



UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS RUSSAS
CURSO DE GRADUAÇÃO EM ENGENHARIA DE SOFTWARE

ARTHUR CÉSAR FERNANDES SILVA NOGUEIRA

**UMA INTERFACE COM FOCO EM USABILIDADE PARA A RELAÇÃO ENTRE
HOMEM E INTELIGÊNCIA ARTIFICIAL PARA UM SISTEMA DE AUTOMAÇÃO
INDUSTRIAL: DESENVOLVIMENTO DE UMA SOLUÇÃO TECNOLÓGICA COM
FOCO NA USABILIDADE PARA A RELAÇÃO ENTRE USUÁRIO E INTELIGÊNCIA
ARTIFICIAL PARA UM SISTEMA DE AUTOMAÇÃO INDUSTRIAL**

RUSSAS

2023

ARTHUR CÉSAR FERNANDES SILVA NOGUEIRA

UMA INTERFACE COM FOCO EM USABILIDADE PARA A RELAÇÃO ENTRE HOMEM
E INTELIGÊNCIA ARTIFICIAL PARA UM SISTEMA DE AUTOMAÇÃO INDUSTRIAL:
DESENVOLVIMENTO DE UMA SOLUÇÃO TECNOLÓGICA COM FOCO NA
USABILIDADE PARA A RELAÇÃO ENTRE USUÁRIO E INTELIGÊNCIA ARTIFICIAL
PARA UM SISTEMA DE AUTOMAÇÃO INDUSTRIAL

Trabalho de Conclusão de Curso apresentado ao
Curso de Graduação em Engenharia de Software
do Campus Russas da Universidade Federal do
Ceará, como requisito parcial à obtenção do
grau de bacharel em Engenharia de Software.

Orientadora: Profa. Dra. Rosineide Fer-
nando da Paz

RUSSAS

2023

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

- N71i Nogueira, Arthur César Fernandes Silva.
Uma interface com foco em usabilidade para a relação entre homem e inteligência artificial para um sistema de automação industrial : desenvolvimento de uma solução tecnológica com foco na usabilidade para a relação entre usuário e inteligência artificial para um sistema de automação industrial / Arthur César Fernandes Silva Nogueira. – 2023.
60 f. : il. color.
- Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Russas, Curso de Engenharia de Software, Russas, 2023.
Orientação: Profa. Dra. Rosineide Fernando da Paz.
1. usabilidade. 2. visão computacional;. 3. plataforma facilitadora. 4. usabilidade na web. I. Título.
CDD 005.1
-

ARTHUR CÉSAR FERNANDES SILVA NOGUEIRA

UMA INTERFACE COM FOCO EM USABILIDADE PARA A RELAÇÃO ENTRE HOMEM
E INTELIGÊNCIA ARTIFICIAL PARA UM SISTEMA DE AUTOMAÇÃO INDUSTRIAL:
DESENVOLVIMENTO DE UMA SOLUÇÃO TECNOLÓGICA COM FOCO NA
USABILIDADE PARA A RELAÇÃO ENTRE USUÁRIO E INTELIGÊNCIA ARTIFICIAL
PARA UM SISTEMA DE AUTOMAÇÃO INDUSTRIAL

Trabalho de Conclusão de Curso apresentado ao
Curso de Graduação em Engenharia de Software
do Campus Russas da Universidade Federal do
Ceará, como requisito parcial à obtenção do
grau de bacharel em Engenharia de Software.

Aprovada em: 14/07/2023

BANCA EXAMINADORA

Profa. Dra. Rosineide Fernando da Paz (Orientadora)
Universidade Federal do Ceará (UFC)

Profa. Dra. Anna Beatriz dos Santos Marques
Universidade Federal do Ceará (UFC)

Prof. Dr. Pablo Luiz Braga Soares
Universidade Federal do Ceará (UFC)

AGRADECIMENTOS

Primeiramente a Deus, pois sem ele nada disso seria possível.

A Profa. Dra. Rosineide Fernando da Paz por ser minha orientadora e mentora por, praticamente, todo o período em que fiz parte do Laboratório de Tecnologias Inovadoras(LTI).

Aos outros Professores do LTI, Prof. Dr. Dmontier Pinheiro , Prof. Dr. Alexandre Arruda e Prof. Ms. Filipe Maciel, por todos os ensinamentos que me repassaram ao longo a minha permanência nos projetos do laboratório.

A todos os integrantes do Projeto *Medusa's Eyes*, em especial Davi Monteiro, aluno de graduação em Ciência da Computação.

Aos meus pais, familiares, amigos e Vanessa Nunes que sempre me apoiaram diante de todas as adversidades enfrentadas longe de casa.

A Matheus Barbosa, Orlando Rodrigues, Fabricio Juliano, Airlon Filho e Moab Souza por se tornarem minha família e por compartilharem comigo todos os momentos felizes e tristes durante o período em que estive longe de casa.

A Ronaldo Pessoa que sempre me amparou durante a minha estadia na cidade de Russas-CE.

RESUMO

A busca pela otimização dos processos de produção sempre esteve presente na história da indústria. As consequências dessa incessante procura trouxeram inúmeras inovações nos campos científicos e tecnológicos. Ao longo do tempo esta perseguição por maiores e melhores resultados nunca diminuiu, pelo contrário, sempre esteve maior a cada revolução industrial. Hoje, na Indústria 4.0, a presença de tecnologias avançadas advindas do universo científico, potencializou ainda mais os ganhos para o ramo industrial. Porém, encontrar um meio termo entre maior tecnologia e facilidade de uso é um dos desafios encontrados, quando se fala de maquinário moderno e de alta performance. Um dos cenários encontrados no contexto industrial é a interação entre homem e máquina. Logo, é possível observar que esta parceria precisa ser consistente e de fácil entendimento. Designadamente, quando se trata da utilização de algoritmos de visão computacional para reconhecimento de objetos em imagens, é exigido do usuário um grande conhecimento prévio em informática e/ou linguagens de programação. Com isso, o presente trabalho propõe uma interface para um sistema que deverá facilitar o treinamento de uma rede neural convolucional na detecção de cristais em imagens microscópicas de clínquer, que é um produto utilizado na produção do cimento Portland, com o foco na usabilidade. Assim, com o uso da ferramenta proposta, uma pessoa sem muitas habilidades em linguagem de programação poderá treinar e utilizar um algoritmo para detecção de objetos, sem grandes dificuldades e de forma intuitiva. Para isso, foram estudadas metodologias e ferramentas que facilitam o desenvolvimento do sistema proposto, além de potencializar os resultados esperados.

Palavras-chave: usabilidade; visão computacional; plataforma facilitadora; usabilidade na Web.

ABSTRACT

The search for the optimization of production processes has always been present in the industry's history. The consequences of this incessant search have brought numerous innovations in the scientific and technological fields. Throughout time, this search for bigger and better results has never diminished, on the contrary, it has always been greater with each industrial revolution. Today, in Industry 4.0, the presence of advanced technologies coming from the scientific universe has potentiated even more the gains for the industrial branch. But finding a middle ground between greater technology and ease of use is one of the challenges when it comes to modern, high-performance machines. One of the scenarios found in the industrial context is the interaction between man and machine, which is increasingly present. So you can see that this partnership needs to be consistent and easy to understand. Particularly when it comes to using computer vision algorithms to recognize objects in images, this requires some prior knowledge in computer science and/or programming languages. With this, the present proposes a user-friendly interface for a system that should facilitate the training of a convolutional neural network in the detection of crystals in microscopic images of clinker, which is a product used in the production of Portland cement. Thus, with the use of the proposed proposed tool, a person without much skills in programming language will be able to train and use an algorithm for object detection, without great difficulties and in an intuitive way. For this, methodologies and tools that facilitate the development of the proposed the development of the proposed system, besides potentiating the expected results. With this, the present work aims to show that it is possible that people with little or no computational knowledge are able to use an on-line platform destined to the use of these algorithms. That is, this person trains and runs an Artificial Intelligence, without great difficulty and in an intuitive way. To this end, methodologies and tools were studied to facilitate the development of the proposed system, and to enhance the expected results.

Keywords: usability; computer vision; facilitating platform; web usability.

LISTA DE FIGURAS

Figura 1 – Cristal de Alita	23
Figura 2 – Esboço da tela de projetos do sistema	25
Figura 3 – Protótipo a tela de projetos do sistema	26
Figura 4 – Arquitetura do Sistema	27
Figura 5 – Fluxograma de funcionamento do sistema	29
Figura 6 – Scrum	32
Figura 7 – Diagrama com um exemplo de funcionamento do Git.	32
Figura 8 – Saída do algoritmo com a máscara em preto e branco.	34
Figura 9 – Saída do algoritmo com a máscara com suas cores originais.	35
Figura 10 – Campos de upload de classes e <i>JavaScript Object Notation</i> (JSON).	36
Figura 11 – Campo de <i>upload</i> de imagens e a lista de imagens	37
Figura 12 – Tela de segmentação.	38
Figura 13 – Cartão com listagem de máscaras.	39
Figura 14 – Cartão com listagem de classes	39
Figura 15 – Cartão com listagem de classes.	39
Figura 16 – Cartão de escolha de classe.	40
Figura 17 – Cartão de <i>status</i> sem máscara selecionada.	40
Figura 18 – Cartão de <i>status</i> com máscara selecionada	41
Figura 19 – Imagem com duas máscaras segmentadas.	42
Figura 20 – Imagem com uma das máscaras selecionada	42
Figura 21 – Imagem sem máscaras segmentadas.	43
Figura 22 – Lista de imagens.	44
Figura 23 – Página de Login.	46
Figura 24 – Página de cadastro.	47
Figura 25 – Página de cadastro com indicação de erro no campo de e-mail.	47
Figura 26 – Página de cadastro com indicação de erro no campo de confirmação de senha.	48
Figura 27 – Máscaras na interface.	53
Figura 28 – Máscaras com as coordenadas retornadas pelo algoritmo.	53
Figura 29 – Máscara inserida na Ferramenta 1.	54
Figura 30 – Máscara com as coordenadas retornadas pelo algoritmo.	55

LISTA DE TABELAS

Tabela 1 – Tabela comparativa entre os trabalhos relacionados	21
---	----

LISTA DE ABREVIATURAS E SIGLAS

JSON	<i>JavaScript Object Notation</i>
API	<i>Application Programming Interface</i>
AWS	<i>Amazon Web Services</i>
HTTP	<i>Hypertext Transfer Protocol</i>
XML	<i>Extensible Markup Language</i>
COCO	<i>Common Objects in Context</i>

SUMÁRIO

1	INTRODUÇÃO	12
2	FUNDAMENTAÇÃO TEÓRICA	14
2.1	Revoluções industriais	14
2.2	Indústria 4.0	15
2.2.1	<i>Inteligência Artificial</i>	15
2.2.1.1	<i>Aprendizado Profundo</i>	16
2.2.2	<i>Big Data</i>	17
2.2.3	<i>Computação em Nuvem</i>	17
2.2.4	<i>Internet das Coisas</i>	18
2.3	Homem-máquina na indústria 4.0	19
2.4	Objetivos	19
2.4.1	<i>Objetivo Geral</i>	20
2.4.2	<i>Objetivos Específicos</i>	20
3	TRABALHOS RELACIONADOS	21
3.1	Trabalho 1 - Arquitetura de sistema em nuvem para apoio à implantação de Visão Computacional em linhas de produção na Indústria 4.0	21
3.2	Trabalho 2 - ODIN - Uma Ferramenta Visual para o Desenvolvimento de Modelos de Detecção de Objetos com <i>Deep Learning</i> no Ensino Superior	21
4	METODOLOGIA	22
4.1	Aplicação para o sistema amigável	22
4.1.1	<i>Microscopia do Clínquer</i>	22
4.2	Usabilidade na aplicação Web	24
4.3	Ciclo de concepção de interfaces web	24
4.3.1	<i>Engenharia de Requisitos</i>	24
4.3.2	<i>Especificação do Site</i>	25
4.3.3	<i>Design do Site</i>	26
4.3.4	<i>Implementação do site</i>	26
4.3.5	<i>Esboço e Protótipo</i>	26
4.4	Heurísticas de Nielsen	27
4.5	Arquitetura do sistema	27

4.6	Fluxograma da interface	29
4.7	Tecnologias	30
4.7.1	<i>Typescript</i>	30
4.7.2	<i>React</i>	31
4.7.3	<i>Ant design</i>	31
4.8	Scrum	31
4.9	Gerenciamento do código	32
5	RESULTADOS	34
5.1	Produto 1 - Ferramenta <i>Medusas Eye's</i>	34
5.1.1	<i>Tela inicial - Upload</i>	35
5.1.2	<i>Tela de segmentação - Segmentation</i>	37
5.1.3	Produto 2 - Sistema completo	45
5.1.4	<i>Pagina Inicial</i>	45
5.1.5	<i>Pagina de cadastro</i>	46
5.2	Heurísticas de Nielsen na Prática	48
5.2.1	<i>Visibilidade do status do sistema</i>	48
5.2.2	<i>Controle e liberdade do usuário</i>	49
5.2.3	<i>Consistência e padrões</i>	49
5.2.4	<i>Prevenção de erros</i>	50
5.2.5	<i>Reconhecimento em vez de recordação</i>	51
5.2.6	<i>Flexibilidade e eficiência de uso</i>	51
5.2.7	<i>Estética e design minimalista</i>	51
5.2.8	<i>Ajuda e documentação</i>	52
5.2.9	<i>Diagnóstico e recuperação de erros</i>	52
5.3	Testes	52
6	CONCLUSÕES E TRABALHOS FUTUROS	56
6.1	Melhorias	56
6.2	Limitações	57
	REFERÊNCIAS	58

1 INTRODUÇÃO

Na história da indústria, a busca pela otimização dos processos produtivos sempre esteve presente. Os frutos dessa constante procura foram inovações nos campos da ciência e da tecnologia. Nessa evolução, foi mais modificando a forma como o ser humano lida com as máquinas, e em como o mesmo esteve em busca de capacitação para poder manusear o maquinário, que estava em constante desenvolvimento. Durante a terceira revolução industrial surgiram as primeiras máquinas eletrônicas, que ajudariam a maximizar os processos produtivos, porém aumentariam o grau de complexidade de manuseio (SILVA, 2017).

Agora, na Indústria 4.0, este cenário se perpetua, pois a presença de novas tecnologias, ferramentas, metodologias e outras novidades que foram surgindo ao longo do tempo, trouxeram melhorias significativas para o meio industrial. Porém, atualmente, é exigido um grau de proficiência e conhecimentos cada vez mais profundos para poder operar os dispositivos existentes (LIMA; PINTO, 2019).

