

Sistema CIAPD: Auxílio a Deficientes para o Mercado de Trabalho

Bruno Pereira Bannwart¹; Fernando Ernesto Kintschner²;
Denise Helena Lombardo Ferreira³

1. Graduado em Engenharia de Computação,
bbannwart.bb@gmail.com; 2. Doutor em Engenharia
Mecânica, Professor Extensionista, [fek@puc-](mailto:fek@puc-campinas.edu.br)

campinas.edu.br; 3. Doutora em Educação Matemática,
Professora Pesquisadora e Membro do Mestrado em
Sustentabilidade, lombardo@puc-campinas.edu.br
Pontifícia Universidade Católica de Campinas,
Campinas, Brasil

Resumo—Torna-se cada vez mais necessário possibilitar oportunidades para que as pessoas que apresentam algum tipo de deficiência ingressem no mercado de trabalho. O Projeto Preparando Pessoas com Deficiência para a Inclusão no Mundo do Trabalho de uma Universidade do interior do estado de São Paulo demonstra que a Instituição busca gerir a diversidade considerando o caminho do compromisso social. As oficinas desenvolvidas demonstraram a preocupação em atender a regulação legal orientada por princípios da inserção social. A pesquisa tem como objetivo apresentar um sistema web (um agregado de três subsistemas) para a abordagem de empregabilidade, fornecendo virtualmente os métodos de aprendizagem disponíveis, além de atuar como ligação entre os alunos deficientes e as empresas parceiras. A solução proposta pode representar uma importante ferramenta para auxiliar as pessoas portadoras a obterem melhores oportunidades de emprego, provendo os métodos de aprendizagem disponíveis para o maior público possível, por meio da acessibilidade. Como consequência, a solução proposta pode possibilitar um crescimento no desenvolvimento, principalmente intelectual, de cada aluno portador de deficiência e, portanto, favorecer a autonomia.

Palavras-chave—deficiência; sistema de informação; mercado de trabalho.

I. INTRODUÇÃO

O termo “deficiência” não é desconhecido para a maioria das pessoas, dado que existe um grande número de deficientes no mundo. Segundo o decreto 3.298/99, um indivíduo é considerado portador de alguma deficiência ao se enquadrar em qualquer uma das seguintes categorias: deficiência física, deficiência auditiva, deficiência visual, deficiência mental ou deficiência múltipla [1].

Conforme o último censo do Instituto Brasileiro de Geografia e Estatística (IBGE), realizado em 2010, o Brasil compreende em seu território mais de 45 milhões de pessoas com deficiência (PCD), representando em torno de 24% (praticamente ¼) da população brasileira e que, no entanto, totalizam apenas 0,9% das carteiras assinadas no mercado de trabalho [2].

Tendo em mente as dificuldades enfrentadas diariamente por PCDs, o Centro Interdisciplinar de Atenção à Pessoa com Deficiência (CIAPD) foi concebido em 1991 por uma

Universidade do interior do Estado de São Paulo com o propósito de promover uma melhor inclusão social para deficientes com idade superior a 16 anos para atuar na Região Metropolitana de Campinas. O trabalho exercido no CIAPD objetiva evidenciar para a sociedade que as PCDs devem ser reconhecidos pelos seus respectivos potenciais, independentemente de suas situações.

Uma das abordagens adotadas pelo CIAPD para alcançar seu propósito é o ingresso de PCD no mercado de trabalho, contribuindo diretamente para o crescimento da taxa apresentada por [2]. Esta abordagem é realizada através de métodos de aprendizagem (como oficinas interdisciplinares e jogos educacionais), com o intuito de enriquecer o currículo e desenvolver intelectualmente PCDs, com a finalidade de obter melhores oportunidades no mercado de trabalho.

Esta pesquisa tem como objetivo apresentar um sistema web (um agregado de três subsistemas) para a abordagem de empregabilidade do público-alvo do CIAPD, fornecendo virtualmente os métodos de aprendizagem disponíveis, além de atuar como ligação entre os alunos deficientes e as empresas parceiras. Deste modo, as PCDs podem ter seus currículos e desenvolvimentos intelectuais aprimorados a qualquer instante e em qualquer localidade, como também ter contato com as vagas de emprego ofertadas pelas empresas. Ressalta-se que o sistema serve para qualquer deficiência, sobretudo a deficiência visual e intelectual, portanto, possui integrados recursos, tais como leitor de tela, reconhecimento facial e ajuda por voz para prover o máximo de acessibilidade possível.

