

# Sistema de Otimização de consultas Laboratoriais via Reconhecimento Facial.

Alexandre Freire Roperio Júnior, Pedro Trindade Francisco, Guilherme dos Reis Freitas, Marcel Stefan Wagner

Departamento de Engenharia da Computação

Universidade Anhembi Morumbi (UAM)

**Resumo** — A pandemia de COVID-19, que começou em 2020, revelou a vulnerabilidade do mundo diante de crises de saúde e a importância de repensar a maneira como lidamos com questões médicas. Hospitais e laboratórios, tradicionalmente lotados e propícios à formação de filas, representavam um risco significativo durante a pandemia, aumentando a exposição e a disseminação do vírus. Como resposta a essa necessidade premente, este projeto visa utilizar a tecnologia para otimizar o atendimento em laboratórios.

O propósito central deste projeto é implementar um sistema inovador que emprega um totem equipado com uma câmera que realiza o reconhecimento facial dos pacientes. Esse processo permite a coleta de dados dos agendamentos do paciente, que são, então, encaminhados para um sistema de gerenciamento de agendamentos baseado em uma aplicação web. Através dessa aplicação, os pacientes podem validar suas informações e agendar consultas de forma ágil e eficiente, sem a necessidade de enfrentar longas filas para realização de exames.

**Palavras-Chave** — Tecnologia. Saúde. Covid-19. Reconhecimento Facial. Agendar Consultas. Totem.

## I. INTRODUÇÃO

A partir de 26 de fevereiro de 2020, quando foi confirmado o primeiro caso de Covid-19 no Brasil, foi possível perceber a falta de estrutura em ambientes que envolvem a área da saúde, principalmente em hospitais e laboratórios, locais onde, para fazer um teste de Coronavírus, os pacientes precisavam esperar horas em uma fila.

Essa situação destacou uma carência significativa em nossa infraestrutura de saúde, e a pandemia global de COVID-19 revelou as deficiências existentes nos sistemas de atendimento médico, principalmente em hospitais e laboratórios. Em muitos casos, para realizar um simples teste de COVID-19, os pacientes eram submetidos a longas e demoradas filas de espera, aumentando consideravelmente o risco de exposição ao vírus [1].

Pensando na necessidade de melhorar o tempo que as pessoas passam em filas de laboratórios, surgiu a oportunidade de criar um projeto baseado em tecnologia, com o propósito de otimizar as consultas médicas em laboratórios. No âmbito deste projeto, a utilização do reconhecimento facial desempenha um papel fundamental.

O objetivo principal deste projeto é implementar um sistema no qual a presença do paciente no local da consulta já marcada seja confirmada por meio do reconhecimento facial. Esta

abordagem visa a redução significativa do tempo de espera dos pacientes e a minimização do risco de exposição à COVID-19.

De acordo com uma pesquisa de mercado realizada pela Juniper Research: Digital Technology Market Research Services, o mercado de reconhecimento facial está em constante crescimento, com uma previsão de aumento de 120% até o ano de 2025 [2]. O reconhecimento facial está se tornando cada vez mais acessível e é amplamente adotado em novos projetos, incluindo soluções de desbloqueio de smartphones e tecnologias inovadoras, como o "selfie pay" [3], que permite efetuar pagamentos por meio do reconhecimento facial.

O projeto proposto visa coletar informações do paciente no momento em que ele entra no estabelecimento de saúde, eliminando a necessidade de aguardar em filas para atendimento inicial. Com o uso do reconhecimento facial, o processo de check-in torna-se automatizado e mais eficiente. O sistema é composto por duas aplicações, ambas com acesso ao mesmo banco de dados, permitindo o fluxo contínuo de informações entre elas.

Ambas as aplicações são web (Uma aplicação web é um sistema que opera na internet, dispensando a necessidade de instalação local), onde uma das aplicações foi feita em Python® e simula um totem, onde o paciente efetua o check-in através do reconhecimento facial, e as informações dos agendamentos do paciente são coletados para confirmar sua presença. A segunda aplicação, feita em Java®, realiza os cadastros gerais e acompanha os status dos agendamentos dos pacientes. Após a aprovação dos agendamentos, os pacientes serão encaminhados à ala de exames do laboratório, garantindo um processo eficiente e simplificado.

## II. PROBLEMA DE PESQUISA

Sabe-se que passar muito tempo esperando para ser atendido pode tornar a experiência do paciente em idas a laboratórios estressante e tediosa [4]. Isso é particularmente relevante em um contexto de pandemia, onde minimizar o tempo de exposição em ambientes de saúde é fundamental. Portanto, o principal problema a ser resolvido através do desenvolvimento da pesquisa é otimizar o tempo que o paciente passa para ser atendido, reduzindo o tempo de espera e, assim, a exposição ao risco de contágio pela COVID-19 e outras doenças.

Alguns dos fatores que contribuem para essa lentidão são os computadores utilizados pelos atendentes, que geralmente não são muito rápidos e não permitem que executem as tarefas com excelência. Além disso, a falta de profissionais para agilizar o

atendimento também é um fator responsável pela demora, ocasionando o mesmo problema. É importante destacar que, além de causar desconforto ao paciente, a demora no atendimento pode prejudicar a eficiência e a qualidade dos serviços de saúde.

Com a adição de uma inteligência artificial para fazer o primeiro atendimento, através do reconhecimento facial do paciente, toda a primeira etapa do atendimento seria substituída por um modelo automático, que pegaria os dados do paciente e entregaria a confirmação do convênio. Isso reduziria o tempo da etapa inicial de atendimento, tornando este processo mais eficaz e seguro, especialmente em tempos de pandemia.