Diante do exposto, facilitar o manuseio de ferramentas computacionais, é necessário para proporcionar um melhor desempenho de ambas as partes envolvidas nesse cenário: Homem e Máquina. Diante disso, Papcun *et al.* (2018) caracterizou três cenários em que pode ser observada a interação entre homem e máquina. São eles: Colaboração Humano-Humano, Colaboração Humano-Máquina e Colaboração Máquina-Máquina. A partir desses cenários é possível concluir que a interação entre Homem-Máquina precisa ser a melhor possível, pois até no contexto de máquina-máquina, em algum momento a presença de um humano pode se fazer necessária.

Particularmente, a respeito de sistemas que necessitam da aplicação de algoritmos de visão computacional, o grau de dificuldade para encontrar um intermédio entre complexidade de uso da ferramenta e nível de conhecimento é propenso a aumentar. Destarte, o presente trabalho é motivado pela realização da automação da microscopia do clínquer Padilha (2022), propondo desenvolver uma interface para um sistema que deverá auxiliar o treinamento de uma rede neural convolucional na detecção de objetos, em imagens de clínquer. Além de focar na usabilidade do sistema, para diminuir o esforço humano que as atuais ferramentas demandam.

Para realizar o desenvolvimento deste trabalho foram elicitados requisitos, que deram origem a modelos/protótipos, atividades que foram advindas de um *backlog*. Ademais, foram estudadas heurísticas e metodologias que devem guiar o desenvolvimento do sistema, com foco em usabilidade. Diante de todo esse contexto as tecnologias foram escolhidas para otimizar os

resultados que foram propostos, tendo em vista o grau de familiaridade do autor com as mesmas.

Este trabalho está organizado da forma a seguir. No Capítulo 2 é apresentada a fundamentação teórica do problema. No Capítulo 3, são mostrados os trabalhos relacionados a este. E por fim, no Capítulo 4, serão apresentadas as metodologias e ferramentas que serão utilizadas, bem como a apresentação do cronograma.

2 FUNDAMENTAÇÃO TEÓRICA

A busca por formas de otimizar processos produtivos fez com que a indústria se tornasse fonte de inovações científicas e tecnológicas. Com isso, ocorreu o processo de modernização da indústria. Tais evoluções refletem na forma como o ser humano se capacita para o uso das tecnologias disponíveis e, também, para atuar no mercado de trabalho. Tendo em vista a importância da implementação de tais inovações, aqui, será dada uma descrição sucinta das revoluções ocorridas na indústria.

2.1 Revoluções industriais

A primeira revolução industrial começou por meados de 1760 até 1840. Esta época foi caracterizada pela substituição dos métodos artesanais pelo uso de máquinas a vapor de água. Uma das grandes vantagens de se usar máquinas ao invés da manufatura, foi que os produtos produzidos passaram a seguir padrões que permitiram que os mesmos fossem produzidos de maneira mais uniforme. Acrescido a tais fatos, pode-se destacar o começo das macros invenções, as quais são diversificadas, não sendo ligadas a movimentos, ciência ou forças de mercado.(LIMA; PINTO, 2019), (SAKURAI; ZUCHI, 2018).

A chegada das máquinas modificou totalmente a forma como o ser humano pensava sobre produção, pois aumentar essa produção era o grande objetivos das industrias da época. Com isso, a qualidade dos produtos foi deixada de lado, fazendo com que as fábricas focassem apenas em obter maiores lucros com o aumento da produção. Diante do novo cenário, a forma como os operadores das máquinas interagem com as mesmas não foi um problema abordado no início dessa revolução. Contudo itens como: manivelas, alavancas e pedais foram os primeiros componentes usados para facilitação do manuseio das máquinas. Vale destacar que Papcun *et al.* (2018) considera esses elementos como interface para melhorar a experiência de uso do maquinário.

No período que vai de 1870 até 1914 ocorreu a segunda revolução industrial. Em que o advento da elétrica possibilitou a criação das máquinas movidas a eletricidade. Além do uso de tais máquinas, esta revolução foi marcada pelas pesquisas científicas que muito ajudaram nas inovações para a época, em áreas como: elétrica, química, biologia, transportes, engenharia de produção, agricultura, dentre outras. Ainda sobre máquinas elétricas, pode-se destacar que o uso de tal maquinário possibilitou avanços econômicos e desenvolvimento de métodos produtivos.

As micro invenções também ganharam espaço nessa janela de tempo, sendo, de certa forma, necessárias para que melhorias fossem implementadas nas macro invenções, aumentando assim, o potencial dessas últimas. Papcun *et al.* (2018) afirma que a segunda revolução industrial marca o início da interação homem-máquina, pois, a utilização do maquinário com eletricidade proporcionou o uso de interruptores e luzes, os quais foram os componentes que caracterizaram os primeiros contatos entre humano e máquina (LIMA; PINTO, 2019).

Em meados de 1970 a terceira revolução industrial teve seu início, trazendo o uso de semicondutores, computadores, automação e robotização de linhas de produção como a sua principal inovação. Um dos marcos dessa revolução foi a substituição da eletromecânica pela eletrônica para ser a base da automação. A presença de máquinas cada vez mais automatizadas e complexas de se operar, segundo Papcun *et al.* (2018), marcou a primeira revolução da interação homem-máquina (SILVA, 2017).

2.2 Indústria 4.0

A indústria 4.0 tem como principais princípios: operação; engenharia; planejamento e controle da produção; logística e análise contínua durante o ciclo de vida de produtos e serviços. Além de tais pilares, é importante criar redes inteligentes para que exista um controle dos módulos de produção de forma autônoma. A partir do conceito de que possa existir uma conexão entre componentes inteligentes de uma indústria, a definição de Internet das Coisas (IOT) é apresentado. Outros conceitos devem ser apresentados: Inteligência Artificial (IA), *Big Data* e Computação em Nuvem (CN) Lima e Pinto (2019), Sakurai e Zuchi (2018), que são brevemente descritos nas seções seguintes.

2.2.1 Inteligência Artificial

A IA é uma área da ciência e tecnologia que desenvolve computadores inteligentes e aplicações para realizar atividades que normalmente precisariam do intelecto humano. É um sistema que pode desempenhar uma variedade de funções que antes de sua existência podiam ser realizadas somente por seres humanos. A inteligência artificial tem uma grande influência nas empresas, tais como fabricação, saúde e cadeias de suprimentos, entre outras. Com os avanços computacionais, sistemas inteligentes foram se tornando-se cada vez mais complexos e capazes de realizar tarefas que nem os seres humanos são capazes. Por essa razão, tais sistemas tem

ampla aplicação. Na indústria, aumentam o desempenho e a eficiência dos processos produtivos (PK, 1984).

A implementação de sistemas inteligentes em máquinas ficou conhecido como Aprendizado de Máquinas (AM). Nesse tipo de aprendizado, uma aplicação recebe um conjunto de tarefas para serem realizadas, e considera-se que a máquina aprendeu com sua experiência se seu desempenho medido nestas tarefas melhora à medida que a mesma obtém mais experiência. Como resultado, a máquina faz julgamentos e previsões com base em informações. Jogos de computador, assistentes pessoais virtuais (como o Google), reconhecimento de padrões, processamento de linguagem natural e mineração de dados são apenas alguns dos domínios onde a aprendizagem de máquinas é utilizado (RAY, 2019).

2.2.1.1 *Aprendizado Profundo*

Do inglês *Deep Learning*, o aprendizado profundo é um tipo de AM que pode gerar resultados excepcionais. Estas abordagens, em geral, são mais indicadas para lidar com grandes conjuntos de dados. Os algoritmos de aprendizagem profunda podem aprender padrões a partir de dados observados, sem a necessidade de engenharia de recursos humanos. Ou seja, muitas vezes elimina a necessidade de uma pessoa acompanhar o processo de aprendizagem, e até mesmo a tomada de decisões com base nos padrões aprendidos pela máquina.

No entanto, em se tratando de problemas específicos e de pouco desenvolvimento, ainda é requerida, muito fortemente, a atuação de profissionais capacitados, tanto na área de tecnologias como em áreas específicas do problema a ser abordado.

Tendo em vista que esses modelos podem aprender representações de recursos a partir dos dados originais, tais como fotos e textos, além de, em muitos casos, superar em eficiência modelos tradicionais de aprendizado de máquina, tem sido depositado, ao longo dos últimos anos, um grande esforço no desenvolvimento de tais algoritmos. Portanto, as aplicações destes modelos são diversas e necessárias, sendo estes usualmente chamados na literatura de "redes neurais multicamadas"(CARBONELL *et al.*, 1983; LIU; LANG, 2019).

Um importante avanço das arquiteturas de redes neurais multicamadas foi a proposição da rede neural convolucional (CNN do inglês *convolutional neural network*). Como uma extensão das arquiteturas propostas por Xu *et al.* (2021) e Waibel *et al.* (1989). LeCun *et al.* (1989) propôs uma CNN para reconhecimento de código postal manuscrito e usou pela primeira vez o termo “convolução”, sendo essa considerada a primeira versão da CNN.

Essa rede foi projetada para receber varias matrizes com dados de entrada e processá-las usando um chamado "operador de convolução"dentro de um "campo local". Devido a convolução, esse tipo de rede representou um avanço considerável na área de visão computacional. Ou seja, em problemas de segmentação e detecção de objetos em vídeos e imagens. Na segmentação de imagens, são adotadas regras para que os *pixels* da imagem sejam divididos em diferentes partes e rotulados.

Por conta da grande importância da área de visão computacional, que segundo Milano e Honorato (2010) é o modo que um computador percebe o mundo ao seu redor, muitas outras arquiteturas foram propostas a fim de otimizar as redes neurais multicamadas (ou ditas profundas, quando existe uma quantidade grande de camadas).

Além disso, foram propostas melhorias na parte de detecção de objetos, com a proposição das redes R-CNN Girshick *et al.* (2014) e *Faster R-CNN* Girshick (2015) e *Mask R-CNN* He *et al.* (2017). No entanto, embora muitos problemas tenham sido minimizados, ainda existem muitos desafios na utilização de uma rede neural convolucional para detecção de objetos.

2.2.2 *Big Data*

Quando o problema é manusear grandes quantidades de dados, faz-se necessária a implementação de uma abordagem diferente daquilo que se tinha como usual na indústria, antes da industria 4.0. Foi a partir dessa necessidade que surgiu o Big Data. Esse conceito é importante, pois ao transformar uma grande quantidade de dados em informações relevantes, as indústrias obtêm uma produção mais eficiente (LIMA; PINTO, 2019).

Big Data, ao contrário dos dados tradicionais, refere-se a grandes coleções de dados em expansão que compreendem uma variedade de formas, incluindo dados estruturados, não estruturados e semiestruturados. Devido a complexidade de se trabalhar com esse tipo de dados, é necessário o uso de tecnologias e algoritmos sofisticados, que sejam capazes de atuar em cenários com essas características (OUSSOUS *et al.*, 2018).

2.2.3 *Computação em Nuvem*

A ideia de poder acessar servidores, computadores, softwares, redes, entre outros, de forma remota, constitui o conceito de Computação em Nuvem. O processamento e obtenção de dados utilizando a internet é de grande importância para indústria, pois evita uma grande estrutura física em suas dependências (LIMA; PINTO, 2019).

A computação em nuvem é um tipo de computação em que recursos virtualizados e dinamicamente escaláveis são entregues como um serviço através da Internet. A nuvem é o hardware e o software em *datacenters* atendem às necessidades do cliente, mais comumente na forma de *datastores* e aplicativos hospedados remotamente. Estas infra-estruturas ajudam as empresas a reduzir gastos, reduzindo a necessidade de equipamentos físicos e permitindo a terceirização de dados e cálculos sob demanda (GOYAL; DADIZADEH, 2009).

Os dados terceirizados são dados que são mantidos num ambiente de nuvem e geridos por outras empresas. Os dados são mantidos de tal forma que são geograficamente independentes, reduzindo o custo de manutenção de necessidades de armazenamento, tais como hardware e software. A principal vantagem dos recursos no local versus os recursos na nuvem é o preço baseado no uso e a rápida disponibilidade de recursos sem preocupação com manutenção (ANIL; THANKA, 2013).

2.2.4 Internet das Coisas

A principal vertente da IOT é poder conectar diversos componentes, para que troquem informações entre si. Trazendo esse conceito para o âmbito industrial. As informações que transitam entre os dispositivos permitem que os processos industriais sejam controlados, além de fornecer uma análise detalhada, possibilitando uma tomada de decisão eficaz e dinâmica. Os dispositivos, ou “coisas”, podem ser sensores, atuadores, celulares, *tablets* e etc. Os quais, como já foi dito, podem interagir e cooperar entre si para atingir objetivos em comum (LIMA; PINTO, 2019).

A IoT pode fornecer às organizações, e aos utilizadores finais, capacidade de desenvolvimento e implementação de aplicações em larga escala, heterogêneas, interoperáveis e sensíveis ao contexto em que está inserida (RAY, 2016).

Ao passo que as revoluções foram trazendo novas tecnologias e metodologias, o homem teve que se capacitar para poder acompanhar o mercado. Diante disso, para a indústria 4.0 não é diferente, pois os profissionais devem possuir conhecimentos multidisciplinares para conseguirem atender as demandas da produção moderna. Logo, entende-se que é indispensável que as ferramentas utilizadas pelos profissionais consigam proporcionar um bom desempenho, além de um fácil entendimento e uso (LIMA; PINTO, 2019).

2.3 Homem-máquina na indústria 4.0

Pode-se notar que a atuação humana, nos processos produtivos, está (passando por uma transformação), muitas vezes perdendo força e até mesmo sentido. É inevitável que esses profissionais precisem estar adquirindo sempre novas competências, para que possam operar máquinas que estão cada vez mais modernas e automatizadas (LIMA; PINTO, 2019).

Em contrapartida, muitos profissionais não serão mais necessários no processo de produção e operação, pois serão substituídos por máquinas ou sistemas computadorizados Kinzel (2017). Por consequência desta automação, Peruzzini *et al.* (2017) cita que existe a possibilidade muito alta de que ocorra um gargalo dentro deste cenário. Onde um único trabalhador desempenha muitos processos, além de controlar diversas máquinas, demandando grandes competências.

À proporção que a complexidade dos sistemas vai aumentando, a qualificação de operadores e trabalhadores deve, também, progredir. O objetivo de estudar formas para melhorar a interação entre homem-máquina, é tentar frear a complexidade para que não exista a necessidade de conhecimentos que vão além da formação dos operadores. Facilitando sua jornada de trabalho e tirando a sobrecarga que foi citada anteriormente.

Segundo Papcun *et al.* (2018) é possível caracterizar três cenários, onde existe a interação entre homem e máquina. São eles: Colaboração Humano-Humano, Colaboração Humano-Máquina e Colaboração Máquina-Máquina. Porém, nos cenários onde existe a colaboração de máquinas, é importante ressaltar que nem sempre uma ou mais máquinas poderão executar tarefas de forma totalmente automática. Portanto, a interação humano-máquina se torna crucial para que haja uma boa execução das atividades desenvolvidas por essas máquinas.