II. MATERIAL E MÉTODOS

As principais tecnologias arquitetadas por terceiros empregadas para os recursos de reconhecimento facial e de ajuda por voz foram: *OpenCV*, *Flask* e *Google Text-To-Speech*.

A. *OpenCV*

Para o desenvolvimento do sistema foi utilizado o *OpenCV* para implementar o recurso de reconhecimento facial. Trata-se de uma biblioteca aberta para desenvolvimento de aplicações nas áreas de *computer vision* e *machine learning*. Esta biblioteca contém uma vasta coleção de algoritmos

disponíveis para diferentes aplicações, incluindo detecção e reconhecimento de faces [3].

Para a detecção facial optou-se pelo algoritmo *HaarCascade* (oficialmente nomeado como algoritmo Viola-Jones) presente na biblioteca *OpenCV*, com o intento de encontrar em uma determinada imagem as faces (localização e tamanho) para serem manuseadas no processo de treinamento e de reconhecimento. Este algoritmo utiliza máscaras (chamadas *Haar Features*) para caracterizar uma face por meio de variações de luminosidade, por exemplo, a região dos olhos ser mais escura comparada com a região do nariz.

Para o reconhecimento facial optou-se pelo algoritmo LBPH (*Local Binary Patterns Histograms*). Tanto o processo de treinamento quanto o de reconhecimento de uma face compartilham as etapas retratadas na Fig. 1.

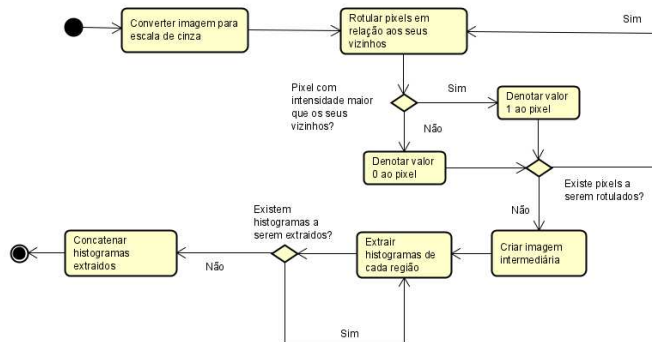


Fig. 1. Diagrama de atividade das etapas realizadas pelo LBPH

Conforme a documentação disponível da biblioteca, o LBPH é um detector de textura que rotula os pixels da imagem convertida previamente para uma escala de cinza, ao comparar cada pixel com seus *neighbors*, ou vizinhos. Cada pixel é analisado como se fosse o centro (rodeado por um limite de vizinhos) e verifica se este possui uma intensidade variando de 0 a 255, maior ou igual em relação aos seus vizinhos. Se o resultado da verificação for verdadeiro, é denotado o valor binário 1 ao pixel analisado. Caso contrário, o valor binário denotado é 0. Esta etapa é realizada com a finalidade de criar uma imagem intermediária, destacando as características faciais.

A próxima etapa executada pelo algoritmo é a extração de histogramas que representam a intensidade de cada pixel da imagem intermediária criada, dividindo-a em múltiplas regiões, ou grades. Logo após ter extraído todos os histogramas de cada grade, o algoritmo realiza o processo de concatenação para formar um histograma que representa as características originais da imagem fornecida.

Este novo histograma é utilizado tanto para o processo de treinamento quanto para a etapa de reconhecimento. No primeiro caso, cada histograma concatenado é armazenado com o seu respectivo identificador (ID ou nome) em um arquivo de treinamento, criando-o, caso seja a primeira face sendo treinada, ou atualizando-o. No segundo caso, o algoritmo realiza uma predição ao comparar o novo histograma com todos os demais armazenados, durante o

processo de treinamento, retornando o identificador e o grau de confiabilidade do histograma mais semelhante encontrado.

B. Flask

O recurso de reconhecimento facial foi desenvolvido como um micro serviço semelhante a uma API – *Application Programming Interface*, sendo compartilhado entre todos os subsistemas da solução e capacitado para receber requisições HTTP para o processo de treinamento ou para o processo de reconhecimento de uma face. Para tal fim, foi utilizado o *Flask* [4].

