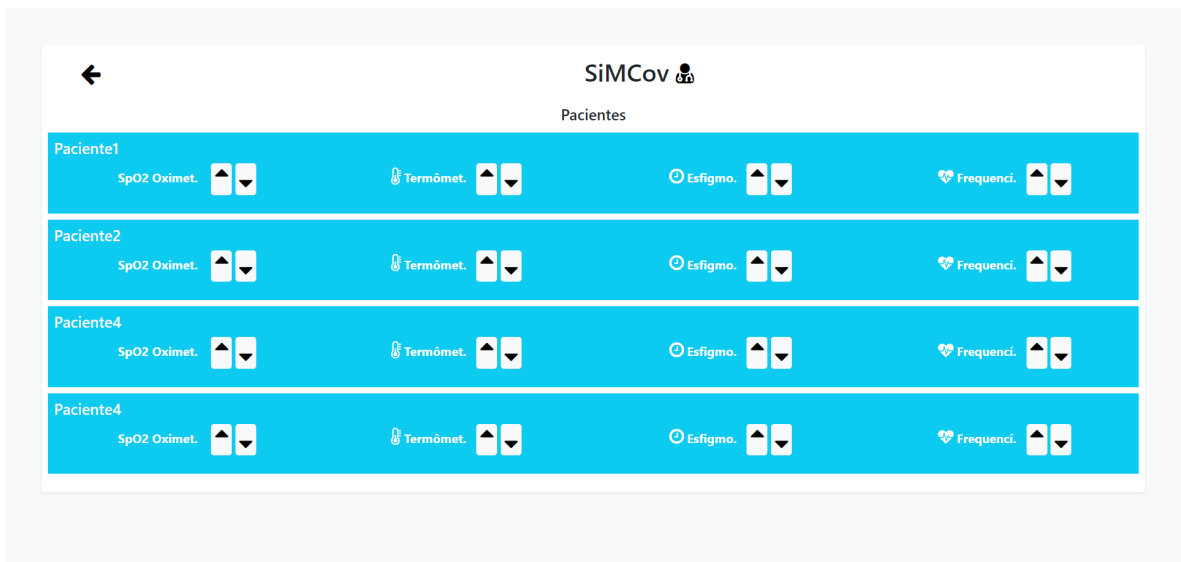
- Lista de prioridade: http://localhost/SiMCov2/client/pages/list_priority.php
 - Sem pacientes ativos:



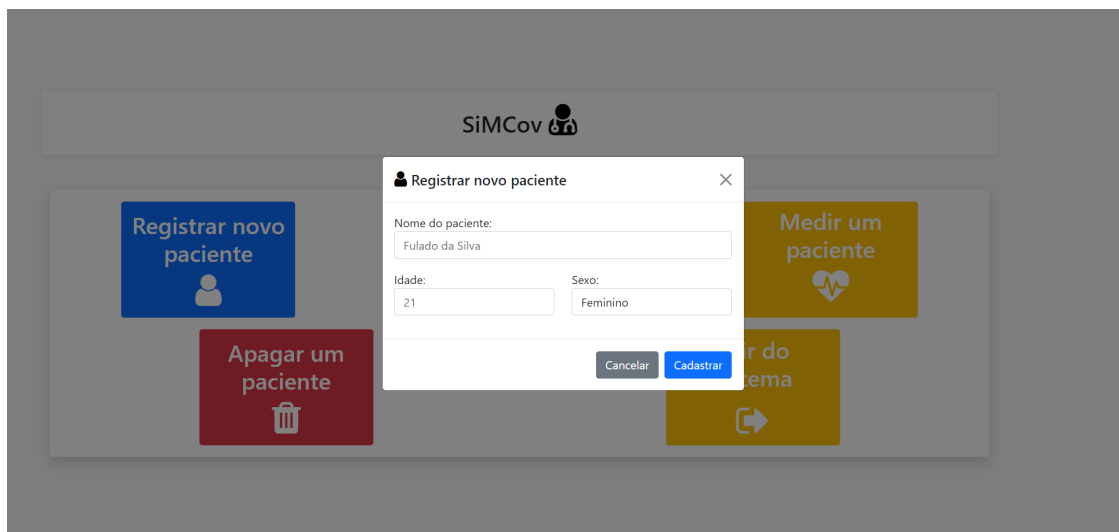
- Com pacientes ativos:



- Medir paciente: http://localhost/SiMCov2/client/pages/change_patients.php

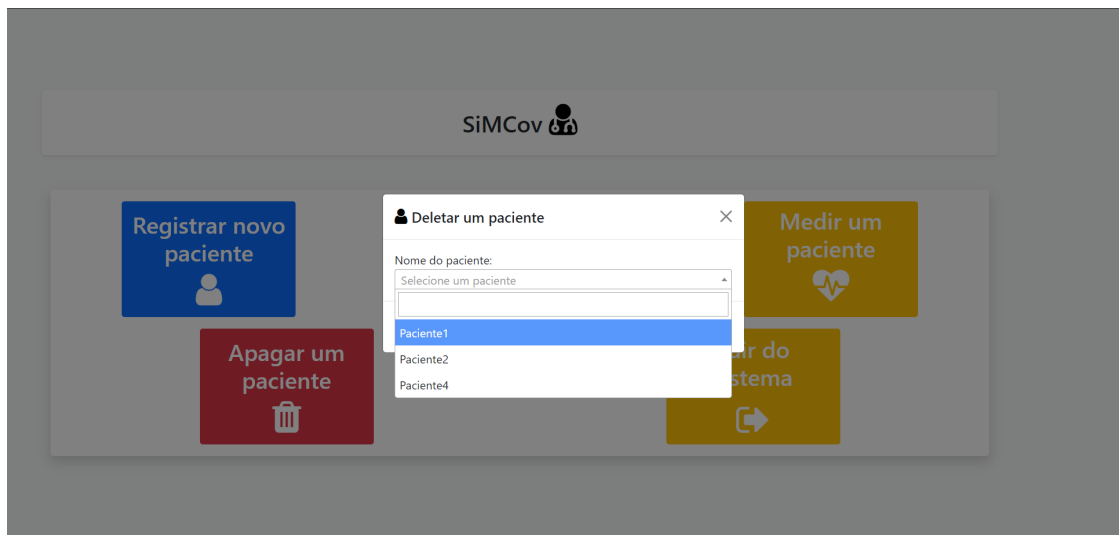


- Cadstrar/Deletar paciente: são modais do dashboard
 - Cadastrar:

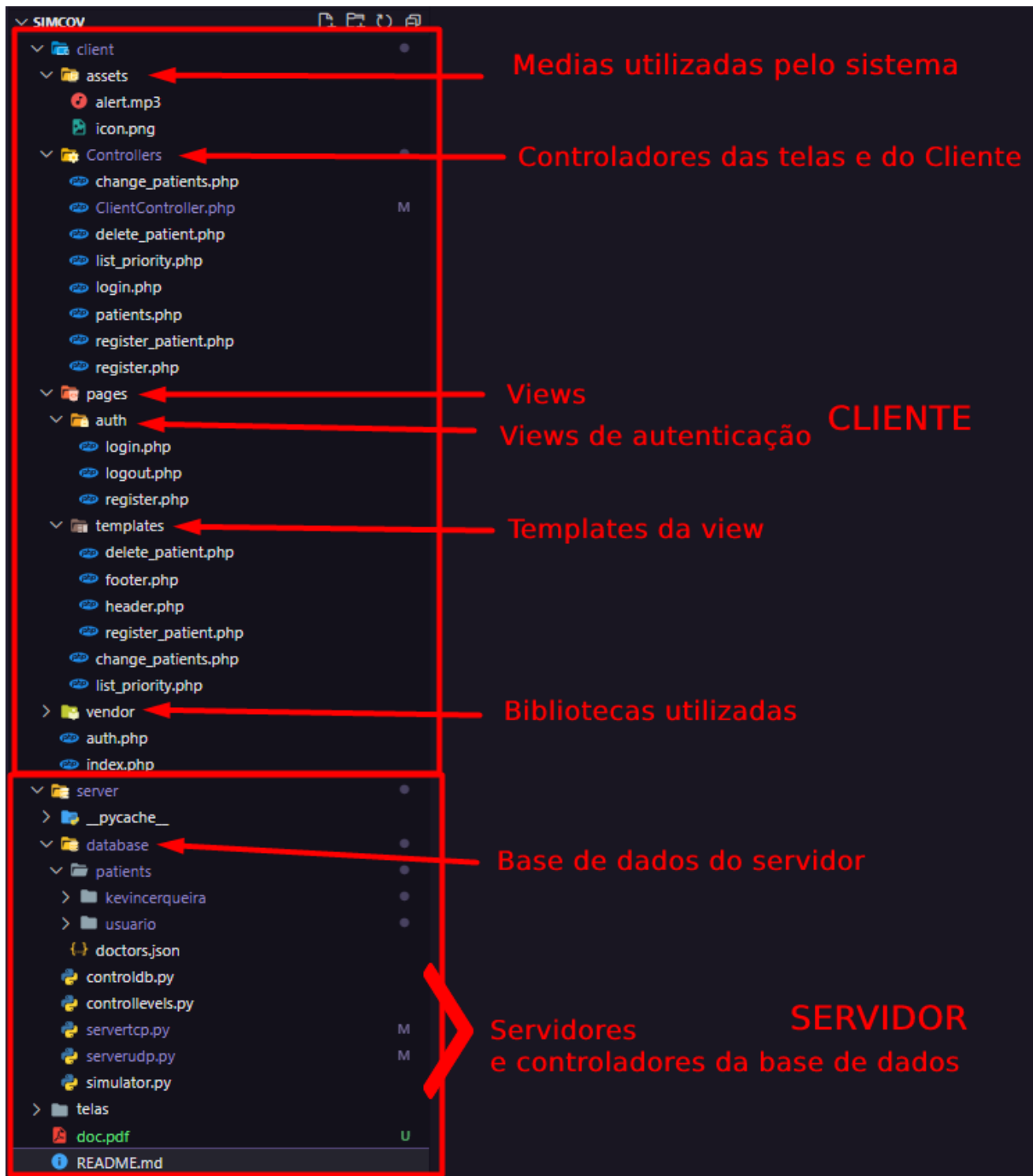


- Deletar:





Construção das pastas



Releases

No releases published

[Create a new release](#)

Packages

No packages published

[Publish your first package](#)

Languages