### III. OBJETIVOS

#### A. OBJETIVOS GERAIS

A experiência do paciente ao visitar laboratórios e hospitais é frequentemente marcada pela ansiedade e desconforto, uma vez que os pacientes muitas vezes estão doentes e em dor. Nesse contexto, a gestão eficiente das filas desempenha um papel crucial na melhoria dessa experiência, tornando-a mais tranquila e menos estressante. Portanto, o objetivo geral deste projeto é desenvolver e implementar um sistema de reconhecimento facial com o propósito de otimizar a etapa inicial de atendimento, proporcionando uma experiência de atendimento mais eficaz e humanizada.

#### B. OBJETIVOS ESPECÍFICOS

Nesta sessão são apresentados objetivos específicos que permeiam os trabalhos a serem realizados durante o desenvolvimento da pesquisa:

- **Otimizar** o Tempo de Espera do Atendimento: O principal objetivo é reduzir o tempo que os pacientes passam aguardando atendimento, tornando o processo de inicial de atendimento mais rápido e eficiente por meio do reconhecimento facial.
- **Facilitar** a Etapa Inicial de Atendimento do Paciente: Facilitar a etapa inicial de atendimento dos pacientes é essencial para uma recepção eficaz e a alocação de recursos adequados para atender às necessidades de cada paciente.
- **Agilizar** consultas e exames de rotina: A agilização do processo de agendamento e atendimento de consultas e exames de rotina é fundamental para otimizar o fluxo de pacientes e proporcionar uma experiência mais eficaz e conveniente.

### IV. PESQUISA TEÓRICA

#### A. Padrão de Arquitetura MVC (Model-View-Controller)

O padrão de arquitetura Modelo-Visão-Controlador (MVC) é amplamente utilizado em projetos de desenvolvimento de software devido à sua estrutura organizada e à facilidade de manutenção do código. O MVC divide o aplicativo em três camadas distintas: *Model* (Modelo), que lida com a manipulação de dados e validações; *View* (Visão), responsável por exibir informações na tela para o usuário; e *Controller* (Controlador), que conecta o Modelo à Visão. A escolha de implementar o MVC neste projeto visa proporcionar uma organização mais eficiente e simplificada, permitindo uma colaboração eficaz entre os membros da equipe de desenvolvimento e, por consequência, a conclusão mais rápida do projeto [5].

Em uma aplicação real, o modelo seria uma entidade, a view seria uma página HTML chamada, e o controller seria uma classe que mapeia todos os *endpoints* e decide qual página será chamada em cada um deles. Ambas as aplicações, a principal e a que simula o totem de autoatendimento [6], estão utilizando esse padrão de arquitetura.

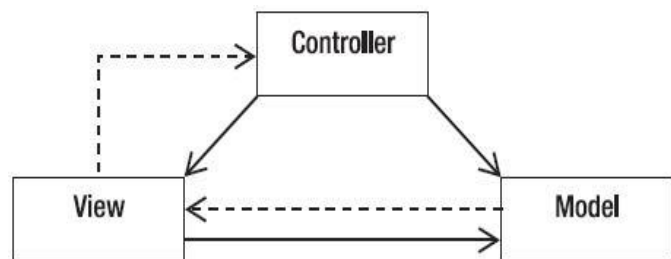


Figura 1 – Padrão de projeto MVC.

Fonte: DevMedia, (2022).

#### B. Spring Framework

O Spring Framework desempenha um papel crucial no desenvolvimento deste projeto, contribuindo para a eficiência e simplificação do código. O Spring é um framework abrangente para desenvolvimento de aplicativos Java, oferecendo uma variedade de ferramentas e bibliotecas que otimizam o processo de desenvolvimento de software.

- **Spring Security:** Esta é uma dependência vital que cuida de toda a configuração de segurança da aplicação. O Spring Security é responsável por proteger a aplicação contra ameaças e garantir que apenas usuários autorizados tenham acesso aos recursos protegidos. Ele oferece uma ampla gama de recursos, desde autenticação básica até controle de acesso avançado.

- **Spring Validation:** No contexto deste projeto, a validação de dados é essencial para garantir a integridade dos dados antes que eles alcancem o banco de dados. A dependência Spring Validation é responsável por validar os dados fornecidos por meio dos DTOs (Objetos de Transferência de Dados), assegurando que estejam em conformidade com os requisitos definidos antes de serem persistidos no banco de dados.
- **Spring Web:** Esta dependência é responsável pela parte do servidor da aplicação. O Spring Web, integrado ao Tomcat configurado, lida com a manipulação de solicitações HTTP, permitindo que a aplicação responda de maneira eficiente a diversas interações do usuário.
- **Spring Data:** Essa dependência foi mencionada anteriormente e desempenha um papel fundamental na integração do aplicativo com o banco de dados. O Spring Data simplifica a interação com o banco de dados, fornecendo uma abstração de alto nível para operações comuns, facilitando a implementação de operações de leitura e gravação no banco de dados.
- **Thymeleaf:** Esta é uma *engine* de modelo Java para servidores web, integrada ao Spring Framework. O Thymeleaf permite a criação de páginas web dinâmicas e interativas, facilitando a incorporação de dados nas visualizações da aplicação. Sua integração suave com o Spring simplifica o desenvolvimento de interfaces de usuário eficazes e responsivas.