Especificamente, quando o problema requer aplicação de algoritmos de visão computacional, a complexidade da solução tende a aumentar. Assim, tendo como motivação a automação da microscopia do clínquer, que será melhor descrita no capítulo seguinte, este trabalho se dispõe a contribuir para a automação de processos em indústrias cimenteiras. Os objetivos são mostrados a seguir.

2.4 Objetivos

Pelo exposto, o presente trabalho tem o seguintes objetivos.

2.4.1 *Objetivo Geral*

Propor uma interface para um sistema que deverá facilitar o treinamento de uma rede neural convolucional na detecção de cristais em imagens microscópicas de clínquer com o foco na usabilidade.

2.4.2 *Objetivos Específicos*

- Apresentar um levantamento de requisitos para criação da interface.
- Propor ferramentas que facilitem o desenvolvimento da interface proposta neste trabalho.
- Apresentar a elicitação dos requisitos.
- Criar um Protótipo do software.
- Criar o *backlog* de atividades futuras para continuidade do desenvolvimento.
- Validar a interface por meio de testes de usabilidade.
- Apresentar a interface proposta.

3 TRABALHOS RELACIONADOS

Neste capítulo serão apresentados 2 trabalhos relacionados, onde serão brevemente descritos e apresentadas as diferenças em relação a este trabalho. Logo após, será apresentada uma tabela comparativa entre os trabalhos encontrados e este.

3.1 Trabalho 1 - Arquitetura de sistema em nuvem para apoio à implantação de Visão Computacional em linhas de produção na Indústria 4.0

O trabalho desenvolvido pelo autor é a proposição de uma arquitetura voltada para visão computacional no âmbito da indústria 4.0. As semelhanças vão desde a proposição de uma arquitetura voltada para computação em nuvem, menos custosa e mais eficiente, até a melhoria de criação de modelos computacionais para reconhecimento de imagens utilizando uma interface web. Contudo, o trabalho não se preocupa como a interação entre o operador e o sistema.

3.2 Trabalho 2 - ODIN - Uma Ferramenta Visual para o Desenvolvimento de Modelos de Detecção de Objetos com *Deep Learning* no Ensino Superior

Esse trabalho tem como objetivo criar uma interface para auxiliar o processo de aprendizado de algoritmos de *machine learning* para alunos do ensino superior. Porém, o autor não utiliza ferramentas externas ao Google Colab para desenvolver a interface. Todo o processo de *upload*, segmentação, dentre outros processos presentes no fluxo de execução de treinamento e execução de um modelo é feito no próprio Colab.

Tabela 1 – Tabela comparativa entre os trabalhos relacionados

Atividades	Trabalho 1 - (ALMEIDA <i>et al.</i> , 2021)	Trabalho 2 - (SANTANA <i>et al.</i> , 2022)	TCC Arthur César
Ambiente para reconhecimento de imagens	X	X	X
Preocupação com a usabilidade		X	X
Interface Web	X		X
Desenvolvimento da própria ferramenta	X		X

Fonte: elaborado pelo autor (2022)

4 METODOLOGIA

Este capítulo é voltado para a apresentação do sistema, das tecnologias que serão utilizadas, bem como metodologias que serão aplicadas para o desenvolvimento da interface. Além disso é descrito o cenário em que a aplicação será inserida, quando concluída.

4.1 Aplicação para o sistema amigável

Para definir as metodologias utilizadas na usabilidade da interface, é preciso conhecer o cenário de aplicação do sistema a ser desenvolvido. Assim, aqui será dada uma descrição desse sistema.

4.1.1 *Microscopia do Clínquer*

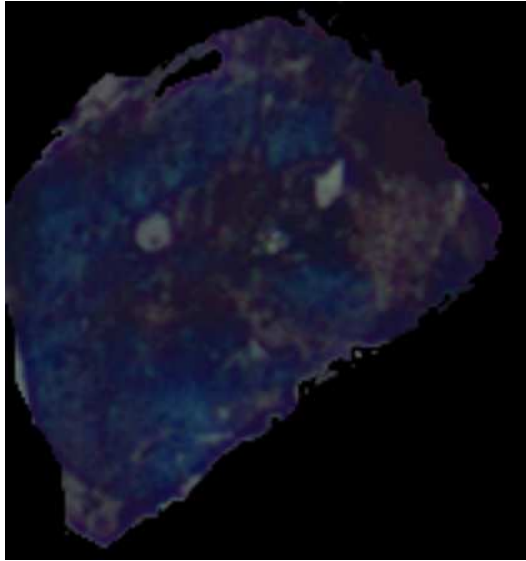
Em processo produtivo de cimento, argila, pedra calcária e outros minérios são moídos e, posteriormente, aquecidos a altas temperaturas em fornos rotativos para dar origem ao clínquer, que é o principal componente do cimento (PADILHA, 2022).

Uma importante etapa desse processo produtivo é o monitoramento do forno rotativo. Em geral esse monitoramento é realizado via raio-x e análise de imagens microscópicas do clínquer já produzido. Sendo essa última técnica conhecida como microscopia do clínquer. Na microscopia do clínquer, analistas com alto conhecimento do processo de produção de cimento identificam e analisam tamanho e forma dos cristais que aparecem na imagem microscópica do produto (PADILHA, 2022).

O principal cristal analisado nas imagens de clínquer é o composto C3S, também conhecido como Alita. Deste modo, o sistema deverá, de forma automática, realizar a identificação desse cristal em imagens microscópicas de clínquer, para que seja possível a realização da análise de suas características. Tais características ajudam no monitoramento das condições de operação do forno. Assim, este trabalho trata-se da proposição de uma interface para o sistema que deverá tornar possível a automação da identificação e segmentação dos cristais de C3S, cujo exemplo é mostrado na Figura 1.

Para realizar a identificação automática dos cristais de alita, será utilizado um algoritmo de aprendizado de máquina, que utiliza aprendizagem profunda, chamado *Mask R-CNN*. Uma rede neural convolucional que faz uso da *Faster R-CNN* adicionando o alinhamento pixel a pixel entre as entradas e saídas da rede. Tal alinhamento, forma a máscara de segmentação,

Figura 1 – Cristal de Alita



Fonte: Padilha (2022)

para cada pixel, que possui uma classe correspondente. Portanto, cada objeto desejado, que esteja presente na imagem, pode ser identificado dada a segmentação semântica aplicada por esse tipo de algoritmo (PADILHA, 2022) (HE *et al.*, 2017).

Diante do exposto, para o funcionamento correto do algoritmo, são necessárias duas entradas para a execução da Mask R-CNN: imagens dos objetos desejados para a detecção e um arquivo textual contendo informações das segmentações das imagens fornecidas. Tais informações contidas nesse arquivo deverão possuir as coordenadas dos vértices do polígono, que formam a máscara e as classes de cada polígono.

Portanto, o sistema a ser desenvolvido deverá possibilitar o corte, a segmentação de imagens e a atribuição de classes para cada polígono formado após a segmentação. As informações referentes a localização do objeto segmentado, bem como identificação de imagens e classes devem ser disponibilizadas em um arquivo em formato de texto, para serem utilizadas em calibração (aqui denominada treino) de algoritmos de detecção de objetos, os quais estão disponíveis nas bibliotecas da plataforma *AWS Sagemaker*.

O sistema será desenvolvido utilizando técnicas, heurísticas, ferramentas, dentre outros métodos que facilitam a sua usabilidade no contexto industrial. Por consequência, deverá requerer menos esforço humano para poder realizar atividades voltadas para a análise da microscopia do clínquer, bem como exigirá menos conhecimentos em programação ou na utilização de ferramentas que exigem muitos conhecimentos em programação.

4.2 Usabilidade na aplicação Web

Segundo Winckler e Pimenta (2002) a satisfação de um usuário, no contexto web é um dos critérios mais importantes para a determinação da qualidade da aplicação. Ainda segundo o autor, interfaces com uma boa usabilidade promovem um aumento na produtividade dos usuários, além de diminuir a ocorrência de erros. Ademais, o autor também destaca que a satisfação do usuário é um critério final para que o sistema seja revisitado. Deste modo, as ferramentas apresentadas aqui, visam a usabilidade da interface a ser criada.

4.3 Ciclo de concepção de interfaces web

Em Winckler e Pimenta (2002) é apresentado um ciclo de concepção de interfaces web. Nesta seção serão citadas algumas das etapas do ciclo proposto: Engenharia de Requisitos, Especificação do site e Design do Site. Além disso, serão pontuados itens a serem aplicados no sistema, bem como os motivos para sua aplicação.

4.3.1 Engenharia de Requisitos

Nesta etapa é feita a elicitação dos requisitos do sistema. Vale ressaltar que são apenas requisitos da interface (*front-end*). Nesta listagem, será utilizada a nomenclatura de Rota para toda a tela de usuário a ser criada.

A seguir serão listados os requisitos levantados até o momento.

1. O sistema deve conter uma rota que possibilite o login, utilizando como campos, email e senha.
2. O sistema deve conter uma rota que possibilite o cadastro de usuários, utilizando como campos e-mail, senha e nome.
3. O sistema terá uma rota que lista todos os projetos já criados do usuário logado.
4. O sistema deverá conter uma rota que possibilite a criação de projetos.
5. O sistema deve conter uma rota que permite o *upload* de imagens para cada projeto.
6. Na tela de carregamento de arquivos, em cada projeto, o sistema deve permitir uma interação onde o usuário poderá criar novas classes e associá-las aos polígonos já criados nas imagens do projeto.
7. O sistema deverá conter uma rota que submeta ao *backend* os carregamentos realizados para o projeto, juntamente com o documento em formato JSON, que irá conter todas as

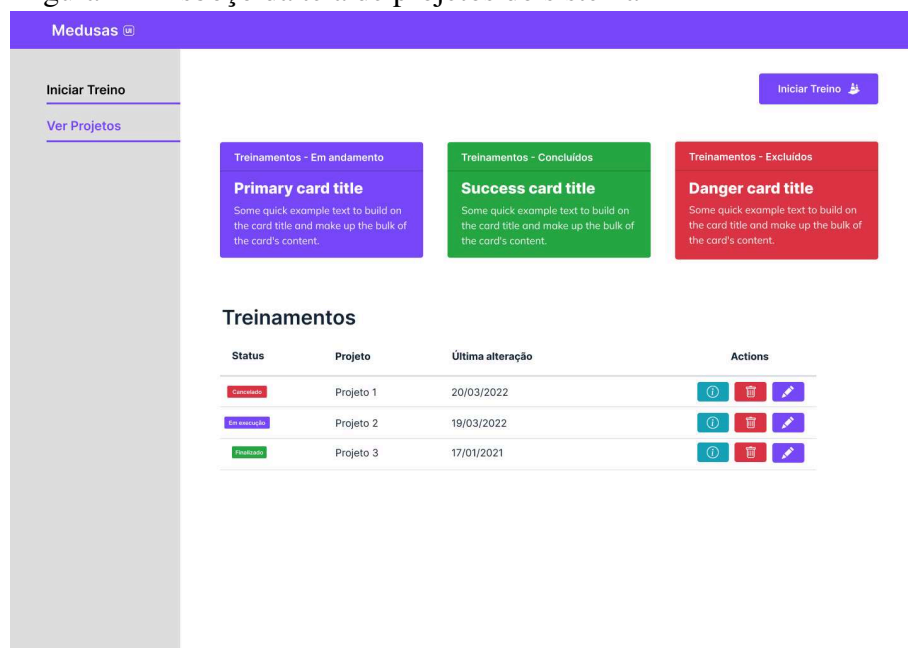
informações necessárias para novos treinamentos.

8. Na tela de projetos o usuário poderá ter acesso ao histórico de modelos gerados e os *uploads* feitos anteriormente, além de dar acesso também as classes criadas, possibilitando adição ou remoção.
9. O sistema deverá mostrar na página inicial cartões indicando a quantidade de treinamentos em andamento, concluídos, cancelados e não iniciados.
10. Na tela de projetos o usuário poderá ter acesso ao histórico de modelos gerados e os *uploads* feitos anteriormente, além de dar acesso, também, as classes criadas, possibilitando adição ou remoção.
11. O sistema terá uma rota que listará todas as detecções realizadas.
12. O sistema deverá conter uma rota que possibilite o usuário iniciar uma nova detecção, onde será feito o carregamento das imagens e a seleção da versão do modelo e do projeto que será utilizado.
13. O sistema deverá conter uma rota que dê acesso ao retorno da detecção.

4.3.2 Especificação do Site

É neste passo que são produzidos os modelos de telas, componentes e elementos em geral, baseados nos requisitos obtidos na etapa anterior. A Figura 2 mostra um modelo de página de projetos de um usuário.

Figura 2 – Esboço da tela de projetos do sistema

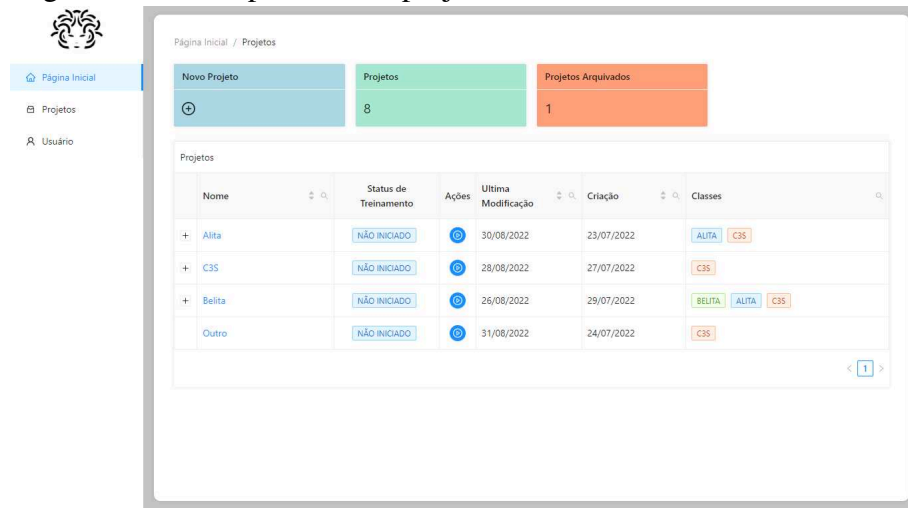


Fonte: elaborado pelo autor (2022)

4.3.3 Design do Site

Nesta etapa, os modelos obtidos no passo anterior são melhorados, dando mais detalhes aos elementos. A Figura 3 mostra um exemplo do protótipo criado baseado em um dos modelos obtidos anteriormente.

