

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
CURSO DE SISTEMAS DE INFORMAÇÃO**

Mirian de França Santos Pereira

**DESENVOLVIMENTO DE UM CURSO PARA ENSINAR *MACHINE LEARNING* NO
ENSINO MÉDIO VOLTADO AO ESTÁGIO DE CRIAR**

Florianópolis
2022

Mirian de França Santos Pereira

**DESENVOLVIMENTO DE UM CURSO PARA ENSINAR *MACHINE LEARNING* NO
ENSINO MÉDIO VOLTADO AO ESTÁGIO DE CRIAR**

Trabalho de Conclusão do Curso de Graduação em Sistemas de Informação, do Departamento de Informática e Estatística, do Centro Tecnológico da Universidade Federal de Santa Catarina, apresentado como requisito parcial à obtenção do título de Bacharel em Sistemas de Informação.

Orientadora: Prof.^a Dr.^a rer. nat. Christiane Gresse von Wangenheim, PMP.

Florianópolis

2022

Ficha de identificação da obra elaborada pelo autor, através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Pereira, Mirian de França Santos
Desenvolvimento de um curso para ensinar machine learning no ensino médio voltado ao estágio de criar /
Mirian de França Santos Pereira ; orientador, Christiane Gresse von Wangenheim, coorientador, Jean Carlo R. Hauck, 2022.
67 p.

Trabalho de Conclusão de Curso (graduação) -
Universidade Federal de Santa Catarina, Centro Tecnológico,
Graduação em Sistema de Informação, Florianópolis, 2022.

Inclui referências.

1. Sistema de Informação. 2. Programação. 3. Inteligência Artificial. 4. App Inventor. 5. Ensino Médio. I. Gresse von Wangenheim, Christiane. II. Hauck, Jean Carlo R. . III. Universidade Federal de Santa Catarina. Graduação em Sistema de Informação. IV. Título.

MIRIAN DE FRANÇA SANTOS PEREIRA

**DESENVOLVIMENTO DE UM CURSO PARA ENSINAR MACHINE LEARNING NO
ENSINO MÉDIO VOLTADO AO ESTÁGIO DE CRIAR**

Trabalho de conclusão de curso submetido ao Departamento de Informática e Estatística da Universidade Federal de Santa Catarina para a obtenção do Grau de Bacharelado em Sistemas de Informação.

Florianópolis, 30 de junho de 2022

Orientadora:

Prof.^a Dr.^a rer. nat. Christiane Gresse von Wangenheim, PMP

Orientadora

Universidade Federal de Santa Catarina

Co-Orientador:

Prof. Dr. Jean Carlo R. Hauck

Avaliador

Universidade Federal de Santa Catarina

Banca Examinadora:

Ramon Mayor Martins

Avaliador

RESUMO

A tecnologia de Inteligência Artificial (IA) vem se tornando cada vez mais influente atualmente, auxiliando e melhorando a vida das pessoas por meio da aplicação em reconhecimento facial, carros inteligentes, segurança de dados na internet, *chatbots*, entre outros, mas também exigindo maior compreensão do desenvolvimento e utilização de tais tecnologias por parte da população. Dessa forma, é importante que o ensino da computação que inclua conceitos de IA seja popularizado para o público escolar. A partir do ensino dos conceitos básicos, ideias chave e formas de programação de IA e computação, as crianças e jovens poderão desenvolver soluções para problemas em diferentes áreas de atuação. Mesmo que atualmente existam várias iniciativas para ensinar computação no Ensino Médio, incluindo, por exemplo, a programação de apps com *App Inventor*, praticamente não existem unidades instrucionais que abranjam de forma mais completa o ensino de Inteligência Artificial. Nesse contexto, este trabalho visa o desenvolvimento de uma unidade instrucional para ensinar programação e Inteligência Artificial criando apps inteligentes alinhadas aos currículos de referência no Ensino Médio. O objetivo desta unidade instrucional é ensinar o processo de desenvolvimento de um modelo de *Deep Learning* e utilizá-lo em um aplicativo desenvolvido pelo aluno. O intuito é que os resultados do presente trabalho possam ser utilizados no contexto de escolas básicas no Brasil por alunos do Ensino Médio com o objetivo de popularizar a computação e IA. Espera-se também que isso motive o interesse e a busca por maiores aprendizados sobre computação e IA, incentivando também a escolha por carreiras profissionais nesta área.