A utilização dessas dependências do Spring no projeto principal contribui para a robustez, segurança e eficiência global da aplicação, proporcionando uma base sólida para o desenvolvimento e manutenção contínua através de bibliotecas conforme ilustrado na figura 2 [7].

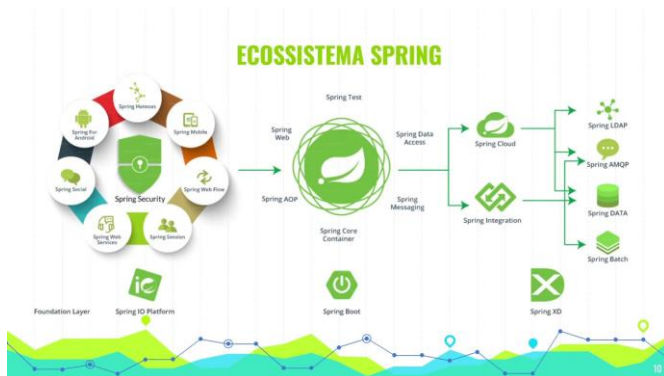


Figura 2 (Traduzida por SETIC UFC) – Spring Tutorial – A Java Framework Providing Efficiency.

Fonte: Edureka, (2023).

### C. Java

A linguagem de programação escolhida para este projeto é o Java. A escolha se baseia no conhecimento prévio da linguagem pela equipe e na sua base de Programação Orientada a Objetos (POO).

Em outras linguagens como C e C++, o código é compilado e transformado em linguagem de máquina sendo executado diretamente pelo sistema. Um problema é que cada sistema operacional tem uma linguagem de máquina específica, por isso não é possível rodar um programa feito no Windows em um Linux.

Porém, com o Java ocorre de uma maneira diferente, primeiramente código é compilado para bytecodes, transformando o arquivo original .java em um arquivo .class, que é interpretado e executado pela JVM (Java Virtual Machine), conforme ilustrado na Figura 3, por isso se diz que Java é uma linguagem interpretada. Com isso, o código pode ser executado em qualquer computador independente do sistema operacional, contanto que possua a máquina virtual Java.

### D. Python

Python foi a linguagem de programação escolhida para a aplicação do totem. A sintaxe simples e a sua versatilidade foram os motivos para a escolha. Além disso, o Python serve como base em muitos projetos de inteligência artificial e apresenta uma grande facilidade de aprendizado.

### E. Flask

Flask é o framework usado na aplicação em Python. Flask é conhecido por ser um framework web leve, simplificado e fácil de usar. Ele fornece as ferramentas necessárias para construir rapidamente aplicações web eficientes em Python. Com sua abordagem "micro", o Flask não impõe uma estrutura rígida, permitindo flexibilidade no desenvolvimento. Sua integração com Python e a comunidade ativa tornam o Flask uma escolha popular para projetos web.

A combinação de Python e Flask proporciona uma experiência de desenvolvimento ágil, permitindo a criação eficiente da aplicação do totem, destacando-se pela simplicidade da linguagem e pela estrutura flexível do Flask [9].

### F. MySQL

Para o banco de dados do projeto, a escolha recaiu sobre o MySQL devido à familiaridade da equipe com a ferramenta e à sua ampla utilização no mercado. Conforme indicado na Tabela 1, o MySQL é um dos sistemas de gerenciamento de banco de dados mais utilizados, ficando atrás apenas do Oracle Database. Essa escolha visa garantir a eficiência e a confiabilidade do banco de dados.

O banco de dados é usado por ambas as aplicações.

TABELA 1

Classificação			SGBD	Tipos de banco de Dados
Abril	Março	Abril		
2022	2022	2021		
1.	1.	1.	Oracle	Relacional
2.	2.	2.	MySQL	Relacional
3.	3.	3.	Microsoft SQL Server	Relacional
4.	4.	4.	PostgreSQL	Relacional
5.	5.	5.	MongoDB	Orientado a documento
6.	6.	↑7.	Redis	Não é SQL
7.	↑8.	↑8.	Elasticsearch	Não é SQL
8.	↓7.	↓6.	IBM Db2	Relacional
9.	9.	↑10.	Microsoft Access	Relacional
10.	10.	↓9.	SQLite	Relacional

Fonte: Austrian IT Consulting. (06/04/2022).

## V. METODOLOGIA

A metodologia científica do projeto pode ser classificada da seguinte forma:

- Quanto à natureza: A pesquisa é considerada aplicada, pois é direcionada para hospitais e laboratórios e visa solucionar problemas encontrados nesses locais. A abordagem aplicada é fundamental para garantir que o projeto atenda às necessidades reais dessas instituições de saúde e que a solução proposta seja prática e eficaz.
- Quanto aos objetivos: A pesquisa é considerada exploratória. A natureza exploratória do projeto envolve o desenvolvimento de uma aplicação inovadora que utiliza
- tecnologias de reconhecimento facial para otimizar o atendimento em laboratórios. Embora o projeto tenha objetivos claros, a exploração está relacionada à implementação de uma solução pioneira que ainda não foi amplamente adotada nesse contexto.
- Quanto aos procedimentos técnicos da pesquisa: A pesquisa é considerada bibliográfica e prática.

- Pesquisa Bibliográfica: A pesquisa bibliográfica desempenha um papel fundamental na compreensão das melhores práticas, conceitos teóricos e tendências em tecnologia de reconhecimento facial, arquitetura de software (como o padrão MVC) e outras áreas relevantes. A maioria dos materiais utilizados para a elaboração do projeto já estão concluídos e pode ser encontrada em fontes bibliográficas, como livros, artigos científicos, sites da internet e

repositórios como o GitHub. Essa pesquisa bibliográfica fornecerá uma base sólida para o desenvolvimento da aplicação.

