

# Critical Monitor

**Elizandro Gabriel Dalastra<sup>1</sup>, Luis Augusto Dias Knob (Orientador), Leandro Mateus Giacomini Rocha<sup>1</sup> (Coorientador)**

<sup>1</sup>Instituto Federal de Educação Ciência e Tecnologia do Rio Grande do Sul - Câmpus Sertão - Rodovia RS 135, Km 25 - Distrito Eng. Luiz Englert - CEP: 99170-000 Sertão-RS

315168@aluno.sertao.ifrs.edu.br, luis.knob@sertao.ifrs.edu.br  
leandro.rocha@sertao.ifrs.edu.br

**Abstract.** This article describes Critical Monitor, a software related to monitoring and counting people through Computer Vision, that will be developed as a partial fulfillment of the requirements for the undergraduate degree in Analysis and Systems Development, from the Federal Institute of Education, Science and Technology of Rio Grande do Sul (IFRS) - Campus Sertão. The System aims to monitor critical situations of SARS-CoV-2 virus contagion, in order to generate alerts about non-compliance with the maximum occupancy limits in indoor environments and social distance. The work resulted in a web application of easy implementation and usability, fulfilling all the proposed requirements.

**Resumo.** O presente artigo descreve o Critical Monitor, um software referente a monitoramento e contagem de pessoas por meio de Visão Computacional, desenvolvido como Trabalho de Conclusão de Curso de Análise e Desenvolvimento de Sistemas, do Instituto Federal de Educação Ciência e Tecnologia do Rio Grande do Sul - Câmpus Sertão. O Sistema objetiva monitorar situações críticas de contágio do vírus SARS-CoV-2, para assim gerar alertas sobre descumprimentos das normas de lotação máxima em ambientes internos e distanciamento social. O trabalho resultou em uma aplicação web de fácil implementação e usabilidade cumprindo todos os requisitos propostos.

## 1. Introdução

Em 11 de março de 2020, a Organização Mundial da Saúde (OMS) anunciou que COVID-19, uma doença causada por um novo tipo de coronavírus chamado SARS-CoV-2 (*Severe Acute Respiratory Syndrome Coronavirus 2*), também chamada de novo coronavírus, é uma doença pandêmica. No momento dessa declaração, a epidemia, que começou em Wuhan, na China, já havia aparecido em 114 países, territórios ou regiões, atingindo 118.319 casos e 4.292 mortes pela doença [WHO 2020]. No Brasil, o primeiro caso de COVID-19 foi confirmado em São Paulo em 26 de fevereiro de 2020 [MS 2020].

A rápida disseminação do novo coronavírus entre populações e comunidades, resultante do alto índice de transmissão, juntamente com a falta de vacinas antivirais até então, tornou os métodos não farmacológicos como opções mais eficientes no controle da COVID-19. Seguindo a orientação de outros países que controlam com sucesso a pandemia, vários estados e cidades brasileiras adotaram medidas de distanciamento social para reduzir o contato entre as pessoas e controlar a propagação do vírus. Essas

medidas incluem o cancelamento de eventos públicos, o fechamento de escolas e empresas e a recomendação de que as pessoas fiquem em casa [Aquino 2020].

Tendo em vista esses pontos e com o intuito de auxiliar instituições e empresas que necessitam essencialmente funcionar no modelo presencial, este artigo visa apresentar o Trabalho de Conclusão de Curso (TCC) de Tecnologia em Análise e Desenvolvimento de Sistemas, do Instituto Federal de Educação Ciência e Tecnologia do Rio Grande do Sul (IFRS) - Câmpus Sertão. Este trabalho se propõe a desenvolver um *software* aplicativo, denominado Critical Monitor, capaz de automatizar o controle e monitoramento de aglomerações em ambientes fechados, utilizando recursos de Visão Computacional (VC).

Por meio de uma aplicação *web* desenvolvida com o *micro-framework* Flask, o Critical Monitor é capaz de detectar pessoas através de um modelo de Inteligência Artificial, de forma performática, visto que, utiliza a plataforma Tensorflow Lite para efetuar a predição em imagens 2D. A aplicação também cria uma abstração de uma “zona segura”, para aferir o distanciamento social, e armazena esses dados em um banco de dados relacional PostgreSQL, apresentando às pessoas detectadas o status do ambiente, alertas e histórico de ocorrências em tempo real.

O presente artigo está dividido em 7 seções que descrevem cada processo da elaboração do *software*. Na seção 2 é apresentado o problema enfrentado no controle da doença e como a solução pode auxiliar. Na seção 3 são expostos os objetivos que o trabalho visa atender. Adiante, na quarta seção, são listados trabalhos relacionados, descrevendo similaridades e diferenças entre o trabalho desenvolvido. Posteriormente, na seção 5, metodologia, requisitos funcionais e não funcionais, tecnologias utilizadas e casos de usos são descritos. A sexta seção descreve a arquitetura de software utilizada, modelagem de dados, arquitetura de Visão Computacional e interface com o usuário. E, por fim, a seção 7 destaca os trabalhos futuros e melhorias que o Critical Monitor poderá ter.

## **2. Definição de problema e justificativa**

Para mitigar o espalhamento da doença infecciosa causada pelo SARS-CoV-2, foi proposta a adoção de distanciamento social, que consiste em uma estratégia de restrição do convívio social. Nesse sentido, essa estratégia é uma das medidas mais importantes e eficazes no combate do avanço da pandemia da COVID-19 [WHO 2020].

O Centro de Desenvolvimento da Tecnologia Nuclear (CDTN), instituição ligada à Comissão Nacional de Energia Nuclear (CNEN) e o Ministério de Ciência, Tecnologia e Inovação (MCTI) fizeram um estudo, publicado na revista *Environmental Research*, sobre as partículas microscópicas e invisíveis e sua presença no ar, observando como o coronavírus locomove-se visto que possui baixa massa.

O estudo demonstrou maior probabilidade da disseminação do novo coronavírus em ambientes internos com circulação de ar insuficiente. Nesses ambientes, o maior

índice de infecção deve-se à baixa troca de ar interno com externo aliada à presença de pessoas tossindo, espirrando, falando ou espirrando vigorosamente, especialmente quando os protocolos de uso de máscara e distanciamento não são respeitados [Passos 2021].

Além disso, segundo a OMS, pessoas completamente vacinadas com os imunizantes do coronavírus, devem continuar seguindo todos os protocolos não farmacológicos de distanciamento social e uso de máscara, isso porque, mesmo vacinado, ainda há a possibilidade de contrair o vírus, de forma assintomática ou não e, portanto, continuar espalhando a doença para pessoas não vacinadas [OMS 2022].