Palavras-chave: computação; programação; inteligência artificial; unidade instrucional; ensino médio; app inventor; design thinking.

ABSTRACT

Artificial Intelligence (AI) technology is becoming increasingly influential today, helping and improving people's lives through its application in facial recognition, autonomous cars, chatbots, among others, but also demanding greater understanding of the development and use of such technologies by the population. Thus, it is important that the teaching of computing that includes AI concepts is popularized already at school. Learning basic concepts, key ideas and ways of programming AI and computing, children and young people will be able to develop solutions to problems in different areas. Even though there are currently several initiatives to teach computing in High School, including, for example, app programming with App Inventor, there are basically no instructional units that more fully cover the teaching of Artificial Intelligence. In this context, this work aims to develop a course to teach programming and Artificial Intelligence by creating smart apps in High School aligned with reference curricula. The purpose of this course is to teach the process of developing a Deep Learning model and using it in a mobile application developed by the student. It is expected that this will increase interest and motivate further learning about computing and AI, as well as encouraging the choice of professional careers in this area.

Keywords: computing; programming; artificial intelligence; instructional unit; high school; app inventor; design thinking

LISTA DE FIGURAS

Figura 1 - Visão geral dos conhecimentos da área de Computação segundo a SBC	
18	
Figura 2 - Visão geral das 5 grandes ideias em IA da AI4K12	20
Figura 3 - Janela Editor de Blocos do App Inventor	24
Figura 4 - Janela Designer do App Inventor	24
Figura 5 - Use-Modify-Create	25
Figura 6 - Design Thinking	26
Figura 7 - Áreas de pesquisa da Inteligência Artificial	28
Figura 8 - Estrutura geral de uma rede	29
Figura 9 - Etapas do desenvolvimento do modelo de Machine Learning	31
Figura 10 - Interface do VULCAN no Google Colab	33
Figura 11 - Quantidade de estudantes que apresentam computadores no domicílio com acesso à internet	45
Figura 12 - É aedes-aegypti	55
Figura 13 - Não é aedes-aegypti	55
Figura 14 - Não é mosquito	55
Figura 15 - Sketchs do app “Xô Dengue”	57
Figura 16 - Wireframe do app “Xô Dengue”	58
Figura 17 - Protótipo de alta fidelidade do app “Xô Dengue”	58
Figura 18 - Exemplo do módulo Avaliação de conceitos de algoritmos e programação no CodeMaster	61
Figura 19 - Exemplo do módulo Avaliação do design visual no CodeMaster	62

