UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
COINT - TECNOLOGIA EM SISTEMAS PARA INTERNET
CURSO DE TECNOLOGIA EM SISTEMAS PARA INTERNET

LUCAS NICOLÁS WENZEL VALLADARES

UTFACE: UMA API RESTFUL PARA GERENCIAMENTO DE PRESENÇA COM RECONHECIMENTO FACIAL

PROJETO DE TRABALHO DE CONCLUSÃO DE CURSO

GUARAPUAVA
2019

LUCAS NICOLÁS WENZEL VALLADARES

UTFACE: UMA API RESTFUL PARA GERENCIAMENTO DE PRESENÇA COM RECONHECIMENTO FACIAL

Projeto de Trabalho de Conclusão de Curso apresentado ao Curso de Tecnologia em Sistemas para Internet da Universidade Tecnológica Federal do Paraná, como requisito parcial para a obtenção do título de Tecnólogo.

Orientador: Prof. Dr. Roni Fabio Banaszewski
Universidade Tecnológica Federal do Paraná

Coorientador: Prof. Me. Guilherme da Costa Silva
Universidade Tecnológica Federal do Paraná

GUARAPUAVA
2019

Dedico esse trabalho aos meus tutores, desde meus pais e avós, e professores de escola até aos professores do curso. Sem eles não poderia ter chegado até esta fase de minha vida.

Eu denomino meu campo de Gestão do Conhecimento, mas você não pode gerenciar conhecimento. Ninguém pode. O que pode fazer - o que a empresa pode fazer - é gerenciar o ambiente que otimize o conhecimento. (PRUSAK, Laurence, 1997).

RESUMO

VALLADARES, L. N. W.. UTFace: Uma API RESTful para gerenciamento de presença com reconhecimento facial. 2019. 21 f. Projeto de Trabalho de Conclusão de Curso – Curso de Tecnologia em Sistemas para Internet, Universidade Tecnológica Federal do Paraná. Guarapuava, 2019.

A atividade de realizar o controle de presença em sala de aula e ou em eventos é sempre algo que deixa a desejar, pois além de demandar certo tempo, podem acontecer equívocos tanto por quem realiza a chamada quanto por quem a responde. Além disso, podem haver fraudes se caso uma pessoa não estiver presente e outra responder ou assinar por ela. Como solução, o controle de presença pode ser realizado com tecnologias de reconhecimento facial. Neste sentido, um sistema poderia capturar uma imagem das pessoas em um ambiente e identificar todos os presentes, assim evitando que o indivíduo presente fique com falta ou que o faltante fique com presença. Atualmente existem vários algoritmos de reconhecimento facial de código aberto, os quais podem ser usados e aprimorados para criar um sistema que seja útil no cotidiano universitário. Com o desenvolvimento deste sistema, pretende-se trazer uma inovação tecnológica para a Universidade Tecnológica Federal do Paraná - UTFPR e realizar melhorias e atualizações em métodos, tarefas e atividades realizadas dentro da universidade.

ABSTRACT

VALLADARES, L. N. W.. Attendance Control System with Facial Recognition. 2019. 21 f. Projeto de Trabalho de Conclusão de Curso – Curso de Tecnologia em Sistemas para Internet, Universidade Tecnológica Federal do Paraná. Guarapuava, 2019.

The activity of making attendance call at a classroom or events are always something which lets some doubts, because either of taking some time to do it, can happen some mistakes from who makes it or who is answering the calling. And above that can happen some frauds if someone is not in the room and someone else answers the call or sign for that person. To solve this problem can be implemented a facial recognition system, which can capture an image of the people inside the room and recognize each one in that picture, avoiding the one who is there to get a fault or the one missing to get a presence. Nowadays have plain open-source facial recognition algorithms which allow us to implement a platform that will be useful in our day-a-day at the university. With the development of this technology, we hope to bring technological innovation to the Federal Technological University of Paraná and carry out improvements and updates on methods, tasks, and practices performed in our institution.

LISTA DE FIGURAS

LISTA DE ABREVIATURAS E SIGLAS

IA	Inteligência Artificial
LBPH	Local Binary Patterns Histograms (Histogramas de Padrões de Binário Local)
API	Application Programming Interface (Interface de Programação de Aplicativos)
REST	Representational State Transfer (Transferência Representacional de Estado)
HTTP	Hypertext Transfer Protocol (Protocolo de Transferência de Hipertexto)
TDD	Test Driven Development (Desenvolvimento Orientado por Testes)
PCA	Primary Component Analysis (Análise de Componentes Principais)
LS	Least Squares (Mínimos Quadrados)
LDA	Linear Discriminant Analysis (Análise Linear Discriminante)
DA	Discriminant Analysis (Análise Discriminante)
LBP	Local Binary Pattern (Padrão de Binário Local)
HOG	Histogram of Oriented Gradients (Histograma de Gradientes Orientados)
HTML	Hypertext Markup Language (Linguagem de Marcação de Hipertexto)
CSS	Cascading Style Sheet (Folha de Estilo em Cascatas)
CERN	European Council for Nuclear Research (Organização Europeia para Pesquisa Nuclear)
W3C	World Wide Web Consortium (Consórcio World Wide Web)
OpenCV	Open Source Computer Vision Library (Biblioteca Multiplataforma de Livre Uso)
IoC	Inversion of Control (Inversão de Controle)
DI	Dependency Injection (Injeção de Dependência)

SUMÁRIO

1 INTRODUÇÃO

Há anos, a Inteligência Artificial (IA) representava uma tecnologia totalmente futurística onde era retratada em filmes e livros como algo que fazia parte do cotidiano das pessoas, e era vista, por muitas pessoas, até mesmo como uma ficção. Nos dias atuais, a IA tem se tornado uma realidade, com algoritmos inteligentes que aprendem, reconhecem padrões e também tomam decisões de forma espontânea. Esses algoritmos são capazes de realizar tarefas interessantes, tal como o reconhecimento facial que, por exemplo, é muito utilizado por autoridades, nos países mais desenvolvidos, para detectar criminosos disfarçados e foragidos entre uma multidão de civis.

A tecnologia de reconhecimento facial é uma tecnologia com grande potencial de aplicação. Algumas de suas funções podem ser adaptadas para serem aplicadas em instituições e universidades. Estas poderiam utilizar da tecnologia para realização de chamada e controle de frequência em sala de aula, palestras e eventos ou até mesmo para áreas de acesso restrito. Com isso, os responsáveis por tais funções conseguiriam mais agilidade na realização de suas tarefas, sem ter que interromper a sua principal atividade e de maior importância. Pode-se levar em conta também outras razões as quais a implementação desse método seria bastante importante, como por exemplo, no intuito de evitar fraudes ou engano quando só se pode esperar de boa fé que tenha sido a própria pessoa que tenha assinado uma lista de presença ou evitar que por equívoco do professor ou por falta de atenção do aluno, aconteça a atribuição de falta quando a pessoa está presente.