### **3. Objetivos**

#### **3.1 Geral**

- Desenvolver um software que auxilie estabelecimentos comerciais, indústrias, instituições de ensino e qualquer outro prestador de serviço, que necessite funcionar essencialmente de forma presencial, a monitorar circulação, distanciamento mínimo e lotação máxima das pessoas presentes em ambientes internos, através de um modelo de predição capaz de identificar e rastrear pessoas em imagens digitais utilizando Visão Computacional.

#### **3.2 Específicos**

- Desenvolver sistema de autenticação de usuário.
- Desenvolver um sistema de cadastro para usuários.
- Desenvolver um sistema que capture imagens a partir de uma câmera com processamento de imagens em 2D.
- Fornecer uma ferramenta de calibração para que o sistema consiga aferir a distância entre pessoas de forma assertiva.
- Implementar uma interface simples contendo dados sobre risco de contaminação, contagem de pessoas e média de ocupação por horário.
- Controlar histórico de ocorrências.
- Permitir que o usuário configure disparo de alertas em caso de situações críticas.
- Controlar dados de usuário, configurações e histórico.

### **4. Trabalhos relacionados**

Para facilitar o controle da quantidade de pessoas ocupando um espaço existem algumas soluções tecnológicas no mercado, que através de sensores ou câmeras conseguem aferir quantas pessoas entram e saem. Soluções com câmeras são muito versáteis, visto que podem ser instaladas facilmente em qualquer lugar. *Softwares* de VC possibilitam a

criação de redes neurais convolucionais capazes de classificar, detectar e segmentar imagens.

Inteligência Artificial (IA) refere-se a sistemas ou máquinas que visam imitar a inteligência humana para executar tarefas triviais de forma automatizada, tomada de decisões, e podem se aprimorar pelo processo iterativo. IAs aprendem a partir de um processo de treinamento utilizando bases de dados que contenham diversos atributos e esses atributos geram um resultado, sendo assim, a IA consegue entender os padrões [Oracle 2022].

VC é uma subárea de Inteligência Artificial (IA) que permite aos computadores simular visão e entendimento de cenários tridimensionais a partir de imagens bidimensionais, vídeos e outras entradas visuais. Utilizando VC pode-se treinar modelos de predição, capazes de detectar objetos, pessoas e qualquer outro padrão [IBM 2021].

Um exemplo é a solução da empresa 3RCorp – Internet, Sistemas e Tecnologia denominada Câmera Inteligente [3RCorp 2021], capaz de realizar a contagem de pessoas através de uma câmera com processador interno, cujo sistema é instalado no próprio *hardware*. A Câmera Inteligente se comunica com aplicativos tanto móveis quanto *desktop*, os quais apresentam dados sobre o monitoramento e a transmissão das imagens capturadas pela câmera, conforme a Figura 1 apresenta.



**Figura 1: Câmera Inteligente 3RCorp e seu aplicativo móvel de monitoramento.**  
**Fonte: 3RCorp, 2021.**

Outro produto de *software* bem consolidado no mercado é o Amazon Rekognition da plataforma Amazon Web Services [AWS 2021]. O Amazon Rekognition possui implementações já prontas de visão computacional para detecção de pessoas, tanto em vídeo quanto em imagens, como mostra o exemplo da Figura 2.



**Figura 2: Detecção de pessoas e veículos utilizando Amazon Rekognition.**

**Fonte: AWS, 2021.**

O serviço é fornecido através de servidores em nuvem, o que impossibilita a usabilidade do serviço para monitoramento em tempo real, por conta da latência de tráfego das imagens, e também da dependência de uma conexão de internet estável.

Por mais que essas soluções sejam capazes de contabilizar o fluxo de pessoas, elas não estão aptas a monitorar o distanciamento social durante a permanência dos indivíduos dentro do ambiente. Neste sentido, Critical Monitor é uma solução que facilmente se integrará com um sistema simples de câmera para inferir o distanciamento social de forma automatizada, sem a necessidade de alocar uma pessoa para aferir manualmente se as medidas de redução do contágio da Covid-19 estão sendo cumpridas.

O Critical Monitor contará com uma interface *web* com a qual o usuário poderá configurar a lotação máxima do ambiente, distanciamento mínimo e configurar diversos ambientes para monitorar.

## 5. Metodologia

Nesta seção serão descritos de forma mais detalhada a metodologia abordada, arquitetura, requisitos funcionais e não funcionais, casos de uso e tecnologias utilizadas na execução do desenvolvimento do sistema Critical Monitor.

### 5.1 Descrição do *software*

O *software* aplicativo denominado Critical Monitor, deve ser capaz de informar aos usuários sobre a situação de risco no ambiente em que o sistema está monitorando, a partir da contagem de pessoas detectadas no espaço e predição de distanciamento entre elas, através de imagens de uma câmera de monitoramento previamente cadastrada pelo usuário

A aplicação fornece parâmetros configuráveis ao usuário, lotação máxima do espaço monitorado, distanciamento mínimo entre pessoas. Logo, o sistema informa

através de contadores o número atual de pessoas no quadro de vídeo, máximo de pessoas registradas naquela sessão de monitoramento e a possibilidade de alterar os parâmetros de capacidade máxima e distanciamento mínimo ao mesmo tempo, para assim ter uma calibração assertiva do monitoramento.

Além disso, o usuário poderá visualizar todo o histórico de ocorrências registradas separadas por dia. Uma ocorrência é determinada pelo sistema em duas situações: quando o número de pessoas detectadas extrapola a lotação máxima e quando o sistema afere que o distanciamento mínimo entre duas pessoas foi violado.

Todos os casos ocorridos serão listados em uma tabela informando data, hora e tipo de ocorrência, e quando selecionada serão exibidas informações de quantas pessoas desrespeitam o distanciamento e total de ocupação naquele momento.

Outra funcionalidade disponibilizada pelo sistema é a opção de salvar dados de configuração e histórico em nuvem a fim de recuperá-los em caso de perda. Para ter acesso a esse recurso, o usuário deve se cadastrar no sistema informando um nome de usuário, nome completo e senha.

## 5.2 Requisitos funcionais

Segundo Sommerville (2011, p.59) “requisitos funcionais são declarações de serviços que o sistema deve fornecer, de como o sistema deve reagir a entradas específicas e de como o sistema deve se comportar em determinadas situações. Em alguns casos, os requisitos funcionais também podem explicitar o que o sistema não deve fazer”. Nos seguintes tópicos descrevem-se os requisitos funcionais analisados da aplicação a ser desenvolvida:

- Manter usuário.
- Autenticar usuário.
- Controlar configurações.
- Controlar histórico de ocorrências.
- Controlar a calibração da câmera.
- Capturar imagens da câmera.
- Detectar pessoas em imagens.
- Aferir o distanciamento entre pessoas detectadas.
- Permitir monitorar ambientes em tempo real.
- Disparar alertas para o usuário.