- Pesquisa Prática: A pesquisa prática envolve a implementação real do sistema de reconhecimento facial e das aplicações associadas. Isso inclui a criação de código, a integração de tecnologias (como o Spring Framework) e a configuração do banco de dados MySQL. A pesquisa prática permitirá testar e aprimorar a solução em um ambiente real, garantindo sua eficácia e usabilidade.

## VI. DESENVOLVIMENTO

O projeto é composto por duas aplicações distintas: uma aplicação *desktop* simulando um totem iterativo para a seleção de exames por parte dos pacientes, utilizando o reconhecimento facial para a coleta de informações; a segunda aplicação é baseada na web e exibe uma fila de espera, mostrando a ordem de chegada dos pacientes e fornecendo uma página para validação do convênio. A Figura 3 ilustra o funcionamento geral do projeto.

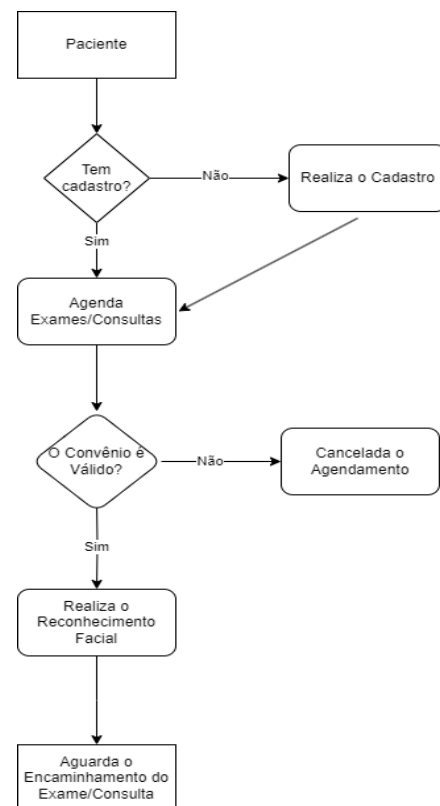


Figura 3 – Fluxograma do funcionamento do projeto

Fonte: Próprio Autor, 2023.

O desenvolvimento do projeto foi conduzido com a utilização de metodologias ágeis, incluindo Kanban e Scrum, desempenhando um papel fundamental no processo de desenvolvimento.

**Kanban:** A metodologia Kanban é baseada em um sistema visual de quadros que auxiliam na gestão e acompanhamento do fluxo de trabalho [11]. Foi empregada para organizar as etapas do projeto e alocar tarefas em colunas distintas, como Backlog. Em progresso, Aguardando Validação e Concluído. Essa abordagem proporcionou maior transparência, permitindo que a equipe acompanhasse o progresso das atividades e identificasse possíveis gargalos, otimizando o fluxo de trabalho.

**Scrum:** Além do Kanban, o Scrum também foi utilizado como uma metodologia ágil adicional, com foco no gerenciamento de sprints, ciclos de desenvolvimento com duração pré-definida [12]. Durante as reuniões diárias, a equipe discutia o andamento do projeto, compartilhava informações sobre as atividades em andamento e identificava quaisquer impedimentos que pudessem afetar o progresso.

A integração das metodologias ágeis contribuiu para um desenvolvimento mais eficiente e adaptável, permitindo que o projeto avançasse de forma iterativa, priorizando entregas incrementais e constantes melhorias no sistema. Isso resultou em um desenvolvimento de forma ágil das aplicações e uma boa capacidade de resposta às mudanças e demandas do projeto.

Além das metodologias ágeis, o desenvolvimento do projeto também incorporou padrões de projeto, como os princípios SOLID e as práticas de Clean Code. Essas abordagens ajudaram a garantir a qualidade do código, tornando-o mais modular, sustentável e de fácil manutenção.

## 1. Princípios SOLID:

**Princípio da Responsabilidade Única (Single Responsibility Principle - SRP):** Cada classe ou módulo deve ter apenas uma razão para mudar, promovendo coesão e garantindo que cada componente seja responsável por uma única tarefa ou funcionalidade.

**Princípio do Aberto/Fechado (Open/Closed Principle - OCP):** As entidades do sistema devem estar abertas para extensão, mas fechadas para modificação, permitindo adicionar novas funcionalidades sem modificar o código existente.

**Princípio da Substituição de Liskov (Liskov Substitution Principle - LSP):** As classes derivadas devem ser substituíveis por suas classes base, sem alterar o comportamento esperado do programa.

**Princípio da Segregação de Interfaces (Interface Segregation Principle - ISP):** Uma classe não deve ser forçada a implementar interfaces que não utiliza. É preferível ter interfaces mais específicas em vez de uma única interface monolítica [13].

**Princípio da Inversão de Dependência (Dependency Inversion Principle - DIP):** Módulos de alto nível não devem depender de módulos de baixo nível; ambos devem depender de abstrações. A dependência deve ser invertida, permitindo maior flexibilidade e facilidade de troca de implementações.

## 2. Clean Code:

Clean Code é um conjunto de boas práticas que visa produzir código claro, legível e fácil de entender [14]. Algumas práticas adotadas no projeto incluem:

- Nomenclatura significativa para classes, métodos e variáveis, tornando o código autoexplicativo.
- Funções curtas e com apenas uma responsabilidade.
- Comentários pertinentes e explicativos quando necessário, para facilitar a compreensão do código.
- Redução de duplicações de código por meio de abstrações e reutilização.
- Organização lógica e estruturada do código, seguindo padrões de formatação.