Algumas tecnologias de IA ainda são muito custosas por demandar equipamentos de alta qualidade e tecnologias proprietárias, o que dificulta a popularização da aplicação das mesmas. Porém, ao buscar a utilização de tecnologias de acesso livre, adaptação das mesmas e com estudo sobre o assunto, é possível conseguir uma solução mais barata para aplicação efetiva da IA no cotidiano das pessoas. Com isso, por exemplo, a aplicação de tais tecnologias pode impactar positivamente a produtividade no âmbito universitário.

1.1 ORGANIZAÇÃO DO TRABALHO

Este documento está organizado da seguinte forma: na Seção 2.1 são apresentados os objetivos gerais e específicos do projeto. A Seção 2.2 descreve o diferencial tecnológico. A Seção 3 está descrita a resenha literária. A Seção 3.1 traz o estado da arte. A Seção 3.2 apresenta a fundamentação teórica para o projeto. A Seção 4 apresenta a metodologia para o mesmo. Na Seção 5 o desenvolvimento. Na Seção 6 as considerações finais e por fim na Seção 7 estão listadas as referências.

2 OBJETIVOS

Nesta seção serão apresentados o objetivo geral e os objetivos específicos referente a este projeto.

2.1 Objetivo Geral

O objetivo desse trabalho é estudar três algoritmos de reconhecimento facial mais tradicionais de código aberto (i.e. EigenFaces, FisherFaces e LBPH) para verificar qual apresenta o melhor desempenho para então, implementar uma API para ser consumida futuramente por uma aplicação que permitirá a verificação de presença de alunos em sala de aula e/ou eventos com a tecnologia de reconhecimento facial.

2.2 Objetivos Específicos

Os objetivos específicos são:

- Capturar imagens e armazená-las em um sistema de arquivos;
- Definir o algoritmo de reconhecimento facial de código aberto com melhor desempenho;
- Realizar o treinamento dos algoritmos de acordo com a base de imagens capturadas;
- Identificar pessoas por meio de imagens a fim de gerar as listas de presença preenchidas;
- Desenvolver uma API RESTful para abstrair operações de entrada e saída para o algoritmo de reconhecimento facial;
- Desenvolver uma documentação para a API;

3 METODOLOGIA

Este capítulo apresenta a metologia a ser utilizada para o desenvolvimento do projeto. A metologia será dividida em etapas que serão abordadas e descritas a seguir:

3.1 Levantamento dos requisitos do sistema

Primeiramente, para embasar a construção dos artefatos referente a modelagem do projeto, os requisitos foram coletados utilizando a técnica de brainstorming com a presença do Professor orientador. Também, para facilitar a detecção de tais requisitos, foi utilizado a técnica de prototipação de telas de um aplicativo a ser construído em trabalhos futuros para melhor visualização e entendimento do escopo do problema.

3.2 Modelagem do sistema

Após o levantamento de requisitos, deu-se a construção da arquitetura do sistema, basicamente no lado do servidor. A arquitetura permite antever como será feita a integração de dados, aonde serão armazenados os dados e como serão processados. Neste momento, projeta-se também o modelo da base de dados. Com isso, consegue-se também projetar a própria API RESTful.

3.3 Estudo e definição das tecnologias a serem utilizadas

Para o desenvolvimento do projeto, também é necessário adquirir conhecimento sobre as tecnologias que serão utilizadas. Desta forma, se faz necessário o estudo e aprofundamento em conceitos de frameworks e dos algoritmos de reconhecimento facial.

3.4 Comparação entre os algoritmos

Os algoritmos de reconhecimento facial serão submetidos a testes em diferentes situações. Com isto, será possível realizar a comparação entre os algoritmos e definir qual algoritmo tem melhor desempenho para ser utilizado para a resolução do problema deste projeto.

3.5 Desenvolvimento do sistema e testes do sistema

Nesta fase dá-se o início ao desenvolvimento da API, acompanhado de testes para verificar seu correto funcionamento através da técnica de Desenvolvimento Orientado a Testes (TDD).

3.6 Validação dos requisitos implementados

Os requisitos implementados serão validados por meio da apresentação a um grupo de professores. Como estes são os principais interessados no contexto da aplicação, esta entrega do protótipo busca provocar e conseguir um maior apoio para a implementação de uma aplicação futura que seja usada por tais professores para agilizar suas atividades em sala de aula.

3.7 Cronograma de desenvolvimento do projeto

Atividades	TCC 1					TCC 2				
	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez
1. Revisão dos apontamentos da banca.										
2. Revisão bibliográfica.										
3. Redação do projeto de TCC.										
4. Atividades Preparativas.										
5. Defesa do projeto de TCC.										
6. Atividades de Execução.										
7. Atividades de Validação.										
8. Escrita da Monografia de TCC.										
9. Elaboração da apresentação final.										
10. Defesa final do TCC.										

Figura 1 – Cronograma de Desenvolvimento

4 RESENHA LITERÁRIA

Neste capítulo serão apresentados algoritmos correlatos de reconhecimento facial de livre uso que são utilizados atualmente no mundo todo, entre eles o Eigenfaces que é o algoritmo mais antigo de reconhecimento facial (BISSI, 2018), a fim de adquirirmos um comparativo entre os algoritmos e embasamento para utilização dos mesmos. Também são abordadas, em estado da arte, algumas aplicações que utilizam reconhecimento facial.

4.1 Estado da Arte

4.1.1 FindFace

É um aplicativo desenvolvido na Rússia que usa a tecnologia de reconhecimento de faces de pessoas. O aplicativo funciona de forma que ao fotografar uma pessoa, a imagem é comparada com os rostos das redes sociais com uma precisão de 70 por cento. Segundo as estatísticas de 2016, o aplicativo já tinha conseguido identificar 500 mil usuários e processado cerca de 3 milhões de buscas. Desde o dia 1 de Setembro de 2018, a empresa finalizou seus serviços com esta aplicação e passou a prestar serviços de reconhecimento facial para o governo (LEANDRO, 2019). A empresa oferece para aqueles que querem utilizar o sistema, um sistema similar chamado FindFace.PRO.