## 5.3 Requisitos não funcionais

Para Sommerville (2011, p. 59),

“Requisitos não funcionais são restrições aos serviços ou funções oferecidos pelo sistema. Incluem restrições de *timing*, restrições no processo de desenvolvimento e restrições impostas pelas normas. Ao contrário das características individuais ou serviços do sistema, os requisitos não funcionais, muitas vezes, aplicam-se ao sistema como um todo.”

Desta forma, os tópicos abaixo apresentam os requisitos não funcionais do sistema Critical Monitor:

- A aplicação será desenvolvida baseada no micro *framework* Flask.
- O SGBD (Sistema Gerenciador de Banco de Dados) utilizado será o PostgreSQL.
- Para realização de predições em imagens será utilizada a plataforma de Aprendizado Profundo TensorFlow.
- A biblioteca OpenCV será responsável pelo pré-processamento das imagens.
- Para testes de validação deve-se utilizar o *framework* Pytest.
- O desenvolvimento dos componentes de interface será auxiliado pelas bibliotecas Bootstrap e Jquery.
- Para controle e versionamento de código fonte será utilizada a ferramenta Git hospedando na plataforma Github.

A escolha do Flask como *framework web* se dá pelo fato de ser pequeno e leve. O Flask utiliza o mecanismo de modelo Jinja para desenvolver dinamicamente páginas HTML utilizando a linguagem Python [The Pallets Projects 2021].

O PostgreSQL é o SGBD relacional de código aberto mais avançado atualmente do mercado, sua escolha deve-se pela sua fácil instalação e sua integração com diversas ferramentas de desenvolvimento incluindo o Flask [PostgreSQL 2021].

O TensorFlow é uma plataforma de aprendizado de máquina completa de código aberto e conta com um ecossistema abrangente e flexível [TensorFlow 2021]. O software conta com a utilização da plataforma pela razão de possibilitar o uso de APIs intuitivas de alto nível e a implementação da arquitetura SSD (*Single Short Multibox Detector*), que é uma rede neural de convolução única que aprende a prever a localização e classificação de objetos e pessoas a cada quadro de vídeo.

A escolha da biblioteca OpenCV deve-se pelo motivo de disponibilizar diversos métodos e funções para processamento e manipulação de imagens, entre as principais funcionalidades estão: redimensionamento, colheita de áreas específicas, inserção de caracteres, linhas e retângulos na imagem [OpenCV 2021].

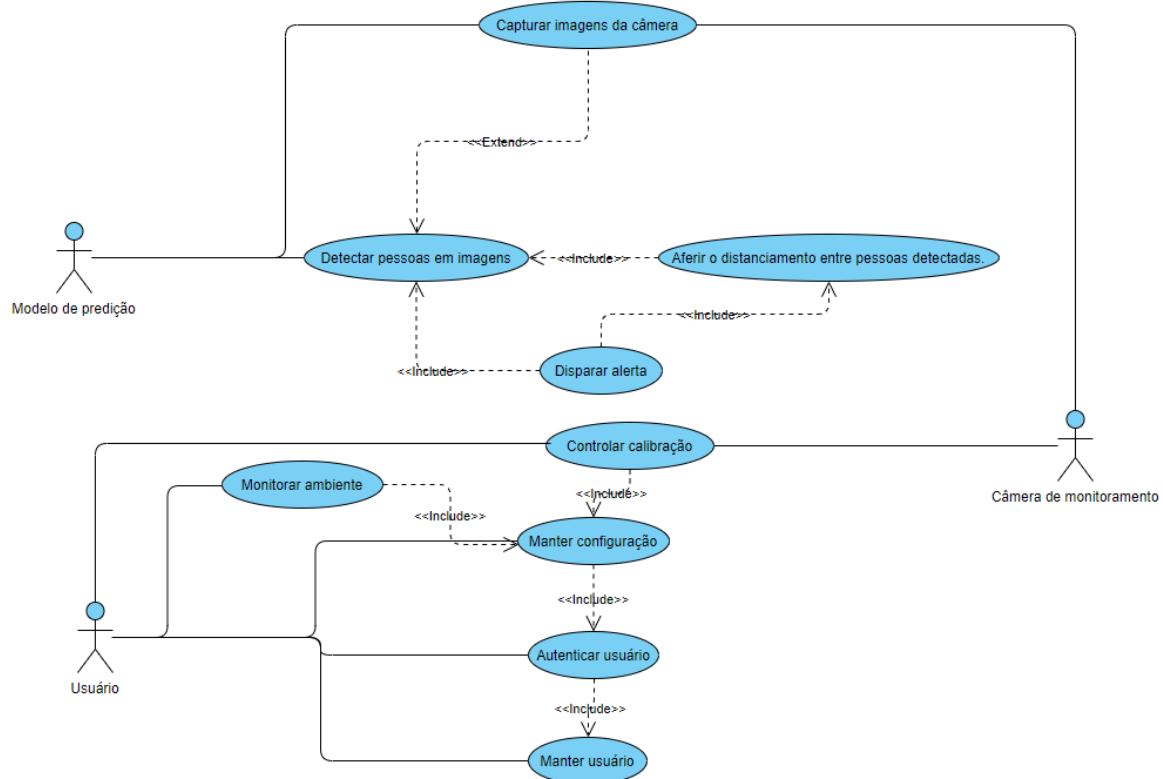
O motivo para ter o Pytest como ferramenta de teste, deve-se ao fato de prover diversas soluções para execução de testes de validação de forma prática e eficiente [Pytest 2021].

O Jquery trata-se de uma biblioteca que simplifica funções do JavaScript que interagem com o HTML, a utilização dele promove códigos mais limpos e organizados [jQuery 2021].

O *Front-End* deve ser construído baseando-se no design do Bootstrap, um *framework* que possui componentes responsáveis, para tipografia de um website com interface simples e de boa usabilidade [Bootstrap, 2021].

## 5.4 Diagrama de casos de uso

Segundo Pressman (2011, p.137), “um caso de uso conta uma história estilizada sobre como um usuário final (desempenhando um de uma série de papéis possíveis) interage com o sistema sob um conjunto de circunstâncias específicas”. Na Figura 3, é representado o diagrama de casos de uso.



**Figura 3. Diagrama de casos de uso da aplicação Critical Monitor.**

### 5.5.1 Descrição dos casos de uso

Alguns casos de uso podem ser descritos mais detalhadamente, a fim de entender o entendimento sobre alguns cenários mais complexos. A seguir são apresentados os requisitos mais relevantes de maneira aprofundada. Na Tabela 1 estende-se a descrição de “Manter configurações”.