A aplicação dos princípios SOLID e das práticas de Clean Code assegurou que o código do projeto fosse de alta qualidade, facilitando a manutenção, extensibilidade e escalabilidade do sistema. Essas abordagens contribuíram significativamente para a robustez e eficiência do projeto como um todo.

A proteção dos dados sensíveis dos pacientes é uma prioridade crucial no desenvolvimento do projeto. Para assegurar a confidencialidade e integridade desses dados, a implementação da criptografia desempenha um papel fundamental.

A criptografia é realizada por meio do algoritmo AES (Advanced Encryption Standard) com o modo de operação CBC (Cipher Block Chaining) e o preenchimento PKCS5Padding. A chave de criptografia, gerada a partir da senha fornecida, é protegida e armazenada de maneira segura. O Vetor de Inicialização (IV) é estático para simplificar a implementação.

A decisão de usar o algoritmo AES com o modo de operação CBC é motivada pela sua robustez comprovada na proteção de dados sensíveis. A escolha do preenchimento PKCS5Padding contribui para garantir a consistência do tamanho dos blocos criptografados.

Abaixo está o código da criptografia, destacando a implementação detalhada desses procedimentos de segurança:

```
@Component
public class EncryptionStringUtil {

    static String IV = "AAAAAAAAAAAAAAAA";

    public static String encrypt(String data, String encryptionKey) throws
    Exception {

        Cipher encrypt = Cipher.getInstance("AES/CBC/PKCS5Padding", "SunJCE");

        SecretKeySpec keySpec = new
        SecretKeySpec(encryptionKey.getBytes(StandardCharsets.UTF_8), "AES");

        encrypt.init(Cipher.ENCRYPT_MODE, keySpec, new
        IvParameterSpec(IV.getBytes(StandardCharsets.UTF_8)));
```

```

byte[] textoEncryptado =
encrypt.doFinal(data.getBytes(StandardCharsets.UTF_8));

return Base64.getEncoder().encodeToString(textoEncryptado);
}

public static String decrypt(String encryptedText,String encryptionKey)
throws Exception {

    Cipher decrypt=Cipher.getInstance("AES/CBC/PKCS5Padding","SunJCE");

    SecretKeySpec keySpec = new
    SecretKeySpec(encryptionKey.getBytes(StandardCharsets.UTF_8),"AES");

    decrypt.init(Cipher.DECRYPT_MODE,keySpec,new
    IvParameterSpec(IV.getBytes(StandardCharsets.UTF_8)));

    byte[] textoEncryptado = Base64.getDecoder().decode(encryptedText);

    return new
    String(decrypt.doFinal(textoEncryptado),StandardCharsets.UTF_8);
}
}

```

A Figura 4 abaixo demonstra como ficaria os dados de um paciente cadastrado no banco de dados.

id	nome	rg
huTMNbtA8Wrad6F3M5/Ug==	13Ue6rvll53emWzS4gIKWw==	
NULL	NULL	NULL

Figura 4 – Dados do Paciente criptografados  
Fonte: Próprio Autor, 2023.

## A. APLICAÇÃO PRINCIPAL

Abaixo está apresentada a figura 5, que demonstra como ficou a estrutura de pastas da aplicação web.

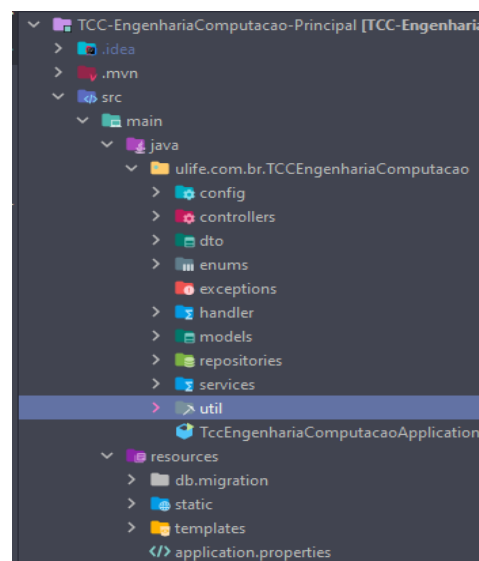


Figura 5 – Estrutura de pastas  
Fonte: Próprio Autor, 2023.

A figura 6 abaixo apresenta o modelo entidade Relacionamento da primeira aplicação e podemos analisar como o sistema irá funcionar.

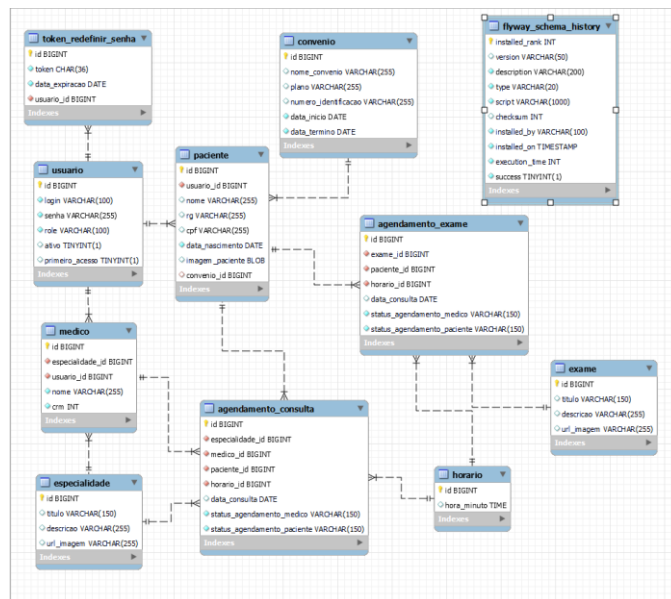


Figura 6 – Modelo Entidade Relacionamento  
Fonte: Próprio Autor, 2023.