O *Flask* é um micro *framework* que provê uma arquitetura simples para o desenvolvimento *web*, porém, com a capacidade de escalar aplicações mais robustas conforme a necessidade do projeto em questão. Esse micro *framework* suporta extensões para adicionar uma funcionalidade à aplicação como se tivesse sido implementada diretamente. Além disso, o *Flask* dispõe de uma ferramenta prática de *debug*, fundamental para aplicar corretamente os algoritmos e tratar possíveis erros.

C. Google Text-To-Speech

Para disponibilizar o recurso de ajuda por voz no sistema, optou-se em conectar com APIs estruturadas por terceiros, no caso, foi utilizado a API *Text-To-Speech* do Google [5]. Esta API tem como serviço fornecido a conversão de textos contabilizando a quantidade de caracteres em áudios, com voz sintética, para ser reproduzida em qualquer tipo de aplicação.

Segundo a sua documentação, a API possui diversas opções disponíveis para customizar esta voz, tais como o volume do áudio, a frequência da fala, o idioma falado e o gênero da voz, conforme os parâmetros passados na requisição HTTP, com a finalidade de sintetizar a voz o mais natural possível, para não causar dificuldades no entendimento, principalmente com o deficiente intelectual.

III. SOLUÇÃO

A solução foi projetada utilizando a linguagem de programação Python [6] e o *framework* Django [7] juntamente com o banco de dados relacional MySQL [8], com o versionamento do sistema efetuado no Github. O gerenciamento do projeto foi feito utilizando o site Trello, seguindo a metodologia de desenvolvimento ágil *Scrum*, com sprints de duas semanas cada [9].

A linguagem de programação Python, o *framework* Django e o banco de dados MySQL foram escolhidos por questão de padronização entre os projetos de extensão da Universidade. Além disso, a linguagem Python é repleta de bibliotecas e documentações on-line, agilizando o processo de implementação de funções existentes e gerando um crescimento na produtividade.

Por sua vez, o Django é um *framework web* desenvolvido com a linguagem de programação Python que segue o padrão de projeto *model-template-view*. Assim como o Python, este *framework* possui uma extensa documentação on-line e uma comunidade de desenvolvedores bastante ativa, possibilitando obter rapidamente soluções para diversos problemas encontrados durante o desenvolvimento da solução.

O padrão de projeto do Django é dividido essencialmente em três camadas: *model*, *view* e *template*. A camada *model* é responsável pelo acesso ao banco de dados, realizando a escrita e a leitura; já a camada *view* é responsável por toda a lógica (as regras de negócio) da aplicação e, por fim, a camada *template* é responsável pela visualização dos dados através das páginas HTML com CSS e Javascript.

O processo de desenvolvimento da solução consiste basicamente em duas etapas. A primeira é relacionada com a implementação da estrutura básica do sistema envolvendo os cadastros gerais, a navegação entre páginas, a autenticação do usuário e o armazenamento/compartilhamento de mídias. A segunda etapa relaciona-se com a implementação e/ou integração de funcionalidades, ou recursos, para prover a acessibilidade (treinamento/reconhecimento facial, leitor de tela e ajuda por voz). A Fig. 2 representa o diagrama de arquitetura da solução proposta, exemplificando todas as funcionalidades presentes no sistema.