**Tabela 1. Descrição do caso de uso “Manter configurações”.**

Nome	Manter configurações
Atores	Usuário
Descrição	Permite que o usuário configure a lotação máxima de pessoas e tempo limite permitido de distanciamento desrespeitado.

Pré-Condições	Estar autenticado no sistema.
Pós-Condições	
Ação dos Atores	Resposta do sistema
1. O usuário sem ambientes configurados realiza autenticação no sistema, ou seleciona a opção “Salas” no cabeçalho de navegação.	2. Abre a página com a listagem de salas.
3. O usuário clica no botão “Nova Sala”.	4. O sistema mostra um formulário de cadastro de informações na mesma página em que o usuário está.
5. Informe os seguintes dados: Nome da sala, capacidade máxima e endereço IP da câmera.	
6. Clicar em “Acessar Câmera”	7. O sistema exibe um quadro de vídeo capturado pela câmera informando que o usuário deve selecionar quatro pontos na imagem para identificar a área de análise.
8. O usuário clica nos 4 pontos desejados.	9. O sistema desenha as linhas de intersecção de cada ponto.
10. O usuário clica em “Enviar”	11. O sistema redireciona para a página “Home”.

A Tabela 2 descreve o requisito “Realizar predição em imagens”.

**Tabela 2. Descrição do caso de uso “Realizar predição em imagens”.**

Nome	Realizar predição em imagens
Atores	Câmera, Modelo de predição
Descrição	Todas as imagens capturadas pela câmera serão submetidas à inferência do modelo de predição, que irá detectar a localização de pessoas no cenário.
Pré-Condições	A câmera deve estar calibrada.
Pós-Condições	Receber alertas em casos de ocorrência.

A Tabela 3 apresenta o requisito “Controlar calibração da câmera”.

**Tabela 3. Descrição do caso de uso “Manter calibração da câmera”.**

Nome	Manter calibração da câmera.
Atores	Usuário, Câmera.
Descrição	O usuário, através de uma ferramenta gráfica do sistema, deve selecionar 4 pontos na imagem em forma de retângulo, que abstraem os 4 cantos do chão na ordem canto superior esquerdo, canto superior direito, canto inferior direito e canto inferior esquerdo.
Pré-Condições	Estar autenticado no sistema.

Por fim, a tabela 4 descreve o requisito “Permitir monitorar ambientes em tempo real.”

**Tabela 4. Descrição do caso de uso “Permitir monitorar ambientes em tempo real.”**

Nome	Permitir monitorar ambientes em tempo real.
Atores	Usuário.
Descrição	Através de uma página de monitoramento o usuário poderá selecionar uma das salas previamente cadastradas e observar em tempo real os alertas e a detecção de pessoas no vídeo.
Pré-Condições	Estar autenticado no sistema e ter ao menos uma sala cadastrada.

## 6. Projeto

Nesta etapa serão descritas as tecnologias utilizadas, arquitetura do software e modelagem dos dados.

### 6.1 Framework de desenvolvimento

Para o desenvolvimento do software aplicativo, foi escolhido o micro-framework Flask. Um micro-framework é um framework modularizado que possui uma estrutura simplificada, comparado aos frameworks tradicionais, no qual funcionalidades mais complexas demandam a instalação de demais módulos, como conexão com banco de dados, renderização de páginas e entre outros.

Desenvolvido e lançado em 2010 por Armin Ronacher e atualmente pertencente à coleção de bibliotecas de código-fonte aberto The Pallets Projects, o Flask é destinado para o desenvolvimento de aplicações mais simples e com requisitos básicos. Portanto, o Flask possui um núcleo mais simplificado, que contém apenas os recursos necessários para sua execução, sendo assim é necessário a instalação de pacotes específicos para funcionalidades mais complexas [The Pallets Projects 2021]. Por ser um *micro-framework*, seus requisitos de funcionamento são reduzidos, facilitando a implantação do sistema projetado.

A opção pela utilização do Flask deve-se principalmente por trabalhar exclusivamente com a linguagem de programação Python que possui implementações de diversas bibliotecas necessárias para as funcionalidades propostas. Por exemplo, o Numpy é uma biblioteca para realização de qualquer tipo de operação matemática em matrizes, Keras/TensorFlow responsáveis pela inferência do modelo e o OpenCV utilizado para efetuação de operações em imagens.

Outro motivo pela escolha do Flask é graças à sua simplicidade na construção de softwares pequenos e rapidez no desenvolvimento, pois não necessita de muitas configurações para poder começar a funcionar e sua robustez que, por mais simples que o *framework* possa ser, é possível criar projetos altamente personalizados e, em caso de expansão do *software*, a criação de uma arquitetura mais complexa.

## 6.2 Funcionalidades e pacotes adicionais

Para possibilitar a construção de templates, o Flask implementa o Jinja2 que é um sistema de *web templates* escrito em Python e que facilita a criação de páginas dinâmicas. O Jinja2 permite a utilização de variáveis, estruturas condicionais e de repetição, reaproveitamento de componentes de *layout* por meio de herança e importação de páginas, escrevendo na linguagem Python diretamente em arquivos HTML. Tem suporte completo a Unicode e, para aplicativos que necessitam de maior segurança, adiciona execução em área restrita e escape automático [Jinja2, 2021].

Para conexão com banco de dados, foi adicionado o módulo SQLAlchemy, que é um kit de ferramentas Python SQL (*Structured Query Language*) e *Object Relational Map* (ORM). Um ORM traduz classes Python em tabelas do banco de dados relacionais, convertendo automaticamente a linguagem *Pythonic SQLAlchemy Expression Language* em instruções SQL, permitindo a escrita de consultas a bancos de dados relacionais usando a sintaxe Python SQLAlchemy (2021). O SQLAlchemy também é responsável por abstrair a camada de conexão entre aplicações e banco de dados, fornecendo o gerenciamento de conexão automaticamente [SQLAlchemy 2021].

Flask-Login é um módulo que fornece gerenciamento de sessões de usuário para o Flask. Lida com tarefas comuns de *login*, *logout* e registro de sessão de usuário por longos períodos de tempo [Flask-Login 2021].