O sistema permite o cadastro de usuários com diferentes funções (roles), tais como ADMIN, MÉDICO e PACIENTE.

Os usuários do tipo administrador (ADMIN) não tem dados pessoais inseridos no banco de dados, sendo armazenadas apenas as informações essenciais na tabela "usuarios": id, ativo, login e senha. O atributo "login" é o email do usuário para fins de autenticação. O campo "ativo" é utilizado para indicar se o cadastro do usuário está ativado (1) ou não (0).



Os dados pessoais de médicos e pacientes são armazenados em tabelas separadas, ou seja, "medicos" e "pacientes". Ambas possuem relacionamento de um para um (um usuário pode ser um médico ou um paciente) com a tabela "usuarios", utilizando a chave estrangeira "id" da tabela "usuário".

Os pacientes podem agendar consultas médicas selecionando a especialidade médica, o médico, a data e a hora da consulta. Essas informações serão armazenadas na tabela "agendamentos". Os médicos também podem visualizar as consultas agendadas para eles, acessando a mesma tabela. Dessa forma, as tabelas "pacientes" e "medicos" possuem relacionamento de um para muitos com a tabela "agendamentos".

Outras duas tabelas, "especialidades" e "horarios", também possuem relacionamentos de um para muitos com a tabela "agendamentos".

Importante ressaltar que um agendamento é único, ou seja, não pode existir mais de um agendamento com a mesma combinação de "dataConsulta", "especialidade\_id", "horario\_id" e "medico\_id". Essa restrição pode ser implementada por meio de um índice composto do tipo unique no banco de dados, porém, também será controlada a nível de aplicação.

## B. APLICAÇÃO DO TOTEM

A aplicação do totem permite que o paciente confirme sua presença no laboratório, para que possa realizar sua consulta ou exame previamente agendados através da aplicação principal, ou pegar uma senha caso não tenha sido cadastrado no sistema anteriormente.

O paciente que deseja confirmar sua presença em seus agendamentos, precisa informar o seu CPF, para que ocorra o processo de reconhecimento facial, filtrando a imagem que é utilizada para treinamento através do CPF. O CPF é utilizado para identificar se o paciente é existente e realizar uma consulta no banco de dados através do campo "cpf", trazendo assim dois campos, o "imagem\_paciente" e o "id" da tabela paciente, ambos atrelados ao mesmo CPF informado, com o campo "id" é realizado mais uma busca no banco de dados com intenção de trazer os exames e consultas marcadas deste paciente em questão, e o campo "imagem\_paciente" é utilizado para realizar o treinamento da IA (Inteligência Artificial) implementada através da biblioteca "face\_recognition", que é a biblioteca responsável pelo reconhecimento facial, capturando os principais pontos do rosto da pessoa, e realiza a comparação com a foto retirada do paciente no Totem. Caso a comparação das fotos confirme a autenticidade do paciente, é feita uma alteração de status do agendamento no banco de dados de "AGUARDANDO\_CONFIRMACAO\_AGENDAMENTO" para "PACIENTE\_PRESENTE", e refletindo na visualização dos agendamentos na aplicação principal.

A implementação da aplicação do totem trouxe algumas mudanças em relação ao planejamento inicial. Inicialmente, a intenção era utilizar a linguagem Java em todo o projeto. No entanto, após uma análise mais profunda sobre as necessidades de reconhecimento facial e inteligência artificial, surgiu a possibilidade de optar por Python, visto que essa escolha se

mostrou mais adequada às demandas do projeto e ofereceu vantagens significativas.

### Mudança de Java para Python:

A mudança de Java para Python se baseou em algumas considerações essenciais:

- **Facilidade na Implementação:** Python é conhecido por ser uma linguagem mais acessível para projetos que envolvem aprendizado de máquina e reconhecimento facial. A existência de bibliotecas bem estabelecidas, como OpenCV e dlib, contribui para a simplificação da implementação de recursos de reconhecimento facial.
- **Comunidade e Recursos:** Python tem uma vasta comunidade de desenvolvedores e é amplamente utilizado em projetos de IA e aprendizado de máquina. Isso se traduz em uma abundância de recursos, documentação e suporte disponíveis.
- **Eficiência na Análise de Dados:** Python também é uma escolha popular para a análise de dados, o que pode ser relevante ao trabalhar com informações de pacientes e convênios em um ambiente médico.

### Escolha entre Aplicação Web e Desktop:

A decisão de escolher entre uma aplicação *desktop* e uma aplicação web para a simulação do totem também passou por uma revisão. A introdução do projeto originalmente descreveu a aplicação do totem como sendo *desktop*. No entanto, após uma análise mais aprofundada e a consideração das vantagens e desvantagens de ambas as abordagens, decidimos que uma aplicação web seria a melhor escolha para a implementação do totem.

Essa mudança foi baseada nas seguintes razões:

- **Facilidade de Atualização:** Uma aplicação web permite atualizações centralizadas de conteúdo e funcionalidades, uma vez que os totens estão conectados à internet.
- **Maior Alcance:** Totens web podem ser acessados por qualquer dispositivo com um navegador, ampliando o alcance do público.
- **Flexibilidade:** Uma aplicação web oferece maior flexibilidade de acesso, tornando o sistema mais acessível para os usuários.