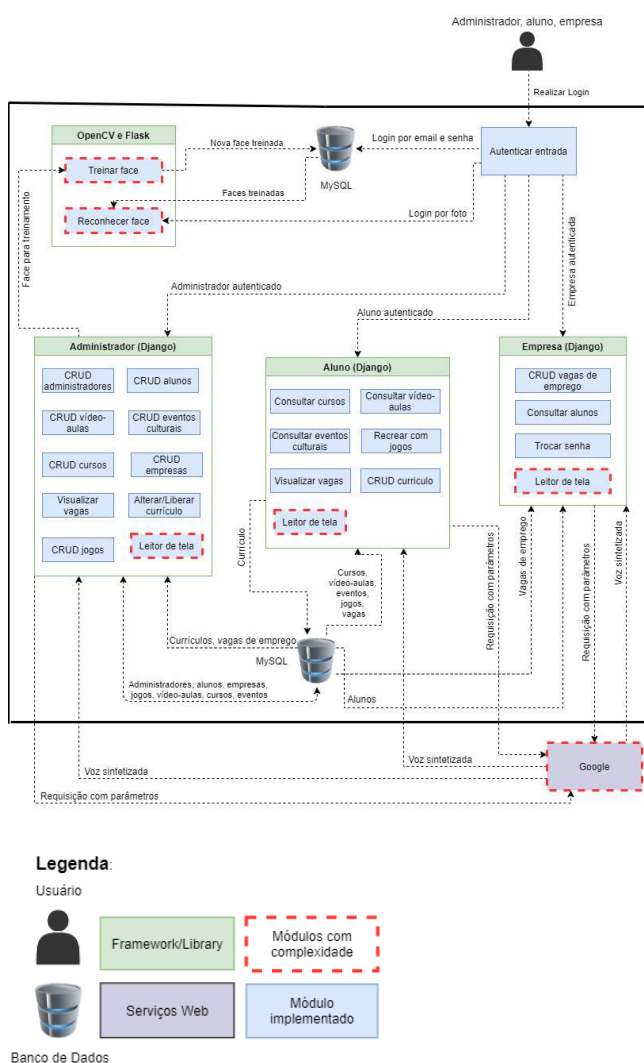


Fig. 2. Diagrama de arquitetura

A. Estrutura básica

A solução proposta consiste em três subsistemas interligados, pertencentes a um macro sistema. Cada subsistema possui funcionalidades conforme o tipo de usuário para o qual foi desenvolvido, sendo eles: o administrador, o aluno e a empresa. Todas as funcionalidades presentes no macro sistema estão exemplificadas na Fig. 3.

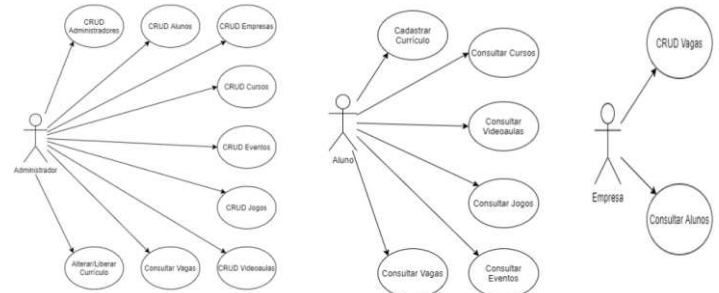


Fig. 3. Diagrama de casos de uso das funcionalidades dos usuários

As interfaces *web* foram projetadas conforme o respectivo usuário, visando proporcionar praticidade, melhor experiência de usuário e acessibilidade, especialmente para os alunos portadores de deficiência.

A figura do administrador, atribuída a um funcionário do CIAPD, é responsável pelos cadastros gerais do projeto, como outros usuários (administradores, alunos e empresas) e também os métodos de aprendizagem (como cursos disponíveis, eventos culturais, jogos educacionais e videoaulas) para que os alunos possam usufruí-los. Também é papel do administrador avaliar as informações presentes nos currículos criados pelos alunos. Caso as informações contidas estejam corretas/coerentes, o currículo é aprovado/liberado e as informações presentes são adicionadas ao perfil do aluno.

A figura do aluno, atribuída a um aluno portador de deficiência, usufruirá dos métodos de aprendizagem com o intento de agregar novos conhecimentos à sua qualificação e, consequentemente, melhorar o seu currículo. Um exemplo é o caso de um deficiente intelectual que pode aprender qual é a melhor vestimenta para utilizar em uma entrevista de emprego através de um dos jogos educacionais disponíveis. O currículo de um aluno é constituído pelas instituições de ensino frequentadas, cursos extras realizados, empresas trabalhadas e cargos exercidos, como também o laudo médico. O aluno também tem acesso às vagas de emprego ofertadas pelas empresas, podendo entrar em contato com o responsável de determinada vaga sem depender da ajuda de terceiros.

A figura da empresa, atribuída a uma empresa parceira, é responsável por cadastrar suas vagas de emprego com todas as informações necessárias, por exemplo, o telefone de contato. A empresa também tem acesso aos perfis dos alunos existentes que contêm suas informações básicas, incluindo quais são suas deficiências, agregadas com as informações de seu currículo, que foram aprovadas por algum usuário administrador. Assim, a empresa poderá entrar em contato com algum aluno que atenda os pré-requisitos de suas vagas.

B. Reconhecimento facial

O recurso de reconhecimento facial foi implementado como sendo um micro serviço, aguardando as requisições HTTP em rotas definidas durante o desenvolvimento. Ao realizar o cadastro de um novo usuário é executada uma requisição a esse micro serviço, enviando tanto a imagem contendo essa nova face quanto a qual grupo é pertencente (administrador, aluno ou empresa), para a rota responsável pelo treinamento facial.