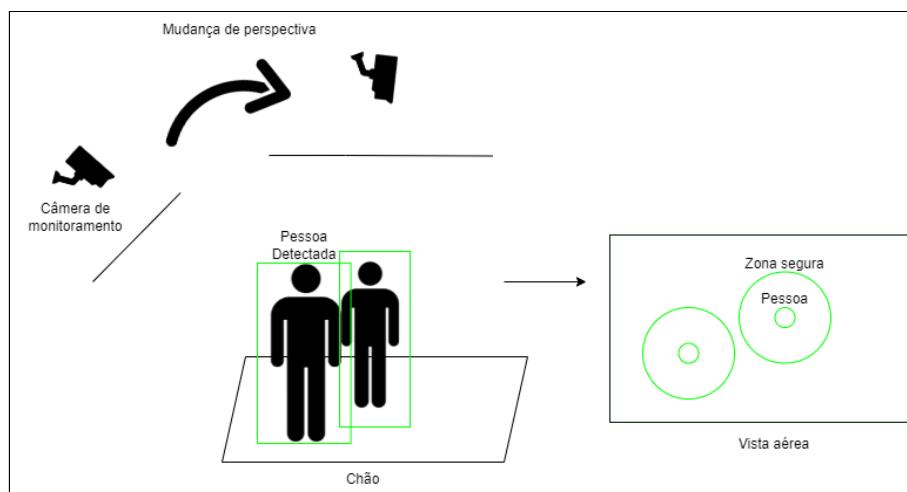
### 6.3 Arquitetura de Visão Computacional

Uma Rede Neural Convolucional ou, em inglês, *Convolutional Neural Network* (CNN) é um algoritmo de aprendizado profundo que pode capturar imagens de entrada, atribuir importância (pesos e vieses auto ajustáveis) a vários aspectos de objetos da imagem e ser capaz de distingui-los uns dos outros [Deep Learning Book, 2021].

Para o desenvolvimento da detecção de pessoas em vídeo, foi implementado o sistema de detecção de objetos em tempo real utilizando Tensorflow Lite, um conjunto de ferramentas de aprendizagem de máquina que auxilia a execução de modelos em dispositivos móveis e sistemas web [Tensorflow 2021].

O modelo escolhido para predição é um detector Faster RCNN (*Faster Region Convolutional Neural Network*) que permite localizar objetos em imagens, indicando sua localização e sua classificação através de redes neurais convolucionais. Para o projeto é utilizado um modelo previamente treinado disponibilizado pelo projeto AVA. O AVA é um conjunto de dados audiovisuais e anotações para construção de soluções cognitivas [AVA, 2021].

A aferição do distanciamento social é realizada através da distorção de perspectiva do chão, sendo assim é necessário que a câmera de monitoramento esteja fixada em um ponto alto, acima das pessoas e inclinada aproximadamente 45° voltada para o chão, garantindo que o sistema consiga criar a vista áerea, como mostra a Figura 4.



**Figura 4. Exemplificação da transformação de perspectiva.**

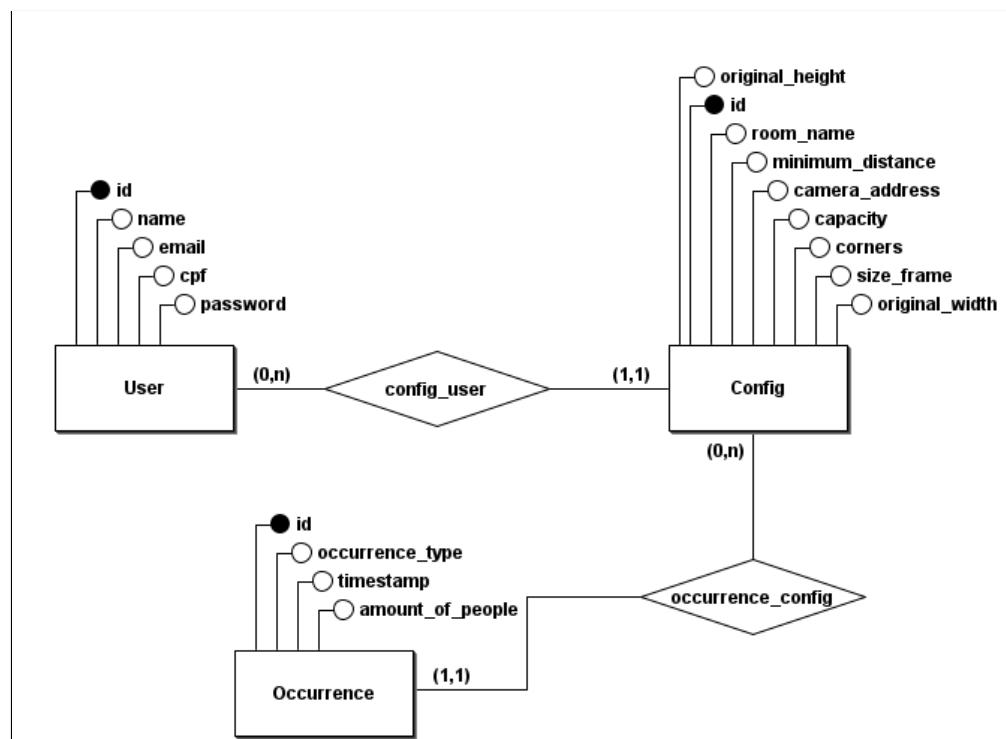
O exemplo da Figura 4 mostra o processo que ocorre na distorção de perspectiva do chão. Para realização desta é necessário ter as quatro coordenadas referentes aos quatro cantos da área em que se deseja aplicar a transformação. A partir dessa deformação é possível determinar uma “zona segura” ao entorno de cada pessoa detectada, essa zona segura se abstrai em um círculo onde o usuário pode definir seu tamanho em medida de *pixels*.

A calibração de perspectiva é um recurso fundamental para a assertividade do sistema. Assim, o sistema disponibilizará uma ferramenta gráfica na qual o usuário mapeará 4 pontos da imagem que definirão os quatro cantos referentes ao chão do cenário.

#### 6.4 Modelagem de dados

Quando se inicia um projeto de *software* é primordial estudar e levantar todos os requisitos e funcionalidades para a construção do produto final antes de colocar o desenvolvimento em prática. Nesse sentido, pode-se desenvolver um modelo conceitual utilizado para guiar o desenvolvimento propriamente dito.

O Modelo Entidade-Relacionamento (MER) é um modelo conceitual estabelecido na área de Engenharia de *Software* para descrever os objetos (entidades), seus atributos e como se relacionam entre si. Em outras palavras, o MER irá abstrair a estrutura que o banco de dados da aplicação final terá. A seguir, na Figura 5 é apresentado o esquema conceitual de banco de dados da aplicação.