Figura 2 – Logo FindFace

Fonte: (FINDFACE, 2018)

4.1.2 Face++

É uma plataforma chinesa de extrema qualidade que superou a precisão dos algoritmos do Facebook e da Google. Ela já está sendo utilizada nas universidades para realização das chamadas presenciais (ELOLA, 2018). A plataforma permite a comparação de duas faces,

buscar uma face no banco de dados da plataforma com base em uma imagem que tenha sido colocada para fazer a busca e cada face detectada é armazenada para futuras análises. Todas as funções do sistema retornam uma pontuação e limites de confiabilidade das comparações.

Figura 3 – Logo Face++

Fonte: (FACE++, 2018)

4.1.3 CredDefense

É uma plataforma brasileira de consignação de crédito. Os dados biométricos são fornecidos pelo próprio cliente e mantidos pela empresa que mantém a plataforma. A empresa se compromete a gerar códigos criptografados das características faciais de seus clientes e armazená-los em base de dados, a fim de evitar fraudes e realizar autenticação em transações (SULIVAN, 2018).

Figura 4 – Logo Creddefense

Fonte: (CREDDEFENSE, 2018)

4.1.4 TÓV-TEC Expert

Esta é uma plataforma brasileira para controle de acesso à áreas restritas. O Expert Fixo funciona com uma câmera instalada na entrada da localidade, como catracas ou barreiras de acesso, que faz o reconhecimento da face e compara com seu banco de dados de faces cadastradas para aquela área e caso seja confirmado, é liberada a passagem para essa pessoa (TÓV-TEC, 2017).

Figura 5 – Logo Tóv-Tec

Fonte: (TóV-TEC, 2018)

4.2 Fundamentação Teórica

Nesta seção serão apresentadas as tecnologias que serão utilizadas para implementação da API que será responsável por realizar o reconhecimento facial. Com o intuito de obter maior conhecimento teórico, um estudo foi realizado sobre as linguagens de programação, banco de dados e linguagens de marcação. Este estudo também contribui para uma visão mais concreta para o projeto e para conhecer como ele será colocado em prática.

4.2.1 Algoritmos de Reconhecimento Facial

4.2.1.1 EigenFaces

A prática de realizar o reconhecimento facial consiste em isolar entradas ou dados de uma imagem em várias classes, as quais podem se referir a diferentes pessoas. Essas entradas podem apresentar ruídos, os quais podem ser causados por diferentes condições de iluminação, posição da pessoa, sombras, entre outros. No entanto, apesar das diferenças entre uma imagem e outra, existem padrões quando se leva em conta imagens de uma mesma pessoa. Esses padrões podem ser observados em todos os sinais de entrada, como a presença de olhos, nariz, boca, bem como certas características de uma face, e também, as distâncias relativas destes membros que são caracterizados como objetos na imagem. Essas características são chamadas de "eigenfaces"no domínio de reconhecimento facial. Essas características são extraídas da imagem por uma ferramenta chamada Análise de Componentes Principais (PCA) (Z.; LI, 2011).

Por meio da PCA é possível extrair essas características de todo o conjunto de imagens de treinamento e formar, por meio destas, uma base de reconhecimento, como uma imagem média de todas essas características, que juntas, somadas compõem uma imagem original da pessoa em questão. Cada face é um valor, e cada característica tem um valor próprio, e para que a face esteja completa, todas as características somadas devem chegar ao valor da face ou ao menos bem próximo para ser reconhecida como aquela face. Deve-se levar em conta que ao juntar todos os valores ou pesos dessas características ou eigenfaces, alguns dados podem ser perdidos na composição da imagem. Neste caso, a formação da imagem não seria perfeitamente igual à imagem original.

Duas coisas devem ser levadas em consideração. A primeira delas é que quando uma imagem é desmanchada por assim dizer, e os pesos das características principais de um rosto forem muito diferentes dos padrões, aquele objeto não será reconhecido como uma face. A segunda consideração é que quando temos faces muito similares, os pesos de suas características serão muito similares. Portanto, estas poderão ser consideradas como a face de uma mesma pessoa. Isto pode gerar uma margem de erro se duas pessoas forem muito parecidas fisicamente.

O algoritmo é chamado de eigenfaces pois ele utiliza de eigenfaces (as características principais de uma imagem) para realizar a detecção facial (BELHUMEUR; HESPANHA; KRIEGMAN, 1997). Ele se comporta da seguinte forma: primeiro, as imagens coletadas do conjunto de treinamento são transformadas em um conjunto de eigenfaces. Então, cada imagem será dividida em pesos para suas eigenfaces e armazenadas em um arquivo. Ao detectar uma face em uma imagem desconhecida, os pesos são calculados para suas eigenfaces e então comparadas com os pesos que foram anteriormente gravados no arquivo.

A maneira que são comparados esses pesos é por distância de valor entre os pesos, pois cada imagem, será dividida em pesos distintos. Portanto, seria quase impossível ter duas imagens distintas de uma mesma face e obter pesos totalmente similares. Assim, a distância que for menor em relação à base de pesos gravados, corresponderá à face que está sendo detectada. Quando os valores detectados forem muito distantes de qualquer valor em base de dados, a mesma não será considerada como uma face.

4.2.1.2 FisherFaces

Um problema corrente, quando se trata de visão computacional, reconhecimento de padrões e até mesmo aprendizado de máquina, é a definição da representação dos dados apropriados para realização de determinadas tarefas. Assim como o algoritmo EigenFaces, o algoritmo FisherFaces utiliza o PCA para dividir uma imagem total em diferentes partes que representam a maior parte de variação de dados em uma imagem, produzindo assim, um conjunto de eigenfaces (BELHUMEUR; HESPANHA; KRIEGMAN, 1997). Esses autovetores detectados são comparados com os autovetores da matriz obtida pelo treinamento das imagens, assim, esses autovetores são correspondentes à solução de mínimos quadrados (LS). O LS é uma excelente maneira de representação de dados, pois garante que a variação de dados entre

uma imagem e outra seja mantida, eliminando assim as características originais de uma face e focando nas partes de maior variação.

Quando o objetivo se trata de classificação e não de representação, a solução LS pode não produzir os resultados esperados. Nesses casos, deseja-se encontrar um subespaço que mapeie os vetores de amostra da mesma classe em um único ponto da representação do recurso e nos de classes diferentes que estão o mais distantes um do outro quanto possível. As técnicas derivadas para atingir esse objetivo são conhecidas como análise discriminante (DA).