Figura 3 – Protótipo a tela de projetos do sistema



Fonte: elaborado pelo autor (2022)

4.3.4 Implementação do site

Nesta etapa é desenvolvida a interface e todos seus elementos, além de ser realizada a integração com outros elementos que compõem o sistema.

4.3.5 Esboço e Protótipo

É importante ressaltar a importância do detalhamento entre estas duas etapas, pois alguns requisitos foram adicionados depois das concepções dos modelos. Tais melhorias foram adicionadas direto no protótipo para não ser preciso voltar para os modelos iniciais, tendo em vista que a maior parte da estrutura que deverá ser mantida.

Comparando a Figura 3 com a Figura 2 é possível observar que, como citado anteriormente, a estrutura permanece praticamente a mesma. Já no componente de tabela, mudanças mais radicais foram adotadas, como a coluna de “ações” presente nos dois exemplos. Além disso, após essa concepção inicial (veja Figura 2) foi notado que mais informações precisariam ser adicionadas na tabela.

4.4 Heurísticas de Nielsen

Nielsen (2005) propõe dez heurísticas para se implementar ao desenvolver sistemas computacionais. Para desenvolver o sistema proposto, estas heurísticas devem servir como base para tomada de decisões de *design*, de arquitetura, de desempenho, dentre outras áreas que são abrangidas. Aqui, são enumeradas todas as heurísticas apresentadas.

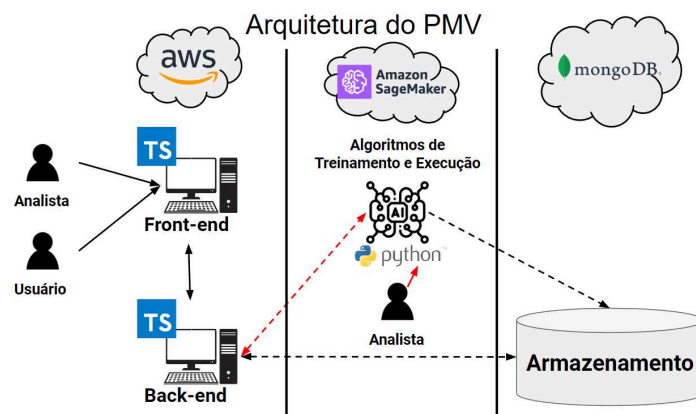
1. Visibilidade do status do sistema.
2. Semelhança com o mundo real.
3. Controle e liberdade do usuário.
4. Consistência e padrões.
5. Prevenção de erros.
6. Reconhecimento em vez de lembrança.
7. Flexibilidade e eficiência de uso.
8. Design estético e minimalista.
9. Ajudar os usuários a reconhecer, diagnosticar e recuperar de erros.
10. Ajuda e documentação.

Diante disso é importante salientar que estas heurísticas guiam o desenvolvimento, juntamente com os requisitos, dado que é importante que exista essa colaboração entre ambos.

4.5 Arquitetura do sistema

Nesta seção será descrita a arquitetura do projeto, incluindo a interface, como também o fluxo de ações para realizar as atividades de treino e execução da rede.

Figura 4 – Arquitetura do Sistema



Fonte: elaborado pelo autor (2022)

Diante do esquema mostrado na Figura 4, conclui-se que o sistema será utilizado por dois tipos de usuários distintos, que possuem necessidades semelhantes, porém com expectativas diferentes. Em Ponte e Silveira (2008) é desenvolvida uma avaliação da usabilidade de softwares usados no âmbito industrial, onde um dos métodos usados foi a criação e aplicação de questionários, baseados nas normas ISO/IEC 9126 e na ISO 9241-11, onde um dos objetivos foi entender os perfis de usuários dos softwares avaliados.

Neste trabalho, o sistema foi pensando, inicialmente, para atender apenas um tipo de usuário. Porém, já pensando em futuras expansões de contexto, foi adicionado um novo perfil, onde não será um operador ou analista de uma indústria. Desta forma, a interface foi idealizada para proporcionar uma melhor interação e usabilidade para todos os usuários.

Assim, os componentes, páginas e outros elementos que irão compor a interface, são idealizados para que as informações sejam completas. Porém, devem poder ser interpretadas e entendidas por todos os tipos de usuários. Por exemplo, as tabelas deverão ter apenas informações relevantes para o seu contexto, mas havendo interações que podem expandir essas informações, seja levando o usuário para outra tela, ou abrindo um modal, ou até mesmo expandindo o tamanho da mesma.

Ainda sobre os perfis de usuários obtidos em Ponte e Silveira (2008), foi mostrado que usuários do tipo "operadores" buscam uma interface mais simples e objetiva possível, com informações básicas para a realização de suas atividades. Já os usuários do tipo "engenheiro", que no caso deste trabalho são tratados como "analistas", almejam uma interface repleta de informações e com visual mais robustos do que é buscado pelos usuários anteriores.

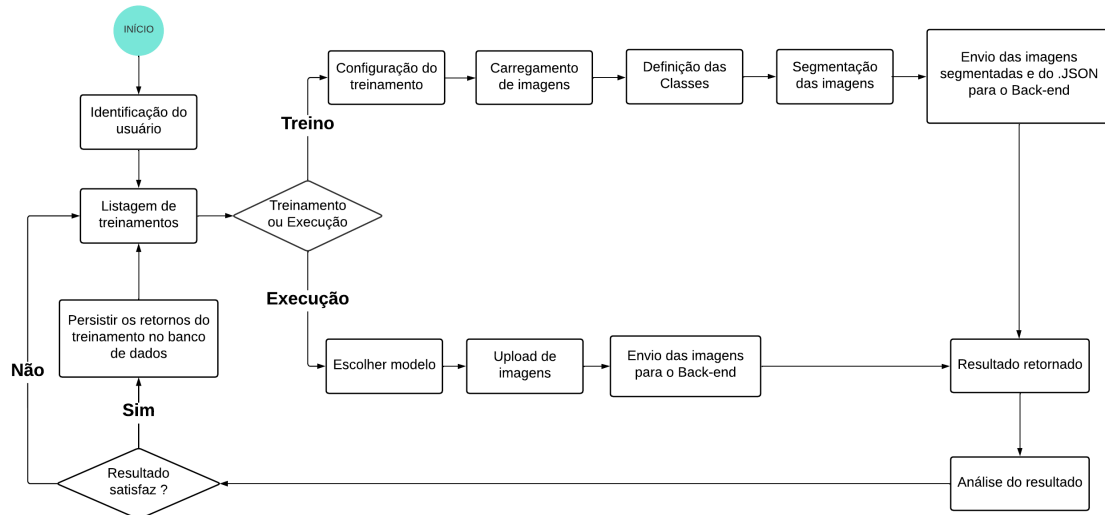
A interface, descrita como *Front-end* na Figura 4, fará a comunicação com uma *Application Programming Interface* (API), *Back-end*, onde ambas estarão hospedadas em servidores *Amazon Web Services* (AWS) (SERVICES, 2022). Os dados dos projetos, das segmentações, resultados obtidos, dentre outras informações devem ser trafegadas entre as aplicações por meio de requisições *Hypertext Transfer Protocol* (HTTP).

O *Back-end* será responsável pela comunicação com os algoritmos de detecção de objetos, os quais estão disponíveis nas bibliotecas da plataforma AWS Sagemaker (SAGEMAKER, 2022). Os retornos da rede serão salvos em um *MongoDB* e enviados para o *Back-end* poder processar e enviar para a interface.

4.6 Fluxograma da interface

Tendo em vista todo o processo para realização das atividades a serem realizadas durante o uso da interface, foi elaborado um fluxograma, apresentado na Figura 5.

Figura 5 – Fluxograma de funcionamento do sistema



Fonte: elaborado pelo autor (2022)

Tendo como base o fluxograma mostrado na Figura 5, a seguir é dada uma descrição das atividades que são realizadas pela interface.

1. **Identificar usuário:** Identificação do usuário através do seu nome de usuário e e-mail.
2. **Listar treinamentos:** Listagem dos treinamentos referentes ao usuário identificado. A página da listagem do treinamento conterá uma tabela contendo informações a respeito do treinamento,
3. **Treino ou execução**
 - **Treino**
 - **Configura treino:** O usuário ao criar um treino, preencherá um formulário com um campo para informar o nome do treino.
 - **Carregar imagens:** Envio de imagens, feito pelo usuário.
 - **Definição das Classes:** O usuário preencherá um formulário com campos para informar o nome das classes que serão identificadas na segmentação das imagens.
 - **Segmentação das imagens:** Para cada imagem, o usuário deverá segmentá-las formando polígonos. Para cada polígono formado em uma imagem, o usuário deverá atribuir uma classe a este polígono.

- **Envio das imagens segmentadas e do JSON para o *Back-end*:** Após finalizar a segmentação das imagens, será retornado um JSON contendo a estrutura de cada classe com informações das suas respectivas imagens segmentadas. Este JSON será enviado para o *Back-end*.
- **Execução**
 - **Escolher modelo:** Será selecionado um modelo para execução da previsão.
 - **Carregar imagens:** Será selecionado as imagens do modelo para a execução da previsão.
 - **Envio das imagens para o *Back-end*:** Após selecionar as imagens, será feito o envio das imagens para o S3 e será iniciado teste da previsão.
- 4. **Resultado retornado:** Resultado retornado do treino/execução.
- 5. **Analisar o resultado:** Verificação e avaliação do resultado.
- 6. **Resultado satisfaz ?**
 - **Sim:** Persistir os retornos do treino no banco de dados.
 - **Não:** Listar treinos.

4.7 Tecnologias

Na seção seguinte são apresentadas as tecnologias utilizadas para desenvolver a interface, bem como explicações sobre os motivos que levaram a escolha das mesmas.

4.7.1 *Typescript*

Com linguagem para o desenvolvimento da interface, foi escolhida o *Typescript*, a qual, não é uma linguagem de fato, mas sim uma extensão do *Javascript*. O *Typescript* vem para adicionar classes, interfaces, tipos estáticos e sistemas de módulos ao seu antecessor. Ao promover assistência aos programadores, os sistemas de módulos e de tipagem são fáceis e flexíveis de usar. Vale ressaltar que o *Typescript* deve se tornar uma nova linguagem, a intenção aqui é apenas melhorar o desenvolvimento usando *Javascript* (BIERMAN *et al.*, 2014).

O *Typescript* foi escolhido por facilitar a prevenção de erros, além de guiar melhor o desenvolvimento por padronizar atributos e tipos de objetos, parâmetros e retornos de métodos, além de indicar erros, quando acontecem, de forma mais específica. Além disso, a familiaridade com a linguagem facilita o processo de desenvolvimento. Logo, o sistema será desenvolvido

usando *React* com o *framework Next.js* com *Typescript*.

4.7.2 *React*

React é uma biblioteca *JavaScript* criada pelo *Facebook* que pode ser usada para projetar interfaces de usuário *online*. Ela foi lançada como software de código aberto em 2013 e tem visto um crescimento significativo entre os desenvolvedores desde então (WOHLGETHAN, 2018). Ao usar *React*, é importante dividir a interface do usuário em componentes menores que podem incluir um ou mais outros componentes cada um. A interface de usuário dos componentes *React* é formatada usando o pré-processador *JavaScript Extensible Markup Language* (XML) (JSX) (VAINIKKA, 2018).

4.7.3 *Ant design*

Para manter a consistência dos componentes de toda a aplicação, foi escolhido usar uma biblioteca de componentes prontos, porém facilmente customável, *Ant Design*. Dessa forma, os padrões de fonte, espaçamento, cores, bordas, dentre outras coisas, que são imprescindíveis para a consistência de uma aplicação web, serão padronizados para toda o sistema.

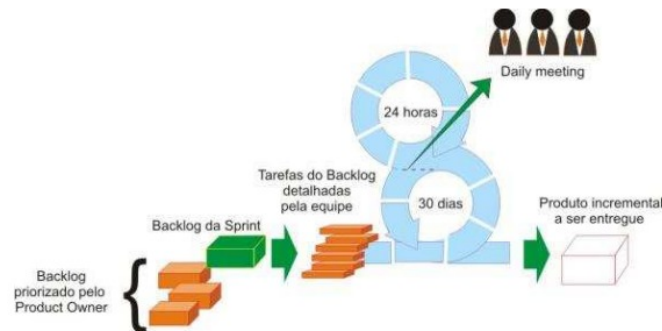
Desse modo, a biblioteca *Ant Design*, será utilizada para agilizar o processo de desenvolvimento, por agilizar o desenvolvimento, além de proporcionar componentes completos e de fácil uso que farão o sistema consistente e padronizado, além de ter compatibilidade com todas as outras tecnologias citadas anteriormente neste tópico.

4.8 Scrum

Para o desenvolvimento da interface será adotado a metodologia ágil *Scrum*. O funcionamento do *Scrum* é baseado em interações, chamadas de *Sprints*, que duram em volta de 2 a 4 semanas. Antes de dar início a uma interação, realiza-se uma reunião de planejamento, denominada de *Sprint Planning Meeting*, onde as atividades que serão desenvolvidas serão estimadas entre os desenvolvedores e o cliente *Product Owner*. Para a realização deste trabalho não serão adotadas as reuniões diárias chamadas *Daily Meetings*, apenas reuniões semanais. Ao final de uma *Sprint*, é realizada a *Sprint Review*, reunião feita para que seja demonstrado o produto gerado pelo desenvolvimento da *Sprint* (PEREIRA *et al.*, 2007).

Para acompanhamento das atividades é utilizada a plataforma *Jira*. (ATLASIAN,

Figura 6 – Scrum



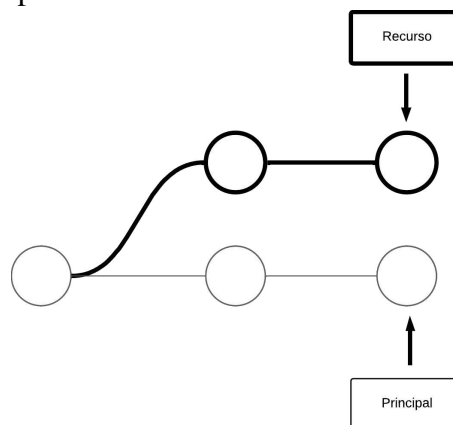
Fonte: (PEREIRA *et al.*, 2007)

2022a).

4.9 Gerenciamento do código

Com o objetivo de monitorar as progressões feitas durante o desenvolvimento da interface, é utilizado um sistema de gerenciamento de versões chamado Git. O uso de tal ferramenta, é justificado pelo auxílio na identificação e controle de modificações, além de garantir a implementação adequada das evoluções, bem como permitir que um histórico dos progressos sejam disponibilizados, dentre outras melhorias (RIVERO *et al.*, 2021) (ATLASIAN, 2022b). Atualmente o Git é um dos sistemas de versionamento mais usados no mundo, muito em função do seu código aberto, maduro e com uma manutenção frequente (RIVERO *et al.*, 2021). As características de ramificações do Git estão entre seus maiores benefícios. Os ramos do Git são baratos e simples de combinar, em contraste com as soluções de controle de versão centralizada.(ATLASIAN, 2022b)

Figura 7 – Diagrama com um exemplo de funcionamento do Git.