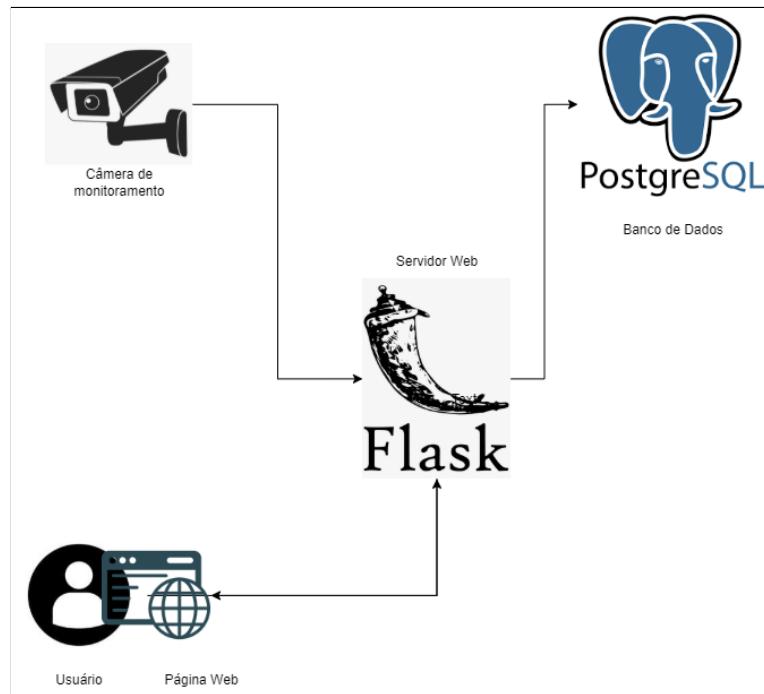


**Figura 5. Diagrama entidade relacionamento do projeto Critical Monitor.**

#### 6.4 Arquitetura de *software*

No desenvolvimento do Critical Monitor foi optado pela utilização do padrão de arquitetura de software MVC (*Model View Controller*), no qual separa-se o projeto em três camadas, sendo *Model* a camada que manipula a lógica dos dados, estado e parâmetros. Já a camada *View* recebe os dados, apresentando-os na interface de usuário, enquanto a camada *Controller* opera sobre os fluxos da aplicação, sendo a camada intermediária entre *Model* e *View*.

A Figura 6, demonstra a comunicação entre os componentes necessários para o funcionamento do software. No centro da arquitetura está o servidor *web* desenvolvido com o *framework* Flask, que recebe a transmissão de vídeo da câmera de monitoramento, controla os modelos, consulta e escreve dados no SGBD PostgreSQL e recebe requisições do cliente respondendo com páginas *web*.



**Figura 6. Arquitetura de solução do Critical Monitor.**

## 6.5 Interface do usuário

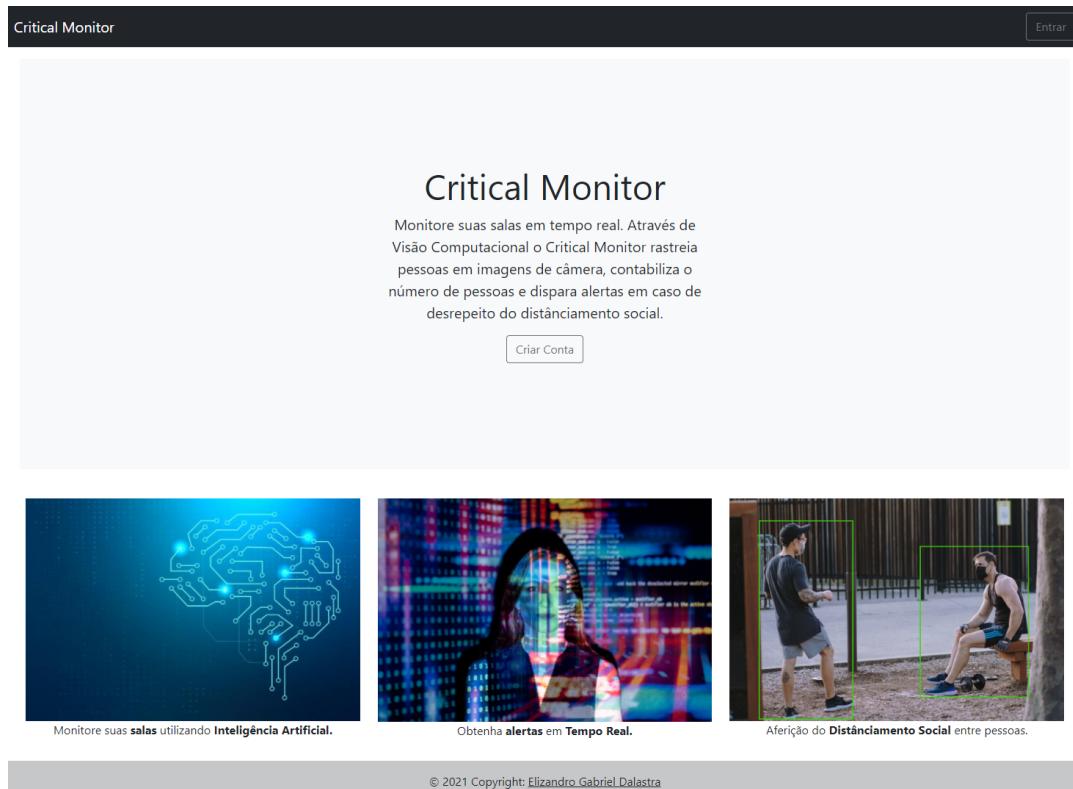
De forma geral, de nada adianta ter uma aplicação super estruturada e bem definida internamente se a parte visível do *software* não for bem elaborada. Uma interface de usuário sem padrões e com uma usabilidade não planejada acaba com a experiência de usuário e passa a impressão de que a aplicação é “ruim” como um todo, portanto, esse ponto é extremamente fundamental para o *software* [Pressman 2011].

Sendo assim, a interface do sistema foi construída utilizando a biblioteca *front-end* Bootstrap que disponibiliza estruturas CSS (*Cascading Style Sheets*), para criação de sites e demais aplicações responsivas [Bootstrap 2021]. Além disso, a Bootstrap fornece estilização a vários componentes de páginas como botões, formulários, imagens, tabelas, parágrafos, listas e diversos outros [Bootstrap 2021].

No Critical Monitor, a interface do usuário foi pensada para ser minimalista e prática, visto que é um sistema que necessita ter uma navegação fácil onde é possível alternar entre configurações, monitor e histórico de forma rápida e prática.