### Desvantagens desta mudança:

- **Dependência de Conexão:** Uma limitação inerente às aplicações web é a sua dependência contínua de uma conexão com a internet. Dessa forma, a ausência ou instabilidade na conexão pode impactar

significativamente o acesso e o desempenho da aplicação.

- Operação mais lenta: Ao contrário das aplicações locais, as aplicações web operam em um servidor, o qual se comunica com o seu navegador, resultando em um desempenho mais lento quando comparado a uma aplicação local.

## VII. RESULTADOS

Esta seção explora em detalhes os resultados obtidos durante os testes na aplicação do totem, com foco especial no reconhecimento facial. Os testes abordaram diferentes cenários, revelando diversos status e comportamentos da aplicação.

### A. Reconhecimento Facial:

Status e Comportamentos Específicos:

#### 1. Paciente Encontrado, mas Não Reconhecido:

Ao inserir um CPF correspondente a um paciente cadastrado, porém o sistema não conseguir reconhecê-lo pela imagem facial, a aplicação exibe o status "Paciente encontrado, mas não reconhecido" (Figura 7). Esta funcionalidade permite uma identificação prévia, mesmo quando o reconhecimento facial enfrenta desafios, garantindo que a informação do paciente seja acessada.



Figura 7 – Paciente encontrado, mas não reconhecido  
Fonte: Próprio Autor, 2023.

#### 2. Paciente Não Encontrado:

Quando um CPF não registrado no banco de dados é inserido, a aplicação exibe o status "Paciente não encontrado" (Figura 8). Essa resposta é essencial para informar ao usuário que o CPF inserido não está associado a nenhum paciente na base de dados, prevenindo possíveis erros.



Figura 8 – Paciente não encontrado  
Fonte: Próprio Autor, 2023.

#### 3. Paciente Sem Agendamento Atual:

Se um paciente é identificado no banco de dados, mas não possui agendamento para o dia atual, a aplicação apresenta uma mensagem indicando que o paciente não possui agendamento (Figura 9). Esse recurso assegura que apenas pacientes com agendamentos válidos possam utilizar a aplicação no momento.



Figura 9 – Paciente não possui agendamento  
Fonte: Próprio Autor, 2023.

### B. Testes de Iluminação:

Os testes realizados envolvendo variações na iluminação, tanto com luz ligada quanto apagada, demonstraram a robustez do sistema de reconhecimento facial. Nas Figuras 10 e 11, pode-se observar que a aplicação foi capaz de reconhecer os pacientes em ambos os cenários, garantindo a eficácia do reconhecimento facial em condições diversas.

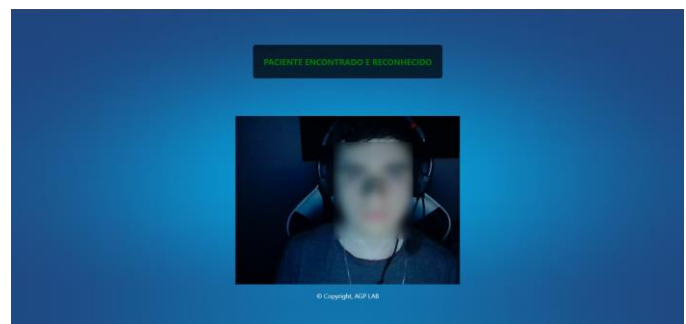


Figura 10 – Reconhecimento com luz apagada  
Fonte: Próprio Autor, 2023.





Figura 11 – Reconhecimento com luz ligada  
Fonte: Próprio Autor, 2023.

### C. Integração com a Aplicação Principal:

A Figura 12 apresenta a tela de login da aplicação principal, destacando a interconexão entre a aplicação do totem e o sistema principal. A integração bem-sucedida fortalece a eficiência global do projeto, proporcionando uma experiência unificada aos usuários.

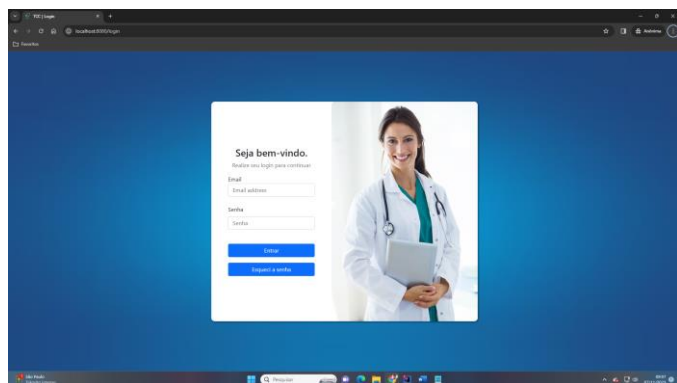


Figura 12 – Tela de Login da aplicação principal  
Fonte: Próprio Autor, 2023.

Esses resultados detalhados fornecem uma compreensão mais profunda do desempenho da aplicação do totem, destacando sua capacidade de lidar com diferentes situações, desde desafios no reconhecimento facial até a integração eficaz com o sistema principal. Essa análise minuciosa é valiosa para orientar futuras melhorias e refinamentos, consolidando o sucesso do projeto como um todo.