A DA mais conhecida é a Análise Linear Discriminante (LDA), que pode ser derivada de uma ideia sugerida pela R.A. Fisher em 1936 (BELHUMEUR; HESPANHA; KRIEGMAN, 1997). Quando o LDA é usado para encontrar a representação do subespaço de um conjunto de imagens de faces, o resultado desses vetores de base que definem esse espaço são conhecidos como Fisherfaces (Z.; LI, 2011).

Para computar os Fisherfaces, assume-se que os dados em cada classe são normalmente distribuídos. Faz-se então uma distribuição normal multivariada, definindo uma matriz média e a covariância, e define-se uma função de densidade de probabilidade.

Em síntese, o FisherFaces é definido pela maior quantidade de variação de uma face, trabalhando assim, não com as características padrões de uma face e sim naquilo que varia entre uma face e outra, utilizando de comparativos de distância entre matrizes.

4.2.1.3 LBPH

O LBPH ou (Local Binary Pattern Histogram) tem sua origem do LBP (Local Binary Pattern) que é um padrão de binário local, um tipo de característica utilizado para classificações. O LBP consiste em uma textura em modelo de espectro que foi proposto em 1990 (WANG; HE, 1990). Finalmente, em 1994 (OJALA; PIETIKäINEN; HARWOOD, 1996), esse padrão foi publicado e a partir desse momento foi descoberta uma das melhores formas de classificação para texturas e também foi concluído que a combinação com HOG (Histogram of Oriented Gradients) levava a uma notável melhora de desempenho em detectar os dados (WANG; YAN; HAN, 2009).

Basicamente, o algoritmo funciona gerando uma matriz, onde cada posição guarda o valor dos pixels dessa parte da imagem. Então, é feita uma comparação entre eles utilizando de vizinhança circular e utilizando uma equação para formar uma matriz binária. A partir dessa matriz, é gerado um histograma que é utilizado para realizar o reconhecimento facial, utilizando os valores de vizinhança.

Para formar um histograma é possível utilizar diferentes critérios de LBP como invariante à rotação ou uniforme, que desta forma pode gerar uma maior discriminação em comparações aos histogramas de padrões individuais. Desta forma, a frequência em que os padrões do LBPH ocorrem é muito rara, então as probabilidades não poderiam ser confiadas (PARRA, 2014).

4.2.2 Tecnologias de Desenvolvimento

4.2.2.1 Linguagem de Marcação HTML, Linguagem de Estilização CSS e Linguagem de Programação JavaScript

A sigla HTML (Hypertext Markup Language) significa Linguagem de Marcação de Hipertexto. Isto quer dizer que é uma linguagem feita para interpretação de textos com marcações, ou seja, para criação de páginas web que serão interpretadas pelos navegadores. Hoje em dia, os navegadores mais utilizados são Google Chrome, Mozila Firefox, Microsoft Edge (substituiu o Internet Explorer), Safari e Opera (ALVES, 2016).

Quando se desenvolve para internet, é comum a utilização de várias linguagens para criar uma aplicação ou website. Para que todas essas linguagens funcionem juntas em uma única aplicação, é necessário separar os códigos em camadas. Como exemplo, o HTML é usado para estruturação, o CSS para estilo e aparência e o JavaScript para tratar eventos e manipular dinamicamente a estrutura e aparência de uma página. Também é possível utilizar linguagens para cuidar de toda a parte de servidor, banco de dados e também de toda a parte lógica do sistema, como por exemplo cálculos. Exemplo destas linguagesm são: PHP, Python, Ruby, Java, C, C++, etc.

O HTML foi criado por Tim Berners-Lee em 1991 no CERN (European Council for Nuclear Research) na Suíça (CASTRO; HYSLOP, 2012). O HTML foi criado para integrar instituições de pesquisas próximas e compartilhar documentos com facilidade. Em conjunto com a biblioteca de desenvolvimento WWW (World Wide Web), que foi liberada em 1992, o HTML alcançou proporção mundial.

A linguagem CSS (Cascading Style Sheet) é definida pela World Wide Consortium (W3C) como folha de estilo em cascata. Essa definição diz que CSS é um mecanismo para adicionar estilo às páginas algum estilo, como por exemplo fontes, espaçamentos, cores, etc (CASTRO; HYSLOP, 2012).

O JavaScript é uma linguagem de programação (HAVERBEK, 2018) que permite com que você implemente partes mais complexas de uma aplicação web no lado cliente e também no servidor. Normalmente, é utilizado para tratar eventos e ações de usuários no lado cliente. Basicamente, o JavaScript é responsável pela maioria dos elementos dinâmicos de uma aplicação, ou seja, informações que se atualizam continuamente, como controle de multimídia, imagens animadas, gráficos, etc. O diferencial do JavaScript é que pode ser utilizado diretamente no código HTML apenas colocando o conteúdo dentro de uma tag "script".

4.2.2.2 OpenCV e JavaCV

O OpenCV (Open Source Computer Vision Library) é uma biblioteca multiplataforma de código aberto desenvolvida pela Intel para desenvolvimento na área de visão computacional (CULJAK et al., 2012). Ela provê uma infraestrutura para as aplicações nessa área e também acelera a percepção de máquinas para produtos, utilizando de módulos de processamento de

imagem e vídeo. Ela possui mais de 2500 algoritmos otimizados de visão computacional e aprendizado de máquina para uso no reconhecimento de objetos ou pessoas, análises estruturais, calibração de câmera, filtros de imagem, e inclusive para extração de modelos 3D de objetos.

Os algoritmos da OpenCV são utilizados por várias grandes empresas e startups. Estes podem ser utilizados em diferentes aplicações. Por exemplo, uso na vigilância pelo governo de Israel; monitoramento de equipamentos de mineração na China; detecção de acidentes de afogamentos em piscinas na Europa e até mesmo reconhecimento e marcação de produtos no Japão (ELOLA, 2018).

Para utilização de tais algoritmos com Java, existe uma extensão da biblioteca OpenCV que se chama JavaCV. Esta contêm classes facilitadoras para programação em diversas plataformas, inclusive para Android. O JavaCV também contêm aceleradores de hardware para exibição de imagem em tela cheia utilizando canvas, e entre outros aprimoramentos para se trabalhar com imagens. (DAVISON, 2013).

4.2.2.3 API Restful

Uma API (Apllication Programming Interface) é conjunto de padrões e rotinas que são estabelecidos e documentados para que seja utilizada por uma aplicação (PIRES, 2017). Não é necessário que aplicação conheça de fato os detalhes da implementação do software, utilizando apenas seus métodos a fim de integrar dados entre diferentes fontes e softwares ou aplicações e até mesmo usuários.