Começando pela Figura 7, é mostrada a página de entrada da aplicação, no centro da interface é possível observar um pequeno texto explicativo que traz, de forma

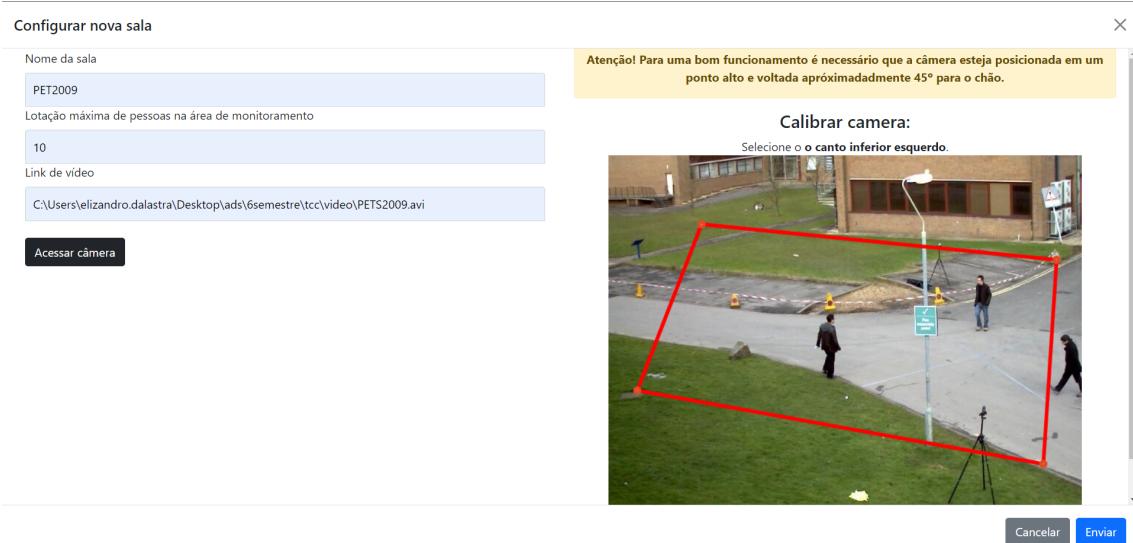
resumida, o que o Critical Monitor é capaz de realizar já trazendo, também, um botão que direciona para o cadastro de usuário no sistema. Logo abaixo, é possível observar 3 imagens de impacto com suas respectivas legendas. O objetivo destas imagens é chamar atenção do usuário sobre os pontos de mais valor do *software*.



**Figura 7. Interface de apresentação do sistema Critical Monitor.**

Para configuração do sistema é necessário uma página de fácil usabilidade por parte do usuário. Na Figura 8, destaca-se a interface de configuração de sala do sistema onde a página é dividida horizontalmente em duas partes, utilizando o recurso de *grid* disponibilizado pelo Bootstrap. Assim, torna-se possível a visualização total de todos os componentes sem a necessidade de usar o recurso de *scroll* do navegador.

No lado esquerdo é exibido o formulário com três campos de texto onde o usuário deve informar nome da sala, quantidade permitida de pessoas na sala e fonte do vídeo. Já no lado direito, exibe-se a imagem capturada pela fonte de vídeo do usuário, onde o mesmo deve clicar nos quatro pontos dos cantos inferiores da área de monitoramento desejado. Abaixo, no rodapé estão os botões “Cancelar” e “Enviar”, quando clicado em “Cancelar” o sistema redireciona para página de listagem de salas e quando selecionado o botão “Enviar” o sistema submete as informações para o servidor.

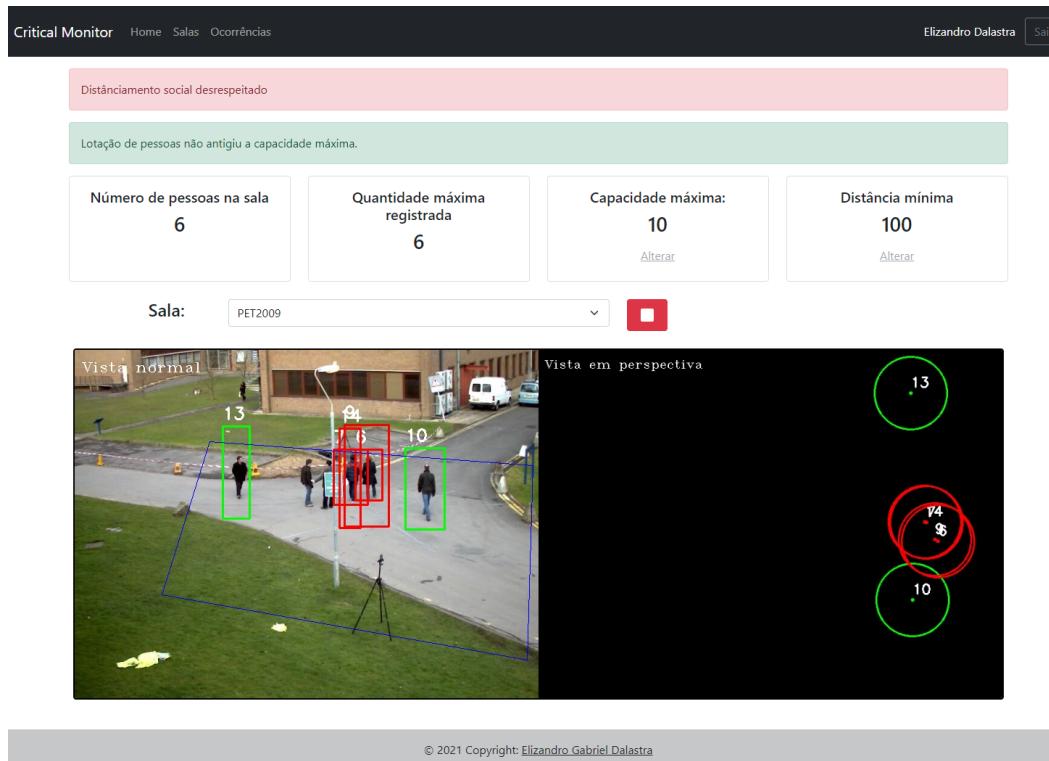


**Figura 8. Interface de configuração do Critical Monitor.**

Na Figura 9, evidencia-se a interface de monitoramento, assim sendo a interface principal do sistema. Na parte superior da página, encontra-se o cabeçalho de navegação, onde os acessos às demais páginas. Abaixo, destacam-se os alertas com cores que variam dinamicamente conforme o estado da sala. Após os alertas é possível notar 4 *cards* que informam o usuário sobre valores de interesse. E, por último, está a visualização do monitor em tempo real.

No monitor da esquerda é mostrado o vídeo em perspectiva normal, com retângulos desenhados ao entorno de cada pessoa detectada e um número identificador acima dela. Os retângulos podem variar entre as cores verde, no caso em que a pessoa não está desrespeitando o distanciamento social, e vermelho, se a pessoa desrespeitar o distanciamento mínimo estipulado.

O monitor da direita exibe uma correspondência do vídeo original, com a transformação de perspectiva que cria uma visualização aérea. Nesse caso, cada pessoa detectada é representada por um ponto e um círculo externo que simboliza a “zona segura”. Nesse formato a cor da representação de cada pessoa também varia conforme as condições citadas acima.