Fonte: adaptado de Atlassian (2022b)

Cada atualização de uma base de código pode ser feita em um ambiente isolado graças a ramos de características. Não importa quão grande ou pequeno seja o projeto, um desenvolvedor sempre começa criando uma nova ramificação. Isto garante que o código na ramificação principal esteja sempre apto para a produção. Como o Git é um sistema de controle de versão distribuído, cada desenvolvedor recebe seu próprio repositório local, completo com um histórico de *commits*. (ATLASIAN, 2022b)

Para este sistema, será adotado a plataforma Github para hospedar o repositório da interface, além de poder acompanhar todo o processo de mesclas, criação de novos ramos, dentre outras funcionalidades que o Git fornece.

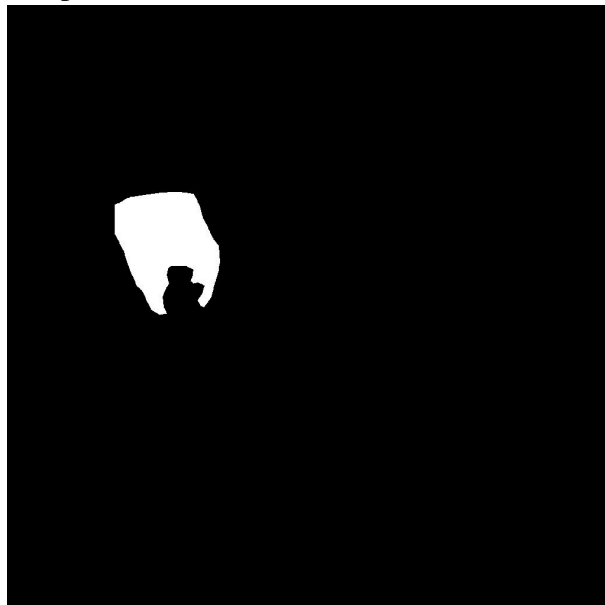
5 RESULTADOS

Nesta seção, são apresentados dois produtos relacionados, sendo o primeiro já desenvolvido e o segundo estando em fase de desenvolvimento, para a segmentação de imagens. O Produto 1 é uma ferramenta inicial com recursos básicos, enquanto o Produto 2 deve ser uma solução mais avançada e completa. Ambos compartilham a mesma abordagem metodológica e estilo de design, garantindo uma transição suave entre eles. Analisaremos os recursos do Produto 1 e as expectativas para o Produto 2 em desenvolvimento.

5.1 Produto 1 - Ferramenta *Medusas Eye's*

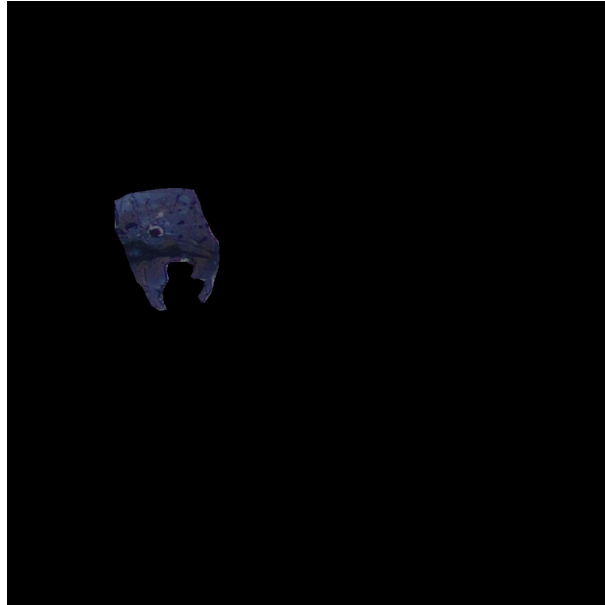
O Produto 1 é uma ferramenta de segmentação de imagens que oferece recursos básicos para anotação e criação de máscaras. Uma máscara é uma representação visual ou numérica que identifica e delimita as áreas de interesse em uma imagem. Ela é geralmente utilizada para separar objetos de interesse do restante da imagem, permitindo a segmentação precisa. Os algoritmos de segmentação utilizam essas máscaras como referência para separar e extrair os objetos de interesse da imagem original. As Figuras 8 e 9 apresentam exemplos de como se comportam as máscaras na interface.

Figura 8 – Saída do algoritmo com a máscara em preto e branco.



Fonte: elaborado pelo autor (2023)

Figura 9 – Saída do algoritmo com a máscara com suas cores originais.



Fonte: elaborado pelo autor (2023)

Foi desenvolvido com uma abordagem metodológica cuidadosa e um estilo de design intuitivo. Permite que os usuários realizem segmentações de forma eficiente e precisa. É uma solução inicial para atender às necessidades de segmentação manual de de imagens. Essas segmentações são necessárias para que sejam criadas amostras para calibração de algoritmos de detecção de objetos. Como, em geral, esse tipo de algoritmo requer muitas amostras para serem treinados, a ferramenta 1 vem para apoiar o usuário nessa tarefa.

5.1.1 Tela inicial - Upload

Na tela inicial, o usuário tem a possibilidade de inserir as informações necessárias, como classes e imagens, para acessar a tela de segmentação. Essa abordagem visa evitar erros futuros, como inserir apenas imagens sem as classes correspondentes, ou vice-versa. A interface foi projetada de forma a habilitar o botão *Create Project* somente quando ambos os campos essenciais forem preenchidos.

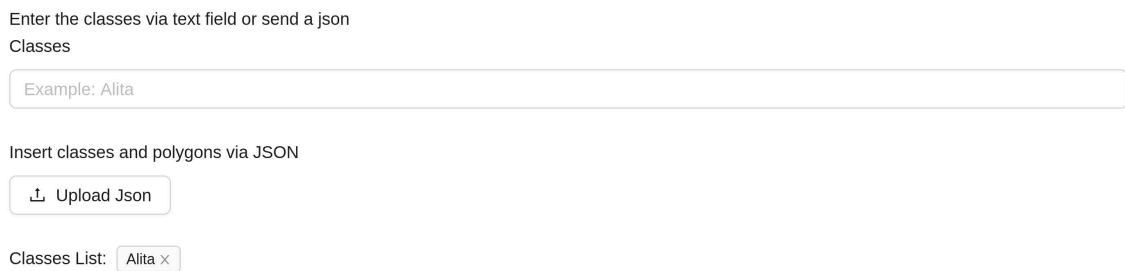
O campo de classes é responsável por receber as classes, permitindo que sejam inseridas uma por vez utilizando a tecla "Enter" do teclado. Abaixo desse campo, as classes inseridas são listadas, facilitando a visualização. Além disso, é possível remover uma classe específica clicando em um botão "X". Como pode ser visto na Figura 10.

Além da inserção das classes por meio do campo correspondente, é possível carre-

gar um arquivo JSON contendo as classes e outras informações adicionais, que são descritas posteriormente. O JSON recebido pela ferramenta pode ser o mesmo formato exportado por ela ou ter atributos semelhantes de acordo com a base *Common Objects in Context* (COCO). Essa base contém peso de algoritmos de detecção de objetos já calibrados (treinados) para outros problemas de detecção, e por isso são úteis para agilizar o treinamento, requerendo menos amostras do que seria preciso, caso não fossem utilizados pesos pré-treinados. Assim, esse formato é importante, pois garante compatibilidade e facilita a integração com outras ferramentas ou sistemas que também utilizam o formato para representar dados de segmentação de imagens.

Ao adotar esse formato padrão, a ferramenta possibilita que os usuários aproveitem as vantagens e recursos oferecidos pela base COCO, como a estrutura de anotações, categorias e segmentações. Isso simplifica o processo de compartilhamento e intercâmbio de dados entre diferentes ferramentas e fluxos de trabalho, proporcionando maior flexibilidade na análise e processamento dessas informações.

Figura 10 – Campos de upload de classes e JSON.



Enter the classes via text field or send a json

Classes

Example: Alita

Insert classes and polygons via JSON

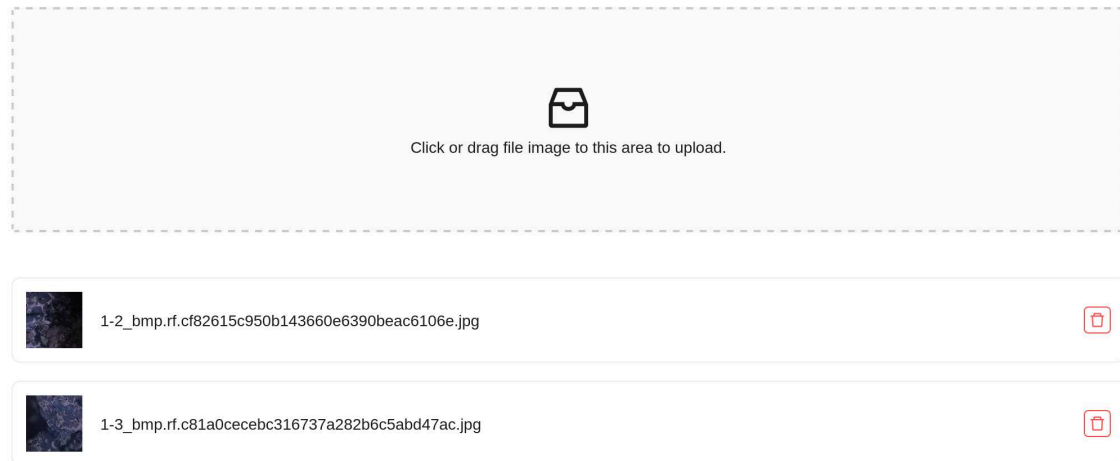
📁 Upload Json

Classes List: Alita x

Fonte: elaborado pelo autor (2023)

No campo de inserção de imagens cada item da lista exibe uma miniatura da imagem, seu nome e um ícone de exclusão, a Figura 11 mostra tal componente. Ao clicar no ícone de exclusão, o item correspondente é removido da lista e a imagem associada é excluída.

Figura 11 – Campo de *upload* de imagens e a lista de imagens



Fonte: elaborado pelo autor (2023)

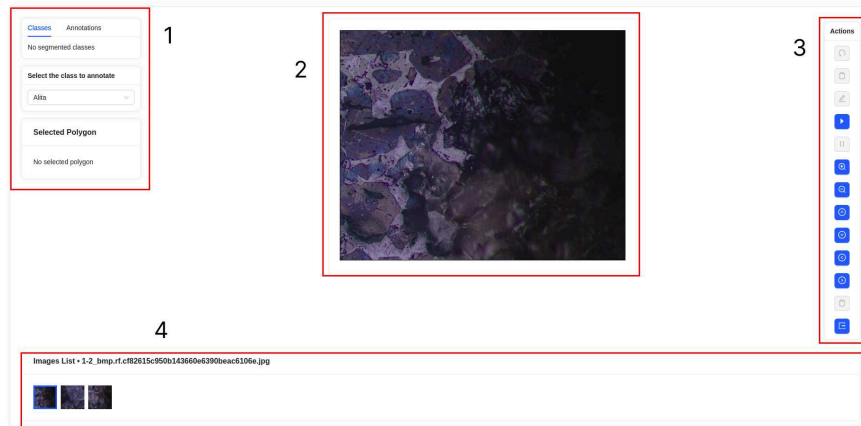
5.1.2 Tela de segmentação - Segmentation

Esta tela lista as imagens enviadas para a função de *upload* e permite a criação de máscaras para cada uma delas. Inicialmente, a tela é carregada com a primeira imagem enviada, no estado de “imagem selecionada”, permitindo sua visualização e criação de máscaras. No lado esquerdo da tela, encontram-se três elementos. O primeiro indica se existem máscaras na imagem selecionada e quais são as classes que foram utilizadas. O segundo permite ao usuário escolher a próxima classe a ser segmentada. O terceiro mostra se há uma máscara selecionada no momento. A Figura 12 ilustra a tela em questão.

Em casos em que o usuário navegue para a tela de segmentação por meio de uma rota direta e não haja imagens carregadas ou classes inseridas no campo correspondente, o usuário é automaticamente redirecionado para a tela de *upload*. Esse redirecionamento é implementado para garantir que o usuário seja guiado corretamente ao utilizar a ferramenta.

Dessa forma, evita-se que o usuário encontre uma tela vazia ou incompleta, fornecendo um fluxo de trabalho mais eficiente e consistente. Ao redirecionar o usuário para a tela de upload a interface busca oferecer uma experiência mais amigável e direcionada.

Figura 12 – Tela de segmentação.



Fonte: elaborado pelo autor (2023)

A primeira parte, destacada na imagem da Figura 12, tem como propósito fornecer ao usuário informações sobre as máscaras e suas respectivas classes, além de oferecer um *feedback* visual que permita ao usuário identificar facilmente quais classes já foram segmentadas.

Na aba de anotações, o usuário tem acesso a uma lista de máscaras. Ao clicar em um item dessa lista, o usuário pode selecionar a máscara correspondente àquela segmentação específica. Essa funcionalidade permite uma interação intuitiva e facilita a associação entre as classes e suas respectivas máscaras.

Além disso, é relevante mencionar que o usuário consegue excluir uma máscara ao clicar no botão representado pela lixeira. Essa opção permite ao usuário remover segmentações incorretas, contribuindo para a precisão e qualidade das anotações.

A aba de classes, mostra ao usuário uma lista de classes, onde cada item dessa lista consiste no nome da classe e a respectiva quantidade de máscaras segmentadas por imagem fornecendo uma visão geral da distribuição das máscaras. Essa informação é útil para entender a proporção de segmentação em cada classe.

As Figuras 13, 14, 15 mostram o comportamento do componente nas suas variadas versões.

Figura 13 – Cartão com listagem de máscaras.



Fonte: elaborado pelo autor (2023)

Figura 14 – Cartão com listagem de classes



Fonte: elaborado pelo autor (2023)

O componente também possui um comportamento caso não existam classes para ambas as abas. Como mostra a imagem da Figura 15.

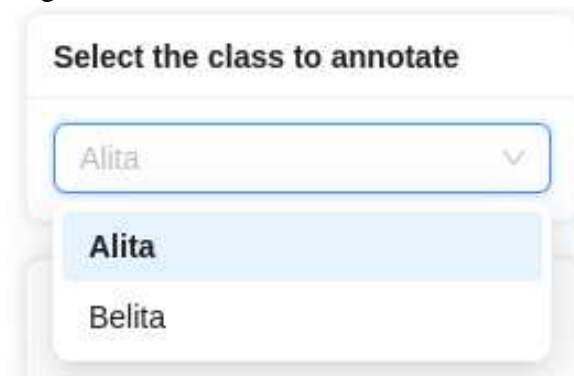
Figura 15 – Cartão com listagem de classes.