Uma API é considerada RESTful quando atende os princípios de REST (Representational State Transfer). Basicamente, REST é a abstração de uma arquitetura e comunicação utilizada no desenvolvimento web, que permite a criação de uma interface bem definida permitindo que as aplicações consigam se comunicar de forma simples. (RICHARDSON, 2013).

4.2.2.4 Spring

Spring é um framework Java de código aberto (open source), criado por Rod Johnson, em 2003 (JOHNSON, 2003), que foi criado no intuito de auxiliar os programadores Java a desenvolverem seus sistemas, sem terem que se preocupar demasiadamente com configurações avançadas. O framework utiliza de Inversão de Controle (IoC) e Injeção de Dependências (DI), a fim de facilitar o desenvolvimento de aplicações. O Spring fornece módulos de persistência de dados, integração, segurança e testes, entre outros, que são muito mais fáceis de compreender e implementar em uma aplicação.

A IoC permite que seja transferido a outro elemento o controle de quando e como um objeto deve ser criado ou instanciado ou quando um método deve ser chamado e executado. Ao utilizar o Spring, o programador não deve mais se preocupar em gerenciar isto, transferindo essa responsabilidade ao elemento que é chamado de "container". Na prática, isso pode ser aplicado, por exemplo, na interface do usuário onde não será mais necessário ficar escutando um componente para ser disparada uma ação do usuário, basta implementar diretamente a

ação para os botões. A DI é uma das formas de implementar a IoC, deixando com que a DI cuide das dependências (EFRAIM, 2012).

5 DESENVOLVIMENTO

O objetivo deste projeto é pesquisar a respeito dos algoritmos de reconhecimento facial e implementar uma API que poderá ser implementada futuramente em uma aplicação para realização de chamada de presença a partir do reconhecimento das faces dos presentes, trazendo assim inovação tecnológica, para algo que até os dias atuais são feitos manualmente em um sistema de alta complexidade e algumas vezes até mesmo por uma lista em folha de papel que é passada de mãos em mãos para ser assinada.

Com esta plataforma os professores poderão realizar a chamada de presença de forma mais segura e rápida e os alunos receberão de imediato uma notificação se foram contados presentes ou não, o que hoje não pode ser feito nem mesmo após o lançamento no sistema das faltas e presenças.

Para hospedagem das fotos coletadas, será feita a contratação da DigitalOcean, e adquirida uma máquina para a hospedagem também do sistema que executará o treinamento das imagens submetidas por meio da API e o reconhecimento facial das faces em imagens enviadas pelos professores a partir das fotos submetidas em sala de aula pelo próprio aplicativo do professor.

Para a realização do reconhecimento facial, são necessárias três etapas, que são elas: a captura, que seria a coletânea de imagens de uma pessoa, o treinamento dessas imagens e por fim o reconhecimento.

A fim de testar os algoritmos propostos, será implementada uma API que será responsável por submeter as fotos coletadas ao servidor para executar o treinamento dessas imagens. Essa API também será responsável por fazer buscas, como buscar alunos de uma disciplina cadastrada, ou um professor de uma determinada disciplina, etc. Também será responsável por submeter os dados de status do aluno de presente ou ausente, ao submetermos via API uma imagem, que encaminhará ao servidor para executar o reconhecimento facial. Levando em conta que o treinamento gera apenas um arquivo para todas as imagens, inclusive de pessoas diferentes, então será um processo que custará processamento, e quanto mais imagens, mais processamento, por tanto, o treinamento inicial será feito uma única vez. Existe a possibilidade de um aluno submeter novas fotos pelo seu próprio aplicativo, caso isso aconteça, essas imagens serão armazenadas, e um "job" executará o processo de treinamento semanalmente em horário que o sistema não estiver em funcionamento, assim evitando um maior custo de processamento, como seria o caso de executar o treinamento das imagens cada vez que uma pessoa submeter uma nova imagem.

Para a coletânea de imagens ou fotos de uma pessoa, podem ser utilizadas fotos públicas, por exemplo, retiradas de um perfil de uma rede social. Após feita a coletânea de imagens, elas serão submetidas pela API para o servidor para realizar o treinamento das imagens.

Feito o treinamento das imagens, em sala de aula, ao submetermos uma imagem de

várias pessoas já cadastradas, ou também várias fotos diferentes a fim de simular uma sala de aula com as fotos repartidas, então o sistema deve identificar os alunos presentes naquela imagem e retornar o status de presença de cada aluno, se presente ou ausente. Caso o aluno não seja identificado, será possível fazer manualmente utilizando um método de nossa API e este seria o pior dos casos que poderia acontecer.

5.1 Análise do Sistema

Nesta seção serão apresentadas a documentação referente a coleta de requisitos que dá embasamento para as funcionalidades do que foram propostas no sistema.

Para coletar informações e projetar a ideia, se fez necessário realizar a prototipação das telas de como ficaria uma aplicação final utilizando a API que foi proposta a ser desenvolvida. A prototipação das telas será utilizada apenas para levantamento de requisitos.

5.1.0.1 Aplicação na visão do professor

Na figura 6 é apresentado protótipo de uma tela onde mostraria as disciplinas cadastradas para um determinado professor, e apenas essas, inicialmente, seriam as disciplinas que este mesmo professor poderia realizar a chamada presencial.

Figura 6 – Protótipo tela professor

Na segunda tela ainda da figura 6 está representada como o professor irá realizar a chamada presencial, podendo tirar uma foto ou até mesmo fazendo manualmente e então submetendo essa lista com o status do aluno em presente ou ausente.

Inicialmente existe uma lista de todos os alunos que foram matriculados na disciplina, e todos estão com o status de "ausente". A partir do momento que o aluno é reconhecido em uma foto submetida pelo professor, o status desse aluno passa a ser de "presente", podendo o professor ainda assim desmarcar ou marcar algum aluno que desejar, em caso de falha.

Na terceira tela da figura 6 está sendo representado o menu lateral do aplicativo onde constam as informações do professor e das disciplinas que ele pode realizar uma chamada presencial.

5.1.0.2 Aplicação na visão do administrador

Na figura 7 é ilustrado como seria o aplicativo na visão de um administrador, que poderá ter a função de cadastra uma disciplina e incluir os alunos à mesma. O administrador poderá também submeter fotos de um aluno ao castrá-lo no sistema, ou até mesmo atualizar seu cadastro submetendo novas fotos.