## VIII. CONSIDERAÇÕES FINAIS.

Ao longo deste trabalho, exploramos a necessidade urgente de otimizar o tempo de espera dos pacientes em ambientes de saúde, especialmente em meio à pandemia de COVID-19. A falta de estrutura evidenciada pelos longos períodos de espera em filas para atendimentos e exames tornou-se o ponto de partida para o desenvolvimento do projeto aqui apresentado.

A implementação do sistema de reconhecimento facial mostrou-se uma solução eficaz para agilizar o processo de check-in, eliminando a necessidade de esperas prolongadas. Através da aplicação principal, os pacientes têm a conveniência de agendar consultas e exames, enquanto a confirmação automática de sua presença, através da aplicação do totem, evita a tradicional espera em filas.

O sucesso do projeto reflete-se não apenas na eficiência operacional alcançada, mas também na melhoria da experiência do paciente. A eliminação da etapa inicial de atendimento manual pode reduzir o tempo de espera e permite uma distribuição mais equitativa dos recursos.

A integração do reconhecimento facial pode proporcionar uma resposta ágil às demandas do ambiente de saúde contemporâneo. O paciente desfrutará de um processo de atendimento mais rápido e eficaz, que contribuirá para um atendimento mais humanizado.

Além disso, a implementação do sistema contribui para a mitigação dos riscos de exposição à COVID-19, destacando sua relevância em um contexto de pandemia. A eficácia do reconhecimento facial como ferramenta tecnológica não apenas simplifica procedimentos administrativos, mas também se traduz em benefícios tangíveis para a saúde pública.

Em resumo, a conclusão bem-sucedida deste projeto confirma a viabilidade e eficácia do reconhecimento facial como uma solução inovadora para otimizar a gestão de atendimentos em ambientes de saúde. O caminho trilhado desde a identificação do problema até a implementação bem-sucedida demonstra não apenas a importância da tecnologia na área da saúde, mas também a capacidade de adaptação e inovação diante dos desafios contemporâneos. Este projeto representa um passo significativo na melhoria do sistema de atendimento médico, oferecendo benefícios tangíveis para pacientes, profissionais de saúde e a comunidade como um todo.

## REFERÊNCIAS

- [1] Brasileiros enfrentam saga para conseguir realizar teste de covid-19. **Exame**, 13 de janeiro de 2022. Disponível em: <https://exame.com/brasil/brasil-enfrenta-saga-para-conseguir-realizar-teste-de-covid-19/>. Acesso em: 25 de set. de 2022.
- [2] Covid drives biometric uptake but users fear spoofing. **National Library of Medicine**, 25 de maio de 2021. Disponível em: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8148536/>. Acesso em: 19 de out. de 2022.
- [3] GARRIDO, Ricardo. O que tem de novo no uso de reconhecimento facial em e-commerces. **E-commerce Brasil**, 19 de set. de 2022. Disponível em: <https://www.ecommercebrasil.com.br/artigos/reconhecimento-facial-e-commerce>. Acesso em: 19 de out. de 2022.
- [4] COELHO, Leandro Callegari. Porque as filas de espera são uma tortura. **Logística Descomplicada**, 27 de nov. de 2012. Disponível em: <https://www.logisticadescomplicada.com/por-que-as-filas-de-espera-sao-uma-tortura/>. Acesso em: 25 de out. de 2022.
- [5] Higor. Introdução ao Padrão MVC. **DevMedia**, 2013 Disponível em: <https://www.devmedia.com.br/introducao-ao-padrao-mvc/29308>. Acesso em: Outubro de 2022.
- [6] Guia prático dos totens de autoatendimento (versão 2022). **Aqua**, jan. de 2021. Disponível em: <https://www.aqua.com.br/blog/totens-de-autoatendimento>. Acesso em: Outubro de 2022.

- [7] TANZU, VMware. Spring Projects. **Spring**. Disponível em: <<https://spring.io/projects>>. Acesso em: Outubro de 2022.
- [8] Spring Tutorial – A Java Framework Providing Efficiency. **Edureka**. Disponível em: <<https://www.edureka.co/blog/spring-tutorial/>>. Acesso em: Dezembro de 2023.
- [9] Flask Documentation. **Flask**. Disponível em: <<https://flask.palletsprojects.com/en/3.0.x/>>. Acesso em Dezembro 2023.
- [10] Ranking de sistemas de banco de dados mais usados em 2021 e 2022. **Austrian IT Consulting**. Disponível em: <http://db-engines.com/en/>. Acesso em: 06/04/2022.
- [11] SABINO, Roberto. Kanban: o que é, o Método Kanban, principais conceitos e como funciona no dia a dia. **Alura**, 30 de jun de 2023. Disponível em: <<https://www.alura.com.br/artigos/metodo-kanban>>. Acesso em Julho 2023.
- [12] O que é o Scrum?. **Amazon AWS**. Disponível em: <<https://aws.amazon.com/pt/what-is/scrum/>>. Acesso em Julho 2023.
- [13] PAIXÃO, João Roberto. O que é SOLID: O guia completo para você entender os 5 princípios da POO. **Medium**, 6 de jan. De 2019. Disponível em: <<https://medium.com/desenvolvendo-com-paixao/o-que-%C3%A9-solid-o-guia-completo-para-voc%C3%AA-entender-os-5-princ%C3%ADpios-da-poo-2b937b3fc530>>. Acesso em: Abril de 2023
- [14] MARTIN, Robert C. Clean Code: A Handbook of Agile Software Craftsmanship. Prentice Hall, 2008.

#### APÊNDICE

GitHub Disponível em: <<https://github.com/Alejunior32/TCC-EngenhariaComputacao>>. Acesso em: junho de 2023.