Esta rota efetua verificações antes de inicializar o treinamento da nova face, tais como: conferir a existência da imagem na requisição e averiguar se o formato do arquivo recebido é válido. Se o resultado de alguma destas verificações for negativo, é retornada uma solicitação de uma nova requisição. Em seguida, com o uso do algoritmo *Haar Cascade*, detecta-se o rosto presente na imagem recebida. Caso nenhum rosto seja detectado, também é retornado um erro. Por fim, com o uso do algoritmo LBPH é realizado o processo de treinamento da nova face e, ao término deste, é retornado o código ID para ser incorporado com as outras informações cadastrais do novo usuário. O funcionamento de ambos os algoritmos está descrito na seção II. A.

Por sua vez, o processo de reconhecimento facial também possui uma rota específica. Ao realizar uma tentativa de login por foto é executada outra requisição a esse micro serviço, enviando uma foto obtida pelo acesso à câmera existente no computador, como também em qual dos subsistemas o usuário está pretendendo conectar. Da mesma maneira que a rota de treinamento, a rota de reconhecimento realiza verificações antes de inicializar o seu respectivo processo. Em seguida, com o uso do algoritmo *Haar Cascade*, detecta-se o rosto presente na foto. Por fim, com o uso do algoritmo do LBPH é realizado o processo de reconhecimento, onde é efetuado uma predição do código ID de treino referente à face mais semelhante. Tendo identificado a face similar, é verificado se esta é compatível com o subsistema que o usuário pretende conectar. Caso o resultado desta verificação seja positivo, é retornado o código de treino para obter as demais informações pertencentes a este usuário.

C. Leitor de Tela

Tendo como referência o funcionamento de outros leitores de tela, foi implementado um específico utilizando a linguagem de programação Javascript para integrar com a solução proposta. O escopo inicial do projeto visava incorporar e acessar remotamente um desses leitores presentes no computador do usuário, contudo, esta abordagem origina vulnerabilidades no quesito segurança, permitindo “ataques” para roubar informações ou instalar programas maliciosos no computador do usuário. Em razão disso, optou-se em desenvolver um leitor de tela próprio.

Leitores de tela são *softwares* desenvolvidos para informar por meio sonoro os elementos presentes na tela de um computador ou de um *browser*. Estes *softwares* percorrem textos e imagens, “lendo” em voz alta os elementos que são encontrados na tela, como também as teclas alfanuméricas e comandos digitados pelo usuário, normalmente sendo um

deficiente visual [10]. Tratando-se do leitor implementado da solução, a navegação é efetuada de dois modos: pressionando a tecla TAB para avançar ao próximo elemento, ou pressionando as teclas CTRL + SETAS, na qual a seta da esquerda retorna para o anterior e a seta da direita avança para o próximo.

D. Ajuda por voz

O recurso de ajuda por voz é disponibilizado através do uso da API *Text-To-Speech* do Google, com o seu funcionamento descrito na seção II. C. As requisições são efetuadas conforme o periférico de entrada (*mouse*) e é posicionado em cima dos campos dos formulários.

Desse modo, a linguagem de programação Javascript, responsável por tratar os eventos originados nas páginas HTML, realiza as requisições para esta API, passando todos os parâmetros e credenciais de autenticação necessárias, como também o texto a ser convertido em voz sintética. Ao receber a resposta da requisição efetuada, o Javascript reproduz o áudio gerado pelo serviço.

IV. DISCUSSÃO DOS RESULTADOS

As avaliações funcionais do funcionamento do sistema foram feitas a partir dos recursos provedores de acessibilidade – no caso, o reconhecimento facial.

A. Avaliações funcionais

A avaliação funcional empregada procura verificar a performance do algoritmo de reconhecimento facial da solução. A performance foi definida pelos seguintes indicadores: Acurácia e Precisão [11]. Deste modo, é possível validar corretamente o funcionamento do algoritmo. Para efetuar o cálculo dos indicadores foram definidas as Eqs. (1) e (2).

$$Acurácia = \frac{Acertos}{Total} \quad (1)$$

$$Precisão = \frac{Verdadeiros\ Positivos}{Verdadeiros\ Positivos + Falsos\ Positivos} \quad (2)$$

A acurácia (Eq. (1)) é um indicador que calcula a quantidade de acertos pelo total de execuções (tentativas), contudo, não pode ser analisada isoladamente devido à ocorrência de resultados falsos positivos. A fim de obter uma análise mais confiável da performance do algoritmo fez-se também o uso do cálculo da precisão (Eq. (2)). A precisão é um indicador que demonstra o quão correto o algoritmo está em relação às suas predições, analisando tanto os falsos positivos quanto os verdadeiros positivos. Por resultado “verdadeiro positivo” é compreendido que foi efetuado o reconhecimento correto da face do usuário e, por resultado “falso negativo”, o reconhecimento da face foi incorreto.