Fonte: elaborado pelo autor (2023)

Dando continuidade a primeira parte da interface, é notável a presença um componente gráfico que mostra as classes disponíveis para segmentação, além de destacar a classe atualmente selecionada. Como pode ser visto na Figura 16.

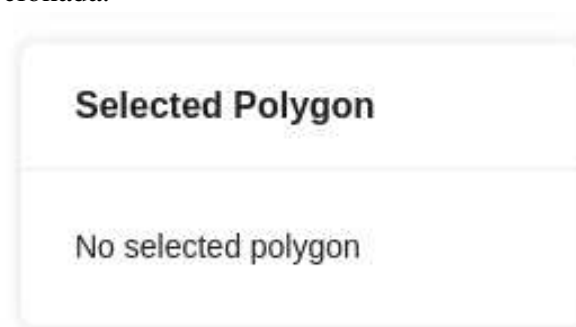
Figura 16 – Cartão de escolha de classe.

A dropdown menu titled "Select the class to annotate". The selected item is "Alita", indicated by a checkmark icon. Below the dropdown, a list of options is shown, with "Alita" highlighted in blue and "Belita" below it.

Fonte: elaborado pelo autor (2023)

O último elemento presente na parte inicial da interface possui a função de indicar a seleção de uma máscara. Esse elemento desempenha um papel crucial, fornecendo informações importantes sobre as máscaras e facilitando a identificação e busca por máscaras específicas. Ele apresenta dois estados distintos: quando uma máscara está selecionada e quando nenhuma máscara está selecionada. Esses estados são fundamentais, pois fornecem um *feedback* valioso ao usuário, mantendo-o informado sobre o progresso e as ações realizadas durante o processo de segmentação.

As Figuras 17, 18, exemplificam como se dá o comportamento do componente nas duas versões.

Figura 17 – Cartão de *status* sem máscara selecionada.A status card titled "Selected Polygon". The card has a light gray background and a thin border. The text "No selected polygon" is displayed in the center.

Fonte: elaborado pelo autor (2023)

Figura 18 – Cartão de *status* com máscara selecionada



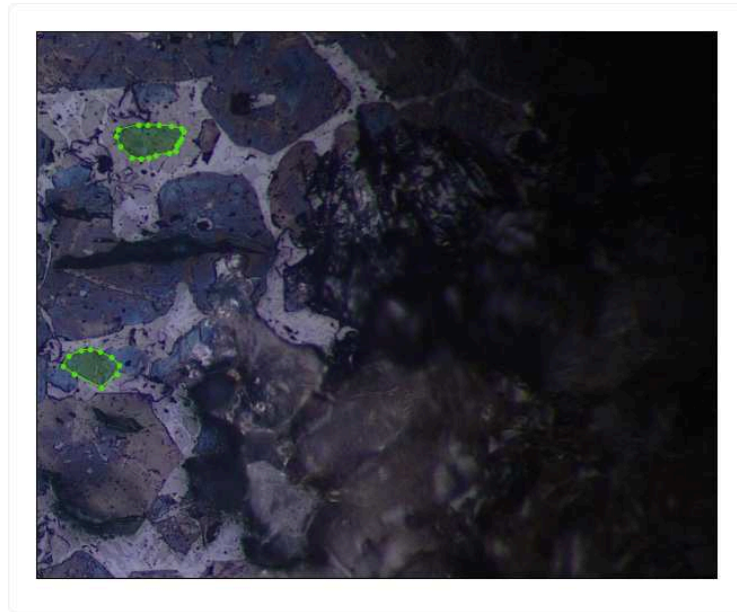
Fonte: elaborado pelo autor (2023)

A segunda parte da interface, que pode ser identificada na Figura 12, tem como objetivo exibir a imagem selecionada de maneira redimensionada, porém preservando sua proporção original, de forma a se adequar ao tamanho da tela. Essa abordagem é implementada com o intuito de evitar que, ao inserir uma imagem de grandes dimensões, a tela fique sobrecarregada com barras de rolagem excessivas, o que poderia comprometer a usabilidade da ferramenta.

Essa abordagem considera a experiência do usuário, proporcionando uma interface mais amigável e facilitando o uso da ferramenta mesmo com imagens de tamanhos variados. É nessa seção que as máscaras serão exibidas. Nas Figuras 19, 20, 21 pode-se encontrar como o componente se comporta ao ter máscaras, máscara selecionada e nenhuma máscara, respectivamente.

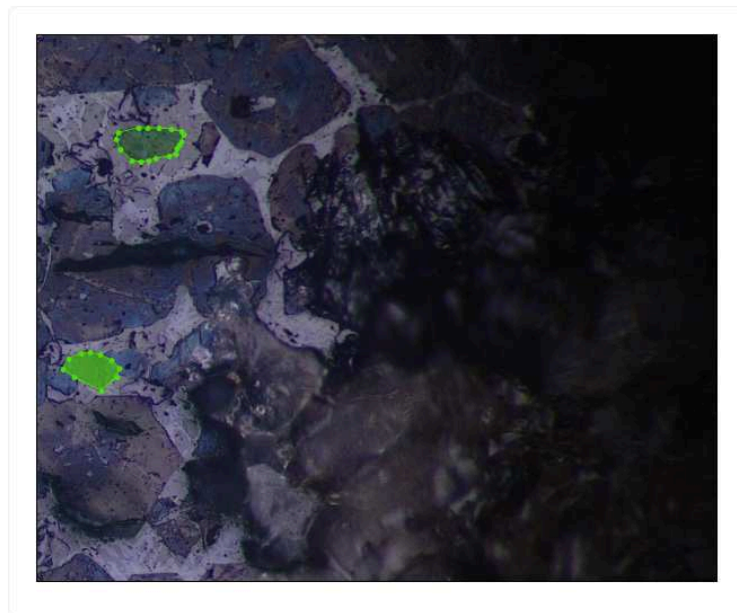
É importante ressaltar que essa parte da interface trabalha em conjunto com a seção dos botões, que pode ser identificada como a terceira parte da interface na Figura 12, que são responsáveis por acionar as ações que podem ser realizadas nessa área específica.

Figura 19 – Imagem com duas máscaras segmentadas.



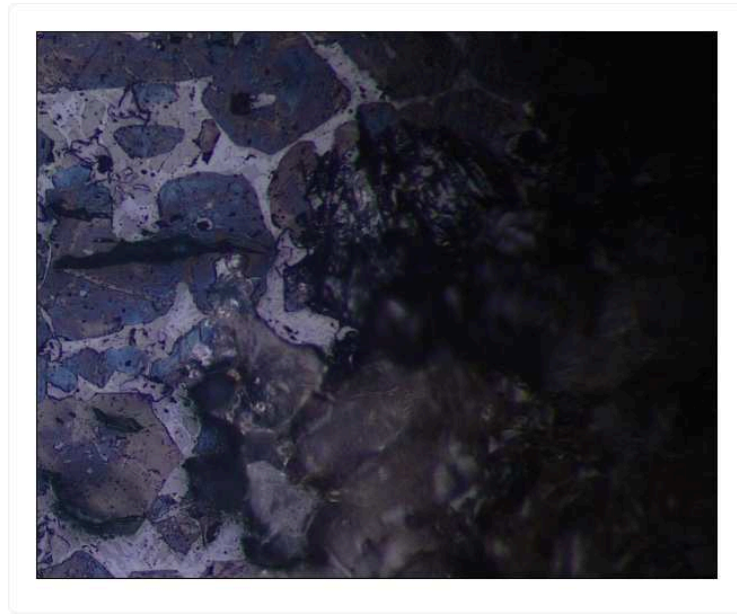
Fonte: elaborado pelo autor (2023)

Figura 20 – Imagem com uma das máscaras selecionada



Fonte: elaborado pelo autor (2023)

Figura 21 – Imagem sem máscaras segmentadas.



Fonte: elaborado pelo autor (2023)

A terceira parte da interface, de acordo com a Figura 12, é composta pela seção de botões, onde cada botão é responsável por acionar uma função específica de interação. Os botões disponíveis nessa seção permitem que o usuário realize diversas ações relacionadas à visualização e manipulação das máscaras na imagem destacada. Ao utilizar esses botões, o usuário tem maior controle sobre a experiência de segmentação das máscaras. Dessa forma, a seção dos botões e a segunda parte da interface trabalham em conjunto para proporcionar ao usuário uma ferramenta completa e intuitiva, permitindo que ele realize suas ações de segmentação de maneira eficiente e personalizada.

Abaixo estão listadas as funções associadas a cada botão:

1. Reverter último ponto da máscara: reverte a última ação realizada na máscara, permitindo desfazer o último ponto adicionado ou modificado.
2. Excluir ponto selecionado: remove o ponto atualmente selecionado na máscara, possibilitando ajustes e correções.
3. Editar ponto selecionado: permite a edição precisa do ponto selecionado na máscara, facilitando refinamentos e ajustes.
4. Iniciar segmentação: inicia o processo de segmentação da imagem, onde o usuário pode começar a marcar as áreas desejadas.
5. Finalizar máscara: finaliza o processo de marcação daquela máscara.
6. *Zoom in*: amplia a imagem para obter uma visualização mais detalhada e precisa.

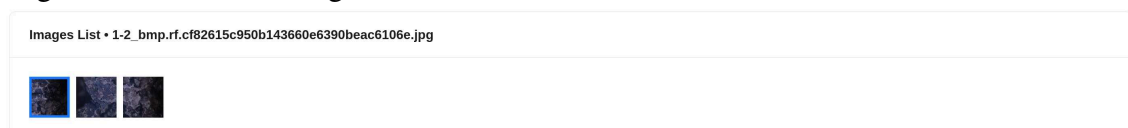
7. *Zoom out*: reduz a imagem para ter uma visão mais geral e ampla da área segmentada.
8. Mover para cima (quando com *zoom*): desloca a imagem para cima, permitindo a visualização de áreas superiores quando estiver aplicado o *zoom*.
9. Mover para baixo (quando com *zoom*): desloca a imagem para baixo, possibilitando visualizar áreas inferiores quando estiver aplicado o *zoom*.
10. Mover para esquerda (quando com *zoom*): desloca a imagem para a esquerda, permitindo a visualização de áreas laterais quando estiver aplicado o *zoom*.
11. Mover para direita (quando com *zoom*): desloca a imagem para a direita, possibilitando a visualização de áreas laterais quando estiver aplicado o *zoom*.
12. Excluir máscara: exclui completamente a máscara selecionada, removendo todas as marcações, referentes aquela máscara, feitas.
13. Exportar JSON: permite exportar os dados da segmentação em formato JSON, possibilitando o uso posterior desses dados em outras aplicações ou análises.

É importante ressaltar que os botões na interface são dinamicamente habilitados ou desabilitados de acordo com a ação que está sendo executada no momento. Por exemplo, no estado inicial, os botões de exclusão de máscara e pausa de segmentação são iniciados como desabilitados. Da mesma forma, os botões relacionados à interação com pontos selecionados também estarão desabilitados.

Essa abordagem tem o objetivo de fornecer uma experiência de usuário mais intuitiva e evitar ações indesejadas. Ao desabilitar os botões que não são relevantes para a ação atual, é possível reduzir a confusão e evitar cliques acidentais, proporcionando uma interação mais eficiente e livre de erros.

A quarta e última parte da interface, vista na Figura 12, consiste na lista de imagens. Lista a qual, são exibidas as miniaturas das imagens, onde o usuário pode navegar pela lista e selecionar a imagem desejada para iniciar o processo de segmentação. Como é representado na Figura 22.

Figura 22 – Lista de imagens.



Fonte: elaborado pelo autor (2023)

Quando uma imagem é selecionada, seu nome é destacado no título proporcionando

uma indicação visual clara da imagem em foco, uma borda azul é aplicada ao redor da miniatura da imagem. Isso ajuda o usuário a manter o controle e a identificar facilmente qual imagem está sendo segmentada. Essa abordagem na interface garante que o usuário possa selecionar rapidamente a imagem desejada para segmentação, mantendo a organização e facilitando a navegação entre as diferentes imagens disponíveis.

A utilização da cor azul, que é a cor principal do sistema, reforça a consistência visual e o design coeso da interface, garantindo que o destaque seja intuitivamente reconhecido pelo usuário.

5.1.3 Produto 2 - Sistema completo

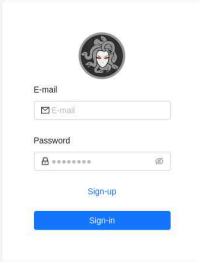
O Produto 2 está sendo desenvolvido para aprimorar ainda mais a experiência de segmentação de imagens, adicionando recursos avançados e uma interface mais poderosa. Ele irá fornecer um conjunto mais completo de funcionalidades, atendendo às demandas de usuários que buscam uma solução mais abrangente para suas necessidades de segmentação de imagens. Algumas funcionalidades que existirão no produto 2 que não estão presentes no produto 1.

O produto 2 possui cadastro de usuários, *login*, criação, listagem, edição e exclusão de projetos por conta, mudança de tema, configurações de usuário, além de todo o produto 1. Serão mostradas ainda nessa seção as telas e funcionalidades já implementadas.

5.1.4 Pagina Inicial

A tela inicial é composta pelo formulário de *login*, bem como os botões de *login* e cadastro, como pode ser visualizado na Figura 23. Ao inserir o email e a senha e clicar no botão de *login*, a interface realiza uma chamada à API para verificar as credenciais fornecidas. Se a API retornar um erro, uma mensagem de erro é exibida na tela, informando ao usuário que ocorreu um problema no *login*. Por outro lado, se as credenciais fornecidas estiverem corretas e a API confirmar a autenticação com sucesso, a interface pode redirecionar o usuário para a tela de projetos após o *login* bem-sucedido.

Figura 23 – Página de Login.



Fonte: Autor (2022).

5.1.5 *Página de cadastro*

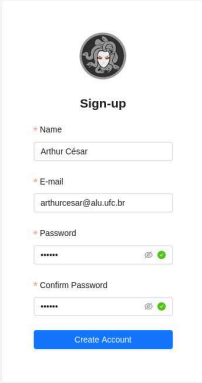
A tela de cadastro apresenta campos essenciais para o preenchimento das informações do usuário: nome, email, senha e confirmação de senha. Com o objetivo de garantir a precisão dos dados inseridos, a interface implementa a heurística "Diagnóstico e recuperação de erros". A Figura 24 mostra como a tela está atualmente.

Ao interagir com os campos, o usuário recebe *feedback* imediato por meio de mensagens que indicam se há algum erro. Por exemplo, se o campo *e-mail* for preenchido fora do padrão adotado para e-mails, uma mensagem de erro é exibida abaixo do campo, como mostrado na Figura 25.