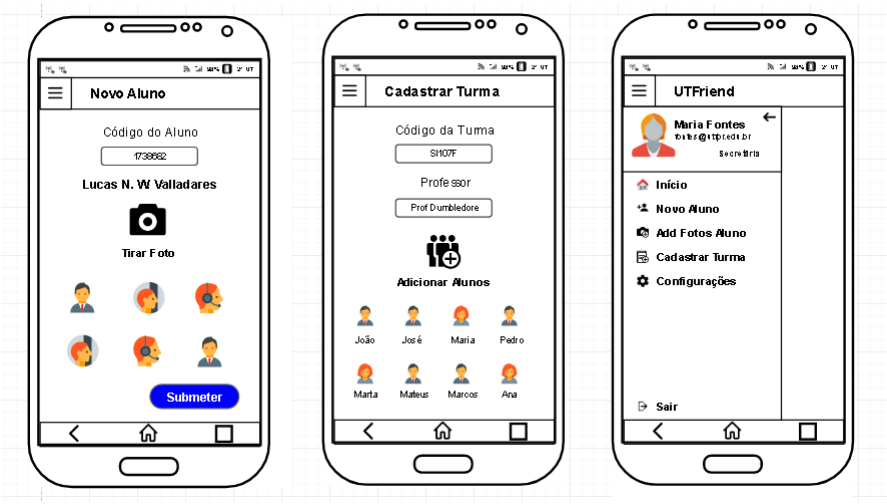

Figura 7 – Protótipo tela administrador

Na primeira tela da figura 7, é representado o cadastro de um aluno no sistema. Enquanto na segunda tela está representado o cadastro de uma disciplina e já a inclusão do professor responsável pela turma e os alunos que deverão participar da disciplina.

A terceira tela da figura 7, representa o menu lateral contendo as informações do administrador e as funções que podem ser realizadas pelo mesmo.

5.2 Projeto do Sistema

Nesta seção serão apresentados os artefatos resultantes do projeto do sistema. Os artefatos serão divididos em Projeto da API, Projeto da Base de Dados da API e Projeto de Arquitetura da Estrutura da Rede.

5.2.1 Projeto da API

Na figura 8, é demonstrado como será a documentação da API e suas funções REST. Apenas como exemplo, foi utilizado apenas um dos modelos das entidades que irão existir na API.

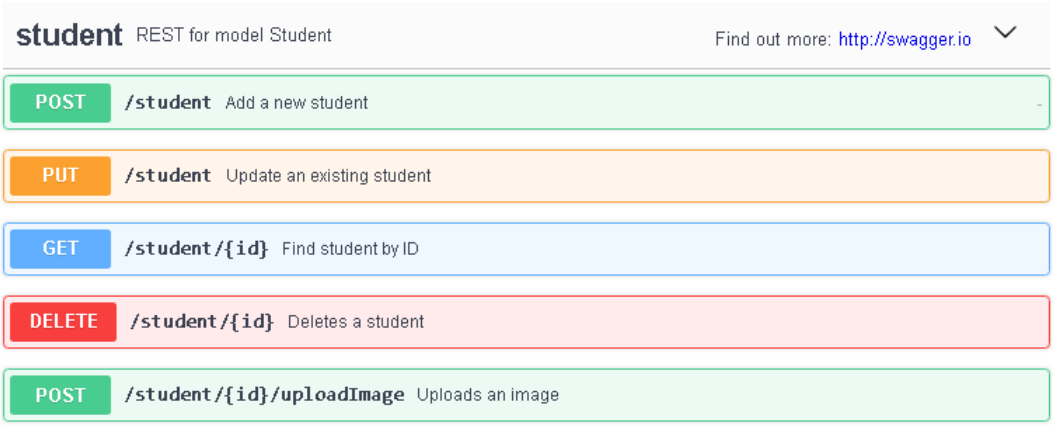

Figura 8 – Protótipo requisições REST

A API será documentada com o Swagger (SWAGGER, 2019), desta forma os usuários poderão facilmente utilizar a documentação para acessar as funções da API e até mesmo ver as entidades e os tipos de atributos que existem nas entidades, como pode-se observar na figura 9.

5.2.2 Projeto da Base de Dados da API

Na figura 10 está representada a Base de Dados, demonstrando o "schema"da base de dados e as ligações entre as tabelas, que é equivalente aos modelos de entidades demonstrados na figura 9.

A tabela de curso, representa os diferentes cursos que uma instituição pode ter, cada curso pode ter vários professores e várias disciplinas, enquanto os professores e disciplinas terão apenas um curso.

Um professor pode ter várias disciplinas, no entanto, cada disciplina pode ter apenas um professor, sendo assim uma relação de (1, n).

As disciplinas terão uma relação com aulas, pois assim, deixamos em aberto para que um evento possa ser considerado uma aula, e possa ser relacionada aos alunos, contendo um atributo para armazenar o status de presença de cada aluno.

A tabela de alunos também tem uma relação de (1, n) com a tabela de fotos. Um único aluno poderá ter várias fotos, no entanto, uma foto sempre irá pertencer a um único aluno.

Um usuário poderá ser um administrador, ou uma secretária. A tabela de usuário recebera um papel, que pode ser um papel de administrador, como mencionado. E a tabela de papel, também será relacionada aos professores e aos alunos.

5.2.3 Projeto de Arquitetura da Estrutura da Rede

Por meio de requisições HTTP serão submetidas imagens para treinamento ao servidor, e a API irá realizar o tráfego de dados e irá fornecer toda a documentação para que os métodos possam ser utilizados sem dificuldades. Os modelos das requisições HTTP e o corpo do Json