Para a análise das métricas, em um total de 32 execuções (tentativas) de reconhecimento das faces, 24 foram assertivas, isto é, 24 acertos (19 verdadeiros positivos; no total o sistema detectou corretamente cinco em que não havia face presente

na imagem, seis não detectou corretamente a face existente e houve dois falsos positivos). Para o cálculo do valor da acurácia, obteve-se 0,75 (24/32) e para o cálculo do valor da precisão, obteve-se 0,9 (19/(19+2)).

V. CONCLUSÃO

A solução proposta pode representar uma importante ferramenta para auxiliar as pessoas portadoras de deficiência a obterem melhores oportunidades de emprego, provendo os métodos de aprendizagem disponíveis para o maior público possível, por meio da acessibilidade. Como consequência, a solução proposta pode possibilitar um crescimento no desenvolvimento, sobretudo intelectual, de cada aluno portador de deficiência e, portanto, favorecer a autonomia.

Contudo, essa solução possui algumas limitações, podendo receber melhorias em trabalhos futuros para obter um resultado ainda mais satisfatório com o público-alvo, por exemplo:

- Reconhecimento facial. Em razão da limitação no escopo do projeto, o *dataset* para realizar o treinamento se resume em apenas uma única imagem para cada novo usuário. A fim de obter resultados ainda mais precisos, seria necessário ampliar a quantidade de imagens durante o cadastro de um novo usuário.
- Leitor de tela. O leitor de tela é uma versão simples, porém efetiva, comparada com os leitores de tela disponíveis no mercado. Seria pertinente buscar um modo de agregar outros leitores.
- Aplicativo mobile. Tendo em mente que nem todas as pessoas com deficiência possuem acesso a um computador, seria interessante uma versão alternativa

do sistema como um aplicativo mobile, visto que o celular é bem mais utilizado nos dias atuais e é mais acessível quando comparado a um computador. Dessa forma, um aplicativo mobile poderia abranger um público maior.

REFERÊNCIAS

- [1] Brasil. Decreto nº 3.298, de 20 de dezembro de 1999. Disponível em: <http://www.planalto.gov.br/ccivil_03/decreto/d3298.htm>. Acesso em: 01 mar. 2021.
- [2] IBGE - Instituto Brasileiro de Geografia e Estatística-Censo demográfico 2010. Características gerais da população, religião e pessoas com deficiência. Disponível em: <<http://www.ibge.org.br>>. Acesso em: 03 mar. 2021.
- [3] OpenCV. Open Source Computer Vision Library. Disponível em: <<https://opencv.org>>. Acesso em: 10 mar. 2021.
- [4] Flask. Welcome to Flask. Disponível em: <<https://flask.palletsprojects.com/en/1.1.x>>. Acesso em: 10 mar. 2021.
- [5] Google. Text-To-Speech. Disponível em: <<https://cloud.google.com/text-to-speech>>. Acesso em: 25 mar. 2021.
- [6] Python. Disponível em: <<https://www.python.org>>. Acesso em: 25 mar. 2021.
- [7] Django. Disponível em: <<https://www.djangoproject.com>>. Acesso em: 29 de mar. 2021.
- [8] MySQL. Disponível em: <<https://www.mysql.com>>. Acesso em: 29 mar. 2021.
- [9] Scrum. Disponível em: <<https://www.scrumalliance.org>>. Acesso em: 06 abr. 2021.
- [10] Teixeira, F. Acessibilidade: Como Funcionam os Leitores de Tela, 2015. Disponível em: <<https://brasil.uxdesign.cc/acessibilidade-como-funcionam-os-leitores-de-tela-3d9b610216e1>>. Acesso em: 06 de abr. 2021.
- [11] Powers, D. Evaluation: From Precision, Recall and F-Measure to ROC, 2011. Informedness, Markedness & Correlation. Journal of Machine Learning Technologies.