Além disso, ao digitar a senha e a confirmação de senha, a interface compara os valores e, se não coincidirem, uma mensagem é exibida para alertar o usuário sobre a inconsistência, de acordo com a Figura 26. Dessa forma, o usuário tem a oportunidade de corrigir os campos que não foram preenchidos corretamente antes de prosseguir com o cadastro.

Essas mensagens de diagnóstico de erro fornecem orientações claras ao usuário, permitindo que ele identifique rapidamente os campos que precisam ser corrigidos. Com este método, a interface ajuda a evitar erros e assegura que as informações fornecidas sejam precisas e válidas para o cadastro no sistema.

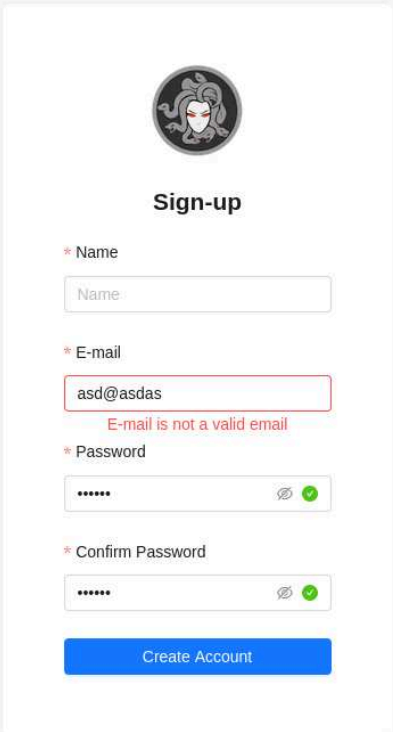
Figura 24 – Página de cadastro.



The image shows a 'Sign-up' form with a circular profile picture icon at the top. Below the icon is the title 'Sign-up'. The form contains four input fields, each with a red asterisk indicating it is required: 'Name' (containing 'Arthur César'), 'E-mail' (containing 'arthurcesar@alu.ufc.br'), 'Password' (containing six dots), and 'Confirm Password' (containing six dots). Each password field has a green eye icon and a green checkmark to its right. At the bottom of the form is a blue button labeled 'Create Account'.

Fonte: elaborado pelo autor (2023)

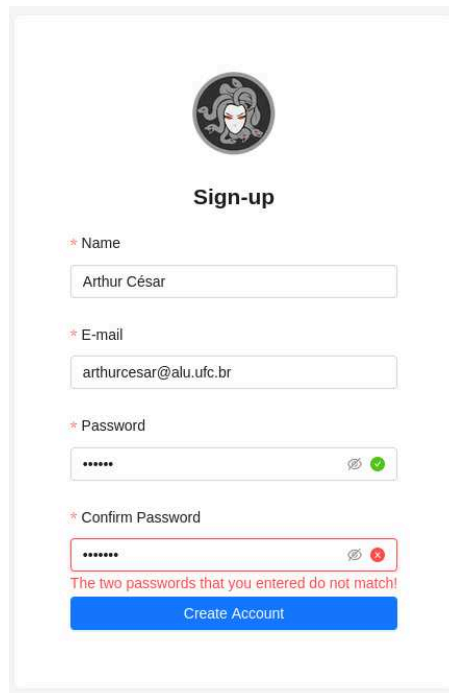
Figura 25 – Página de cadastro com indicação de erro no campo de e-mail.



The image shows the same 'Sign-up' form as in Figure 24, but with an error in the 'E-mail' field. The 'Name' field is empty. The 'E-mail' field contains 'asd@asdas' and is highlighted with a red border. Below the email field, the text 'E-mail is not a valid email' is displayed in red. The 'Password' and 'Confirm Password' fields are filled with six dots and have green eye icons and checkmarks. The blue 'Create Account' button is at the bottom.

Fonte: elaborado pelo autor (2023)

Figura 26 – Página de cadastro com indicação de erro no campo de confirmação de senha.



A imagem mostra uma interface web para cadastro, intitulada "Sign-up". No topo, há um ícone circular com uma máscara. Abaixo, há campos de entrada para "Name" (contendo "Arthur César"), "E-mail" (contendo "arthurcesar@alu.ufc.br") e "Password" (com caracteres ocultos por pontos). O campo "Confirm Password" também contém pontos, mas está rodeado por uma borda vermelha, indicando um erro. Abaixo deste campo, uma mensagem de erro em vermelho diz: "The two passwords that you entered do not match!". No rodapé da seção, há um botão azul com o texto "Create Account".

Fonte: elaborado pelo autor (2023)

5.2 Heurísticas de Nielsen na Prática

No decorrer desta seção, destacaremos algumas das heurísticas de Nielsen é aplicada no Produto 1, demonstrando como elas contribuem para a melhoria da usabilidade e eficácia do sistema como um todo. Com base nessas diretrizes, as ferramentas foram desenvolvidas com o objetivo de atender aos princípios de usabilidade, oferecendo aos usuários uma experiência fluida e satisfatória durante o processo de segmentação de imagens.

5.2.1 Visibilidade do status do sistema

No produto 1 a interface foi projetada para fornecer *feedback* visual ao usuário, indicando claramente em qual etapa do processo de segmentação se encontra. Essa abordagem ajuda os usuários a compreender quais ações estão disponíveis e quais não estão, facilitando a interação e a compreensão do contexto.

Um exemplo concreto dessa heurística pode ser observado na tela de *upload*, onde o usuário é orientado a preencher todos os campos necessários antes de prosseguir. O botão *Create Project* permanece desativado até que todos os campos sejam preenchidos corretamente, fornecendo uma indicação visual de que informações são necessárias.

Dessa forma, o usuário é guiado para a conclusão correta da tarefa e evita erros ou omissões. Além disso, a interface também retorna mensagens relevantes ao usuário ao executar ações nessa mesma tela. Essas mensagens fornecem *feedbacks* imediatos sobre o resultado da ação realizada, permitindo que o usuário compreenda e reaja adequadamente.

Outro exemplo é na tela de segmentação, onde alguns botões são desativados e ativados conforme o status da segmentação, tal como o botão de exclusão de máscara, que não ficará habilitado até que exista uma máscara selecionada.

5.2.2 Controle e liberdade do usuário

Para proporcionar um maior controle e liberdade aos usuários, a ferramenta oferece várias opções interativas. Os usuários têm a flexibilidade de inserir as classes manualmente, digitando seus nomes, ou importá-las através de um arquivo JSON. Essa abordagem permite que os usuários escolham o método de entrada mais conveniente para eles, adaptando-se às suas preferências e fluxos de trabalho.

Além disso, a ferramenta permite que os usuários tenham controle sobre os pontos de segmentação. Eles têm a liberdade de desfazer ou editar um ponto que não esteja de acordo com suas expectativas, garantindo uma experiência de segmentação mais precisa e satisfatória. Além disso, os usuários têm a opção de excluir completamente uma máscara, caso não seja mais necessária ou tenha sido criada por engano.

Outro aspecto em que a heurística é aplicada é na seleção das imagens para a segmentação. Os usuários podem escolher qual imagem desejam utilizar para inserir as máscaras. Isso permite que eles trabalhem com as imagens relevantes para suas tarefas específicas, proporcionando um controle total sobre o processo de segmentação.

Dessa forma, a heurística de "Controle e liberdade do usuário" é incorporada na ferramenta, garantindo que os usuários tenham a capacidade de tomar decisões, personalizar o fluxo de trabalho e realizar ações conforme suas necessidades e preferências individuais. Isso promove uma experiência mais intuitiva e empoderadora durante o uso da ferramenta.

5.2.3 Consistência e padrões

As interfaces dos produtos 1 e 2 seguem padrões de design consistentes em todo o sistema, garantindo uma experiência de uso familiar e previsível para os usuários.

A consistência é observada em diferentes aspectos da interface, como na disposição

dos elementos, na organização das seções e na representação visual dos botões e controles. Os usuários podem esperar que os elementos de interação estejam localizados nos mesmos lugares em todas as telas, facilitando a familiarização e a navegação eficiente.

Além disso, a utilização de padrões de design conhecidos e estabelecidos, como a representação de botões com ícones claros e compreensíveis, facilita o entendimento das funcionalidades disponíveis e o uso intuitivo da ferramenta. Isso reduz a curva de aprendizado e permite que os usuários se concentrem nas tarefas de segmentação de máscaras em vez de gastar tempo tentando entender como interagir com a interface.

A consistência e o uso de padrões também se estendem à linguagem e terminologia utilizadas na interface. Os rótulos dos botões, mensagens de feedback e instruções são formulados de maneira consistente e compreensível, seguindo convenções de linguagem adequadas ao contexto da segmentação de máscaras.

5.2.4 Prevenção de erros

Existem várias medidas implementadas na interface para prevenir erros. Uma delas é a desativação de botões ou controles que não são aplicáveis em determinados momentos, como a exclusão de uma máscara quando nenhuma está selecionada ou a edição ou remoção de um ponto quando nenhum ponto está selecionado. Isso evita que os usuários realizem ações incorretas e fornece um *feedback* visual claro sobre o estado atual do sistema.

Além disso, a interface realiza validações nos campos de entrada, como a lista de classes e o carregamento de imagens, para garantir que sejam fornecidos os dados necessários. Por exemplo, o botão *Create Project* na tela de *upload* continua desativado até que todos os campos obrigatórios sejam preenchidos corretamente, evitando que o usuário prossiga sem as informações necessárias.

Além das validações na própria tela de *upload*, a interface também oferece suporte ao redirecionamento do usuário para a tela inicial caso ele acesse a tela de segmentação via rota e não tenha fornecido os campos necessários anteriormente. Isso garante que o usuário seja direcionado de volta à etapa correta, permitindo que ele preencha os campos ausentes antes de prosseguir para a segmentação.

Outro aspecto que contribui para a prevenção de erros é a exibição de mensagens de *feedback* relevantes. Por exemplo, ao excluir uma máscara ou uma classe, uma mensagem é exibida ao usuário confirmando a ação realizada, evitando assim a exclusão acidental de

elementos importantes.

5.2.5 Reconhecimento em vez de recordação

A interface do Produto 1 apresenta informações e *feedback* visual de forma clara e visível, sem exigir que o usuário precise lembrar de detalhes ou informações anteriores. Por exemplo, ao carregar uma imagem para segmentação, o nome da imagem é destacado no título, facilitando o reconhecimento da imagem selecionada.

Além disso, a lista de classes disponíveis é exibida na interface, permitindo que o usuário visualize e selecione as classes desejadas para segmentação, sem precisar lembrar os nomes das classes previamente inseridas.

Essa forma de reconhecimento em vez de lembrança torna a interação mais intuitiva e reduz a carga cognitiva do usuário, facilitando a utilização do Produto 1 e proporcionando uma experiência mais eficiente e satisfatória

5.2.6 Flexibilidade e eficiência de uso

A interface do Produto 1 oferece recursos e opções que permitem aos usuários realizar tarefas de segmentação de forma flexível e eficiente. Por exemplo, os usuários têm a opção de inserir as classes manualmente ou importá-las através de um arquivo JSON, o que proporciona flexibilidade na forma como as classes são adicionadas.

Além disso, a interface permite que os usuários desfaçam ou editem pontos de máscara, excluam máscaras inteiras, escolham qual imagem desejam segmentar e apliquem *zoom* para uma segmentação mais precisa. Essas opções e recursos dão aos usuários a capacidade de adaptar a ferramenta às suas necessidades específicas e realizarem suas tarefas de segmentação de maneira eficiente.

5.2.7 Estética e design minimalista

Os produtos 1 e 2 seguem uma abordagem de *design* minimalista e estético, com o objetivo de fornecer uma experiência visualmente agradável e livre de distrações. A interface apresenta uma organização clara e concisa dos elementos, com ênfase na simplicidade e na legibilidade.

A disposição dos elementos na interface é cuidadosamente projetada para garantir a

compreensão intuitiva das funcionalidades e fluxos de trabalho. As informações são apresentadas de forma hierárquica e estruturada, permitindo que os usuários naveguem facilmente pelas diferentes páginas e seções e realizem suas tarefas de segmentação de forma eficiente.

A paleta de cores utilizada na interface é escolhida de forma agradável aos olhos. Elementos visuais como botões, ícones e campos de entrada são projetados de forma simples e clara, facilitando a identificação e a interação do usuário.

5.2.8 Ajuda e documentação

O produto 1 possui uma documentação disponível em César (2023). Nessa documentação é possível encontrar as tecnologias usadas, bem como um guia de uso que facilitará a ambientação do usuário com a ferramenta.

5.2.9 Diagnóstico e recuperação de erros

A interface do Produto 1 possui mecanismos que ajudam os usuários a identificar e corrigir erros. Por exemplo, ao realizar ações na tela inicial, como exclusão de classes ou imagens, a interface exibe mensagens de retorno imediato informando o resultado da ação. Se ocorrer algum erro durante o processo, uma mensagem de erro específica é exibida.

5.3 Testes

Durante o desenvolvimento, foram realizados testes com membros da equipe do Projeto *Medusas Eye's*. Foi ao decorrer desses testes que foram identificadas oportunidades de melhoria, especialmente relacionadas ao arquivo JSON exportado. Os integrantes da equipe destacaram que essas melhorias eram necessárias para garantir que o arquivo fosse facilmente aceito por determinados algoritmos.

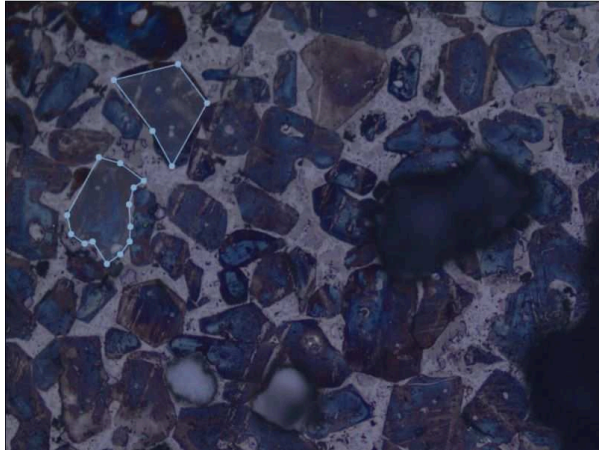
Com base nesse *feedback*, foi criado um formato de arquivo, inspirado no formato COCO. Essas alterações impactaram algumas funcionalidades, como a exportação e importação do arquivo JSON. No entanto, após todas as modificações realizadas, constatou-se que o arquivo de saída estava satisfatório e era compatível com os algoritmos previstos.