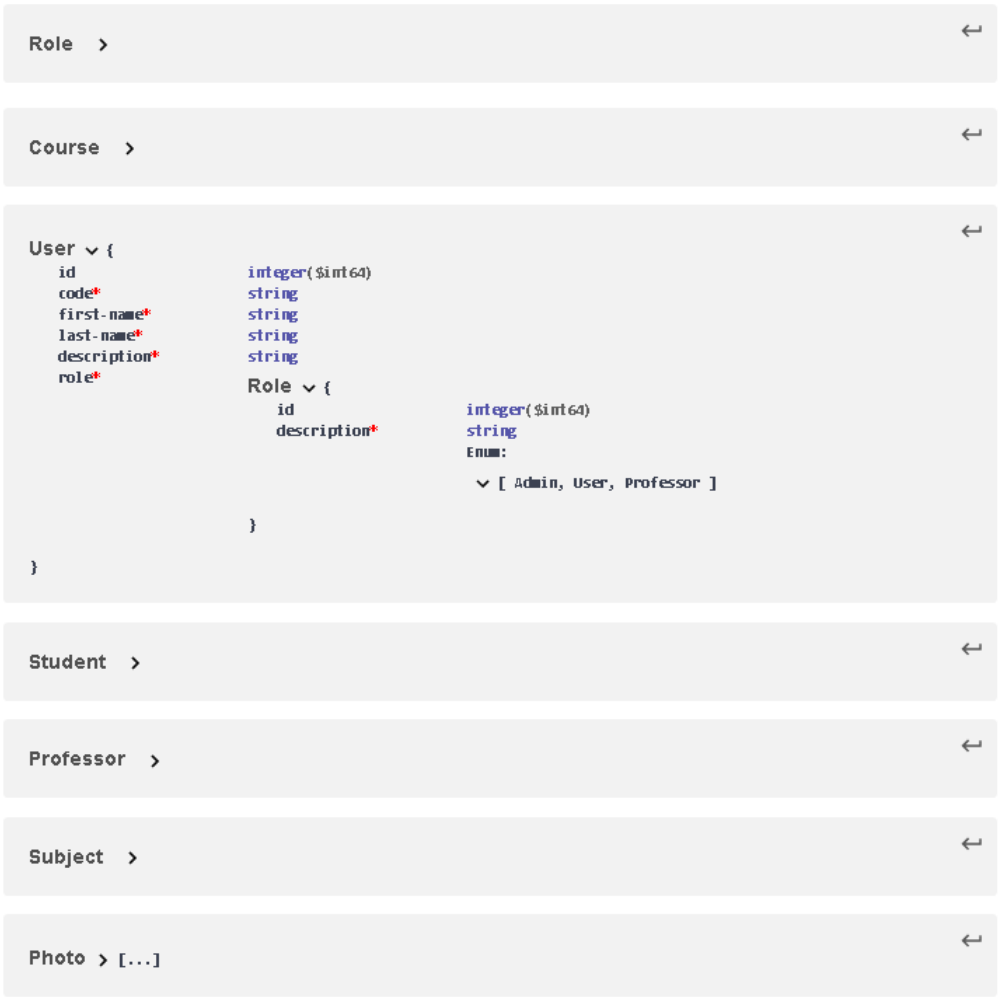

Figura 9 – Protótipo entidades na API

que devem ser submetidos poderão ser checados no Swagger (SWAGGER, 2019), plataforma que será utilizada para documentar a API.

Como representado na figura 11, as imagens submetidas serão armazenadas na nuvem. Como serão utilizadas inúmeras imagens e irá existir um grande tráfego desse tipo arquivos, será necessário contratar um plano, inicialmente o básico e gratuíto de uma plataforma de nuvem para armazenamento como o Cloudinary (CLOUDINARY, 2019).

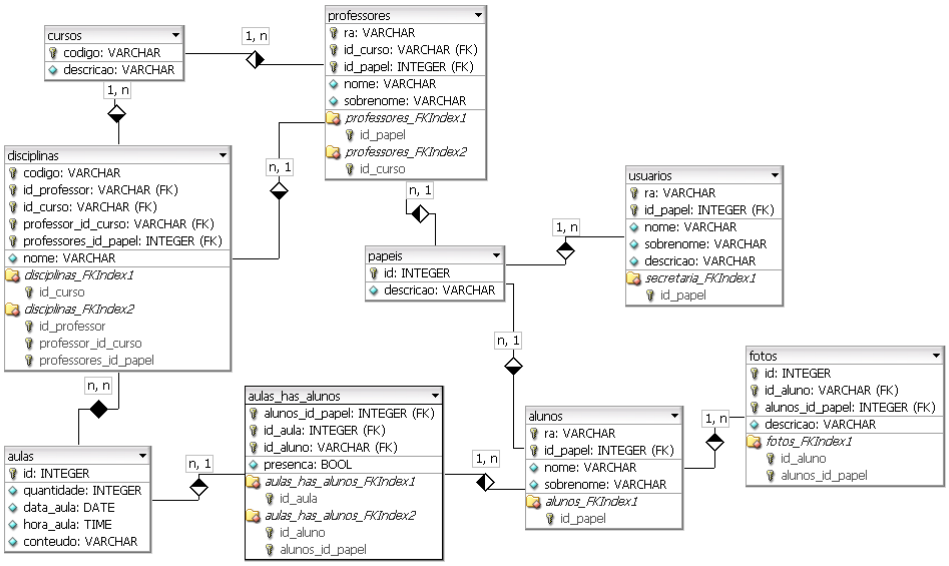

Figura 10 – Base de Dados

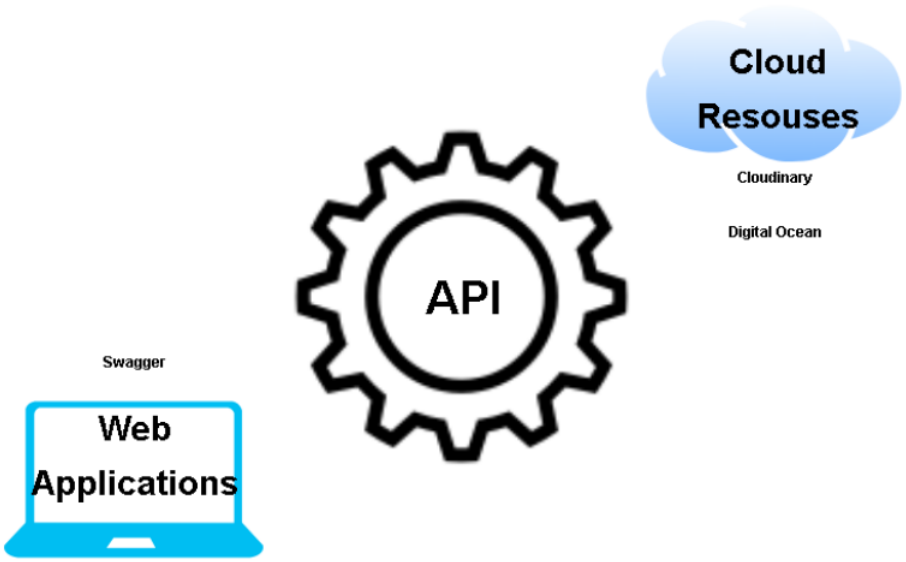

Figura 11 – Arquitetura da Estrutura da Rede

6 CONCLUSÃO

Com a realização de reuniões e discussões a fim de gerar um brainstorm foram coletados os requisitos para o projeto, com os quais foi possível definir uma visão do produto, que foi transformada em prototipação de telas, dando aos participantes do projeto a mesma ideia sobre o que está sendo proposto e projetado ao desenvolvimento.