**Figura 9. Interface de monitoramento do projeto Critical Monitor.**

E, por fim, a Figura 10 mostra o histórico de ocorrências para cada dia. O *layout* é dividido horizontalmente em duas partes: na direita está a tabela contendo todas as ocorrências registradas naquele dia, enquanto na esquerda é exibido um gráfico no qual o eixo x representa as horas do dia, dispostas de 30 em 30 minutos, e o eixo y representa a quantidade de ocorrências registradas.



**Figura 10. Visualização do histórico de ocorrências do Critical Monitor.**

## 7. Trabalhos Futuros

Como trabalhos futuros, o Critical Monitor terá um modelo de aprendizado profundo próprio, o qual será treinado visando o problema em específico e, assim, melhorando a precisão e desempenho na detecção de pessoas.

Também, a aplicação deve ter é adotar algum recurso de disparos de alertas para outros sistemas, como o aplicativo de mensagens instantâneas Telegram através de um *chatbot* onde o usuário poderá consultar a situação de suas salas por mensagem.

Outra melhoria importante para o *software* é desacoplar a detecção de pessoas da aplicação, implementando-a em um dispositivo de borda como por exemplo um Raspberry PI, que poderá ter uma câmera diretamente conectada a ele. Essa mudança tornará a solução mais portátil, pois o processamento das imagens é executado localmente e também elimina o problema de latência causado pela comunicação entre câmera e servidor.

## 8. Considerações Finais

O desenvolvimento da aplicação Critical Monitor permitiu aperfeiçoamento e aprendizado de alguns conhecimentos, entendendo como as tecnologias integram-se entre si para, assim, construir uma solução única.

Além disso, é importante destacar que o trabalho trouxe uma abrangência no conhecimento na área de inteligência artificial no campo de aprendizado de máquina e visão computacional que, por sua vez, são tecnologias inovadoras e que vêm cada vez mais transformando os negócios digitais. Assim como o aprimoramento dos conhecimentos obtidos no curso.

Outro fator de extrema importância que foi abordado durante todo o processo de desenvolvimento do sistema é o desempenho. Algoritmos de predição de imagem tendem a consumir bastante carga de processamento, o que para uma aplicação que deve monitorar cenários críticos e disparar alertas é um empecilho. O sistema foi pensado numa arquitetura capaz de otimizar as inferências, sendo assim apto a responder de forma quase instantânea.

Por outro lado, o Critical Monitor se mostra como uma solução muito considerável para o cenário atual de pandemia. A Covid-19 foi responsável por muitas mudanças na sociedade, diversas empresas necessitam se adaptar às normas de contingência do vírus, portanto o sistema desenvolvido pode auxiliar expressivamente para mitigar o espalhamento da doença.

## 9. Referências

3RCorp. CÂMERA INTELIGENTE. Disponível em:  
<http://3rcorp.com.br/camera-inteligente>. Acesso em Junho de 2021.

OpenCV.About - OpenCV. Disponível em: <https://opencv.org/about/>. Acesso em Agosto de 2021.

Amazon Web Services (AWS). Amazon Rekognition. Disponível em:  
<https://aws.amazon.com/pt/rekognition/>. Acesso em Junho de 2021.

AVA. “Action Dataset”. Disponível em: <https://research.google.com/ava/index.html>. Acesso em novembro de 2021.

Bootstrap. Build fast, responsive sites with Bootstrap. Disponível em: <https://getbootstrap.com/>. Acesso em Agosto de 2021.

Deep Learning Book. “Introdução às Redes Neurais Convolucionais”. Disponível em: <https://www.deeplearningbook.com.br/introducao-as-redes-neurais-convolucionais/>. Acesso em novembro de 2021.

IBM. “Computer Vision”. Disponível em: <https://bitlyli.com/jP0es>. Acesso em Junho de 2021.

jQuery. What is jQuery?. Disponível em: <https://jquery.com/>. Acesso em Agosto de 2021.

Ministério da Saúde (MS). “Mesmo após vacinação, uso de máscara e distanciamento social precisam ser mantidos, alerta SES”. Disponível em: <https://www.saude.ms.gov.br/mesmo-apos-vacinacao-uso-de-mascara-e-distanciamento-social-precisam-ser-mantidos-alerta-ses/>. Acesso em dezembro de 2021.

Ministério da Saúde (MS). Boletim Epidemiológico Especial - 14. COE-COVID19 [https://portalarquivos.saude.gov.br/images/pdf/2020/April/27/2020-04-27-18-05h-BE\\_E-Boletim-do-COE.pdf](https://portalarquivos.saude.gov.br/images/pdf/2020/April/27/2020-04-27-18-05h-BE_E-Boletim-do-COE.pdf). Acesso em Julho de 2021.

Oracle. “O que é inteligência artificial – IA?”. Disponível em: <https://www.oracle.com/br/artificial-intelligence/what-is-ai/>. Acesso em janeiro de 2022.

Passos, R. Silveira, S. E Abrahão, J. (2021) “Exploratory assessment of the occurrence of SARS-CoV-2 in aerosols in hospital facilities and public spaces of a metropolitan center in Brazil”, Em: Environmental Research. Disponível em: <https://pubmed.ncbi.nlm.nih.gov/33513382/>. Acesso em: Julho de 2021

PostgreSQL: about. PostgreSQL. Disponível em <https://www.postgresql.org/about/>. Acesso em Agosto de 2021.

Pressman, R. S. (2011). “Engenharia de Software: Uma Abordagem Profissional”, McGraw-Hill, 7a edição.

Pytest, “pytest: helps you write better programs”, Disponível em: <https://docs.pytest.org/en/6.2.x/>. Acesso em Deszembro de 2021.

Read the Docs, “Flask-Login”, Disponível em: <https://flask-login.readthedocs.io/en/latest/>, Acesso em: Dezembro de 2021.

Sommerville, I (2011). “Engenharia de Software”, Pearson Prentice Hall, 9a edição.

SQLAlchemy, “The Python SQL Toolkit and Object Relational Mapper”. Disponível em: <https://www.sqlalchemy.org/>. Acesso em Novembro de 2021.

TensorFlow. Porque usar TensorFlow,. Disponível em:  
<https://palletsprojects.com/p/flask/>. Acesso em Agosto de 2021.

The Pallets Projects. Flask. Disponível em: <https://palletsprojects.com/p/flask/>. Acesso em Agosto de 2021.

The Pallets Projects. Jinja. Disponível em: <https://jinja.palletsprojects.com/en/3.0.x/>. Acesso em Novembro de 2021.

World Health Organization (WHO). Coronavirus disease 2019 (COVID-19). Situation report

-51. <https://bitlyli.com/EubLw>. Acesso em julho de 2021.