Essas melhorias no formato do arquivo JSON foram implementadas visando aprimorar a interoperabilidade da ferramenta e garantir uma integração mais eficiente com outros sistemas e algoritmos de processamento de dados. O resultado final foi um arquivo de saída bem

recebido pelos integrantes da equipe de trabalho e capaz de atender às necessidades específicas dos algoritmos relacionados ao projeto *Medusas Eye's*.

Ao longo dos testes, foi observado que as coordenadas dos pontos das máscaras estavam incorretas devido à aplicação da escala da imagem. Isso foi evidenciado por meio das Figuras 27 e 28, que ilustram o problema.

Figura 27 – Máscaras na interface.



Fonte: elaborado pelo autor (2023)

Figura 28 – Máscaras com as coordenadas retornadas pelo algoritmo.



Fonte: elaborado pelo autor (2023)

É possível notar que existe uma disparidade entre as imagens, pois estão com as máscaras com coordenadas diferentes, onde na Figura 27 mostra a ferramenta com duas máscaras

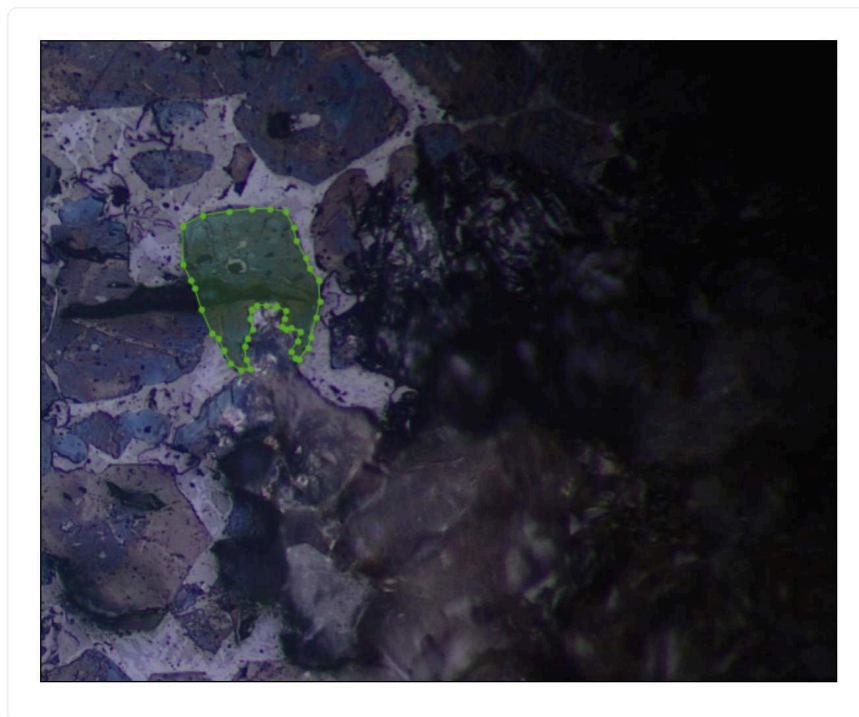
segmentadas e na imagem 28 é notável que o algoritmo recebeu coordenadas errôneas no que diz respeito a proporção da imagem.

Para solucionar essa questão, foi implementada uma função que calcula as coordenadas corretas a serem aplicadas na imagem original, levando em consideração a escala adotada para a imagem redimensionada. Essa função permite mapear os pontos da máscara na imagem original, mesmo após o redimensionamento.

Com essa melhoria, as coordenadas dos pontos das máscaras estão agora corretamente alinhadas com a imagem original, garantindo uma segmentação precisa e consistente, independentemente do redimensionamento realizado. Isso contribui para uma experiência de uso mais confiável e facilita a integração com outros processos que dependem dessas coordenadas corretas.

As Figuras 29 e 30 são exemplos de como as funções de cálculo de redimensionamento de coordenadas estão se comportando de maneira adequada e satisfatória.

Figura 29 – Máscara inserida na Ferramenta 1.



Fonte: elaborado pelo autor (2023)

Figura 30 – Máscara com as coordenadas retornadas pelo algoritmo.



Fonte: elaborado pelo autor (2023)

6 CONCLUSÕES E TRABALHOS FUTUROS

Durante o desenvolvimento da ferramenta de segmentação de imagens, o objetivo principal foi criar uma interface amigável para facilitar o treinamento de uma rede neural convolucional na detecção de cristais em imagens microscópicas de clínquer. Considerando que esse processo requer conhecimento em algoritmos de aprendizado de máquina, foram adotadas metodologias que visam simplificar o uso dos produtos desenvolvidos.

Com base nas diretrizes propostas por Nielsen (2005), a interface do Produto 1 foi cuidadosamente projetada para atender às heurísticas estabelecidas. Além disso, ela tem sido utilizada pelos integrantes do Projeto *Medusa's Eyes*, cumprindo as expectativas e requisitos levantados pelo time.

Ao adotar uma abordagem centrada no usuário e incorporar as heurísticas de Nielsen, a interface proporciona uma experiência intuitiva, permitindo que os usuários realizem as tarefas de segmentação com facilidade e eficiência. O *feedback* recebido durante o desenvolvimento e o uso pelos integrantes do projeto validam a eficácia da interface em atender às necessidades dos usuários.

6.1 Melhorias

Durante o desenvolvimento e os testes do Produto 1, foram identificadas oportunidades de melhoria. Uma delas é permitir que o usuário personalize as cores das classes para facilitar a identificação nas imagens. Além disso, buscando tornar a interface ainda mais intuitiva, foi idealizada uma melhoria relacionada à interação com o *zoom*.

Atualmente, para mover a visualização da imagem para outra área quando o *zoom* está aplicado, o usuário precisa clicar repetidamente nos botões de direção correspondentes à sua intenção. Com o objetivo de facilitar essa interação, foi pensada a funcionalidade de mudar a visualização por meio de clique e arrasto do *mouse*. Dessa forma, o usuário poderá simplesmente clicar na imagem e, mantendo o botão pressionado, arrastar o cursor para mover a visualização para a área desejada.

Portanto para próximas etapas têm-se como objetivo trazer e implementar melhorias para o Produto 1, além de concluir o desenvolvimento do Produto 2.

6.2 Limitações

O foco principal do desenvolvimento do Produto 1 não foi direcionado para dispositivos móveis, uma vez que as telas desses dispositivos tendem a ser pequenas, o que dificulta o processo de segmentação. Uma boa experiência de uso requer que o usuário tenha uma visão abrangente da interface, a fim de usufruir de todos os mecanismos de maneira adequada.

Com base nos resultados obtidos, conforme ilustrado nas Figuras 30 e 30, e na positiva receptividade relatada pelos membros da equipe de algoritmos, pode-se concluir que o Produto 1 atingiu seus objetivos estabelecidos. Essa conclusão evidencia que a ferramenta cumpriu suas funcionalidades de maneira satisfatória.

Como resultado do sucesso alcançado, planeja-se integrar o Produto 1 ao Produto 2 em futuras etapas de desenvolvimento. Essa integração visa aprimorar ainda mais a ferramenta, tornando-a mais robusta e completa, atendendo às demandas e expectativas do projeto como um todo. Dessa forma, busca-se oferecer aos usuários uma solução abrangente e eficiente para o treinamento de redes neurais convolucionais na detecção de cristais em imagens microscópicas de clínquer.

REFERÊNCIAS

- ALMEIDA, C. D. F. de; LIMA, J. P. d. O.; OLIVEIRA, J. V. M. de; OLIVEIRA, R. C. de; OLIVEIRA, E. F. Arquitetura de sistema em nuvem para apoio à implantação de visão computacional em linhas de produção na indústria 4.0. In: **Simpósio Brasileiro de Automação Inteligente-SBAI**. [S. l.: s. n.], 2021. v. 1, n. 1.
- ANIL, S. L.; THANKA, R. A survey on security of data outsourcing in cloud. **International Journal of Scientific and Research Publications (IJSRP)**, Citeseer, v. 3, 2013.
- ATLASIAN. **Para que serve o Jira?** 2022. Disponível em: <https://www.atlassian.com/br/software/jira/guides/use-cases/what-is-jira-used-for#jira-for-agile-teams>. Acesso em: 04 dez. 2022.
- ATLASIAN. **Por que usar Git na sua empresa.** 2022. Disponível em: <https://www.atlassian.com/br/git/tutorials/why-git>. Acesso em: 03 dez 2022.
- BIERMAN, G.; ABADI, M.; TORGENSEN, M. Understanding typescript. In: SPRINGER. **European Conference on Object-Oriented Programming**. [S. l.], 2014. p. 257–281.
- CARBONELL, J. G.; MICHALSKI, R. S.; MITCHELL, T. M. An overview of machine learning. **Machine learning**, Elsevier, p. 3–23, 1983.
- CÉSAR, A. **Ferramenta de Segmentação.** 2023. <https://github.com/nogueiraac/segmentation-tool>. Acesso em: 6 de jul. de 2023.
- GIRSHICK, R. Fast r-cnn. In: **Proceedings of the IEEE international conference on computer vision**. [S. l.: s. n.], 2015. p. 1440–1448.
- GIRSHICK, R.; DONAHUE, J.; DARRELL, T.; MALIK, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S. l.: s. n.], 2014. p. 580–587.
- GOYAL, A.; DADIZADEH, S. A survey on cloud computing. **University of British Columbia Technical Report for CS**, v. 508, p. 55–58, 2009.
- HE, K.; GKIOXARI, G.; DOLLÁR, P.; GIRSHICK, R. Mask r-cnn. In: **Proceedings of the IEEE international conference on computer vision**. [S. l.: s. n.], 2017. p. 2961–2969.
- KINZEL, H. Industry 4.0—where does this leave the human factor? **Journal of Urban Culture Research**, v. 15, p. 70–83, 2017.
- LECUN, Y.; BOSER, B.; DENKER, J. S.; HENDERSON, D.; HOWARD, R. E.; HUBBARD, W.; JACKEL, L. D. Backpropagation applied to handwritten zip code recognition. **Neural computation**, MIT Press, v. 1, n. 4, p. 541–551, 1989.
- LIMA, A. G. de; PINTO, G. S. Indústria 4.0: um novo paradigma para a indústria. **Revista Interface Tecnológica**, v. 16, n. 2, p. 299–311, 2019.
- LIU, H.; LANG, B. Machine learning and deep learning methods for intrusion detection systems: A survey. **applied sciences**, Multidisciplinary Digital Publishing Institute, v. 9, n. 20, p. 4396, 2019.

MILANO, D. de; HONORATO, L. B. Visao computacional. **Faculdade de Tecnologia, Universidade Estadual de Campinas**, 2010.

NIELSEN, J. **Ten usability heuristics**. [S. l.]: <http://www.nngroup.com/articles/ten-usability-heuristics/>(acc-essed . . . , 2005.

OUSSOUS, A.; BENJELLOUN, F.-Z.; LAHCEN, A. A.; BELFKIH, S. Big data technologies: A survey. **Journal of King Saud University-Computer and Information Sciences**, Elsevier, v. 30, n. 4, p. 431–448, 2018.

PADILHA, R. M. S. Automação da identificação de c3s na microscopia do clínquer. 2022.

PAPCUN, P.; KAJÁTI, E.; KOZIOREK, J. Human machine interface in concept of industry 4.0. In: IEEE. **2018 World Symposium on Digital Intelligence for Systems and Machines (DISA)**. [S. l.], 2018. p. 289–296.

PEREIRA, P.; TORREÃO, P.; MARÇAL, A. S. Entendendo scrum para gerenciar projetos de forma ágil. **Mundo PM**, v. 1, p. 3–11, 2007.

PERUZZINI, M.; GRANDI, F.; PELLICCIARI, M. Benchmarking of tools for user experience analysis in industry 4.0. **Procedia manufacturing**, Elsevier, v. 11, p. 806–813, 2017.

PK, F. A. What is artificial intelligence? “**Success is no accident. It is hard work, perseverance, learning, studying, sacrifice and most of all, love of what you are doing or learning to do**”, p. 65, 1984.

PONTE, M. J. M. da; SILVEIRA, A. M. da. Uma metodologia para avaliação da usabilidade de softwares de automação industrial utilizando redes neurais artificiais: Um estudo de caso eletrobrás. 2008.

RAY, P. P. A survey of iot cloud platforms. **Future Computing and Informatics Journal**, Elsevier, v. 1, n. 1-2, p. 35–46, 2016.

RAY, S. A quick review of machine learning algorithms. In: IEEE. **2019 International conference on machine learning, big data, cloud and parallel computing (COMITCon)**. [S. l.], 2019. p. 35–39.

RIVERO, L.; SILVA, I.; CUTRIM, P.; PAIVA, A.; OLIVEIRA, M.; ALVES, E.; JUNIOR, G. B.; DIAS, D. Implementando o gitflow para gerencia de configuração em um projeto de desenvolvimento de software ágil: Um relato de experiência. **Anais do Computer on the Beach**, v. 12, p. 178–185, 2021.

SAGEMAKER, A. **Amazon SageMaker**. 2022. Disponível em: <https://aws.amazon.com/pt/sagemaker/>. Acesso em: 05 dez. 2022.

SAKURAI, R.; ZUCHI, J. D. As revoluções industriais até a indústria 4.0. **Revista Interface Tecnológica**, v. 15, n. 2, p. 480–491, 2018.

SANTANA, J. P. *et al.* Uma ferramenta visual para o desenvolvimento de modelos de detecção de objetos com deep learning no ensino superior. Florianópolis, SC., 2022.

SERVICES, A. W. **Amazon Web Services**. 2022. Disponível em: <https://aws.amazon.com/pt/>. Acesso em: 05 dez. 2022.

SILVA, D. G. d. **Indústria 4.0: conceito, tendências e desafios**. Dissertação (B.S. thesis) – Universidade Tecnológica Federal do Paraná, 2017.

VAINIKKA, J. Full-stack web development using django rest framework and react. Metropolia Ammattikorkeakoulu, 2018.

WAIBEL, A.; HANAZAWA, T.; HINTON, G.; SHIKANO, K.; LANG, K. J. Phoneme recognition using time-delay neural networks. **IEEE transactions on acoustics, speech, and signal processing**, IEEE, v. 37, n. 3, p. 328–339, 1989.

WINCKLER, M.; PIMENTA, M. S. Avaliação de usabilidade de sites web. **Escola de Informática da SBC Sul (ERI 2002). Porto Alegre**, v. 1, p. 85–137, 2002.

WOHLGETHAN, E. **Supporting web development decisions by comparing three major javascript frameworks: Angular, react and vue.js**. Tese (Doutorado) – Hochschule für Angewandte Wissenschaften Hamburg, 2018.

XU, J.; FU, D.; SHAO, L.; ZHANG, X.; LIU, G. A soft sensor modeling of cement rotary kiln temperature field based on model-driven and data-driven methods. **IEEE Sensors Journal**, IEEE, v. 21, n. 24, p. 27632–27639, 2021.