O objetivo deste trabalho é desenvolver uma API que possa ser utilizada por um aplicativo para que os professores possam realizar a chamada de presença em sala de aula de forma automática, apenas submetendo uma ou várias fotos da turma para que cada aluno presente seja reconhecido e contado como presente na aula. Também poderá ser utilizada em eventos onde são passadas listas de presença entre muitos alunos e onde não se têm o controle e garantia da assinatura que consta na lista.

Os professores experienciam no seu cotidiano a custosa tarefa ao realizarem a chamada presencial em sala de aula e em eventos, e esta exige com que seja utilizado um sistema complexo que tomam tempo de aula ou deve ser passada uma lista de presença onde o aluno deve assinar para comprovar sua presença, no entanto, em ambas as formas podem ocorrer equívocos ou até mesmo fraudes. Podemos então imaginar o quanto a automatização da tarefa seria sonhada já que vivemos em uma época com predominância da tecnologia onde a realização de chamada presencial com um papel e caneta pode ser considerada até mesmo um método arcaico para as próximas gerações. Com isto trazemos a inovação tecnológica para tarefas simples do nosso cotidiano, para uma universidade que leva em seu nome a tecnologia.

É notável o alto nível de aceitação com professores ao discutirmos a respeito da implementação de uma ferramenta com essas características e funcionalidades, trazendo maior motivação para realização do projeto proposto, buscando realizar um projeto onde trará comodidade não apenas à uma pessoa, mas sim para uma comunidade inteira, que é a comunidade acadêmica da UTFPR.

Referências

ALVES, P. **Os navegadores de Internet mais usados no Brasil e no mundo**. TechTudo, 2016. Disponível em: <https://www.techtudo.com.br/listas/noticia/2016/01/os-navegadores-de-internet-mais-usados-no-brasil-e-no-mundo.html>. Citado na página 10.

BELHUMEUR, P. N.; HESPANHA, J. P.; KRIEGMAN, D. J. Eigenfaces vs. fisherfaces: recognition using class specific linear projection. IEEE Transactions on Pattern Analysis and Machine Intelligence, v. 19, 1997. Citado 2 vezes nas páginas 8 e 9.

BISSI, T. D. Reconhecimento facial com os algoritmos eigenfaces e fisherfaces: proposta metodológica. p. 12, 2018. Citado na página 5.

CASTRO, E.; HYSLOP, B. **HTML5 and CSS3, Seventh Edition: Visual QuickStart Guide**. 7. ed. [S.l.]: Peachpit Press, 2012. Citado na página 10.

CLOUDINARY. 2019. Disponível em: <https://cloudinary.com/>. Acesso em: 28 de Junho de 2019. Citado na página 17.

CREDDEFENSE. 2018. Disponível em: <https://creddefense.com.br/>. Citado na página 6.

CULJAK, I. et al. A brief introduction to opencv. IEEE, 2012. Citado na página 10.

DAVISON, A. **Vision-based User Interface Programming in Java**. [S.l.]: Amazon Digital Services, 2013. Citado na página 11.

EFRAIM. Introdução ao spring framework. DevMedia, 2012. Disponível em: <https://www.devmedia.com.br/introducao-ao-spring-framework/26212>. Acesso em: 28 de Junho de 2019. Citado na página 12.

ELOLA, J. **O reconhecimento facial abre caminho para o pesadelo de George Orwell**. 2018. Disponível em: <https://brasil.elpais.com/brasil/2018/01/05/tecnologia/1515156123_044505.html>. Acesso em: 11 de Setembro de 2018. Citado 2 vezes nas páginas 5 e 11.

FACE++. 2018. Disponível em: <https://www.faceplusplus.com/face-detection/>. Citado na página 6.

FINDFACE. 2018. Disponível em: <https://findface.pro/en/>. Citado na página 5.

HAVERBEK, M. **Eloquent JavaScript**: A mordern introduction to programming. 3. ed. [S.l.: s.n.], 2018. Citado na página 10.

JOHNSON, R. **Expert One-on-one J2ee Design and Development**. [S.l.]: Wiley Publishing, 2003. Citado na página 11.

LEANDRO, E. **Aplicativo FindFace permite que estranhos identifiquem você**. 2019. Disponível em: <https://www.areah.com.br/vibe/aplicativo/materia/171535/1/pagina_1/aplicativo-findface-permite-que-estranhos-identifiquem-voce.aspx>. Acesso em: 28 de Junho de 2019. Citado na página 5.

OJALA, T.; PIETIKäINEN, M.; HARWOOD, D. **A comparative study of texture measures with classification based on featured distributions**. [S.l.]: Elsevier, 1996. v. 29. Citado na página 9.

PARRA, D. E. Parra. comparação entre algoritmos de reconhecimento de face no contexto de acessibilidade. p. 56–58, 2014. Citado na página 9.

PIRES, J. **O que é API? REST e RESTful? Conheça as definições e diferenças!** 2017. Disponível em: <https://becode.com.br/o-que-e-api-rest-e-restful/>. Acesso em: 27 de Junho de 2019. Citado na página 11.

RICHARDSON, L. **RESTful Web APIs**. [S.l.]: Reilly Media, 2013. Citado na página 11.

SULIVAN, F. . **Sistema de Reconhecimento Facial para Prevenção de Fraudes de Crédito**. 2018. Disponível em: <https://br.nec.com/pt_BR/safety/pdf/finance_infographics.pdf>. Acesso em: 11 de Setembro de 2018. Citado na página 6.

SWAGGER. 2019. Disponível em: <https://swagger.io/>. Acesso em: 28 de Junho de 2019. Citado 2 vezes nas páginas 16 e 17.

TÓV-TEC. **A TÓV-TEC É UMA EMPRESA DE TECNOLOGIA QUE REALIZA O DESENVOLVIMENTO E PERSONALIZAÇÃO DE SOFTWARES**. 2017. Disponível em: <http://feirasenegocios.com.br/noticias/2017/06/tov-tec/>. Acesso em: 11 de Setembro de 2018. Citado na página 7.

TóV-TEC. 2018. Disponível em: <http://tovtec.com.br/>. Citado na página 7.

WANG, L.; HE, D.-C. **Texture classification using texture spectrum**. [S.l.]: Elsevier, 1990. v. 23. Citado na página 9.

WANG, X.; YAN, S.; HAN, T. X. An hog-lbp human detector with partial occlusion handling. **Computer Vision - IEEE 12th International Conference**, p. 32–39, 2009. Citado na página 9.

Z., S.; LI, A. K. **Handbook of Face Recognition**. [S.l.]: London: Springer, 2011. Citado 2 vezes nas páginas 7 e 9.