LISTA DE QUADROS

Quadro 1 - Objetivos de aprendizagem do Framework CSTA K-12 CS relacionado a Inteligência Artificial	19
Quadro 2 - Objetivos de aprendizado voltado a Aprendizagem	21
Quadro 3 - Categorias de algoritmos de ML	22
Quadro 4 - Frameworks predominantes mundialmente	30
Quadro 5 - Termos de busca e sinônimos	35
Quadro 6 - Strings de busca por base	35
Quadro 7 - Resultado da busca	36
Quadro 8 - Informações extraídas das unidades instrucionais encontrados	36
Quadro 9 - Unidades instrucionais encontradas	38
Quadro 10 - Visão geral sobre as competências ML ensinadas nas unidades instrucionais	39
Quadro 11 - Visão geral sobre as características das unidades instrucionais	40
Quadro 12 - Visão geral das características de contexto da unidade instrucional	41
Quadro 13 - Método de avaliação das unidades instrucionais	42
Quadro 14 - Visão geral sobre as avaliações das unidades instrucionais	43
Quadro 15 - Objetivos de aprendizagem do curso (todos os objetivos marcados em cinza se referem a uma parte futura do curso tendo como pré-requisito a disponibilidade de uma extensão para App Inventor)	47
Quadro 16 - Plano de ensino do curso	49
Quadro 17 - Exemplo de slides do curso	50
Quadro 18 - Exemplo de vídeos do curso	52
Quadro 19 - Caderno exemplo	53
Quadro 20 - Caderno para documentação dos artefatos criados pelo estudante	54
Quadro 21 - Objetivo do modelo de ML	54
Quadro 22 - Conjunto de dados	55
Quadro 23 - Treinamento	56
Quadro 24 - Ferramentas utilizadas no curso	58
Quadro 25 - Rubrica de avaliação de aprendizagem com base no desempenho	60

LISTA DE ABREVIATURAS E SIGLAS

ACM - *Association for Computing Machinery*

CSTA - *Computer Science Teachers Association*

IA - *Inteligência Artificial*

IEEE - *Institute of Electrical and Electronics Engineers*

MIT - *Massachusetts Institute of Technology*

ML - *Machine Learning*

SBC - *Sociedade Brasileira de Computação*

SUMÁRIO

1 INTRODUÇÃO	11
1.1 Objetivos	13
1.2 Metodologia de pesquisa e trabalho	14
1.3 Estrutura do documento	15
2 FUNDAMENTAÇÃO TEÓRICA	16
2.1 Ensino de Inteligência Artificial na Educação Básica	16
2.2 App Inventor	23
2.2.1 Ensino de computação por meio de desenvolvimento de apps	25
2.3 Deep Learning	27
2.3.1 Frameworks e bibliotecas de Deep Learning	30
2.3.2 Processo de desenvolvimento de modelos de Deep Learning	31
2.3.3 Ambiente de desenvolvimento de modelo de ML	32
3 ESTADO DA ARTE	34
3.1 Definição do Protocolo de Revisão	34
3.2 Execução da Busca	36
3.3 Extração das Informações	36
3.4 Análise dos Resultados	37
3.4.1 Quais unidades instrucionais existem?	38
3.4.2 Quais competências ML são ensinadas nas unidades instrucionais?	38
3.4.3 Quais são as características instrucionais das unidades instrucionais?	40
3.4.4 Quais são as características de contexto das unidades instrucionais?	41
3.4.5 Como foram desenvolvidas e avaliadas as unidades instrucionais?	41
3.4.6 Como foi avaliada a qualidade das unidades instrucionais?	42
3.5. Discussão	43
4 CURSO “DESENVOLVA SEU APLICATIVO INTELIGENTE PARA CLASSIFICAÇÃO DE IMAGENS USANDO VULCAN”	45
4.1. Análise do Contexto	45
4.1.1 Objetivos de Aprendizagem	46
4.2. Projeto do Curso	48
4.2.1 Contexto de Aplicação	48
4.2.2 Plano de ensino	48
4.2.3 Desenvolvimento do Material Didático	50
4.2.4 Ferramentas utilizadas no curso	58
4.2.5 Avaliação da Aprendizagem	59
5 CONCLUSÃO	63

REFERÊNCIAS

64

APÊNDICE A - ARTIGO NO FORMATO SBC

68

1 INTRODUÇÃO

Com o avanço tecnológico ocorrido principalmente no final do século XX e no início do século XXI, contando com tecnologias voltadas para o uso de Inteligência Artificial (IA) que vêm tornando-se mais presentes no dia a dia na sociedade em geral, surge a necessidade do ensino de IA no currículo educacional (YU; CHEN, 2018). Sendo uma linha de pesquisa da computação muito utilizada na automação de processos com capacidade de processamento de grande quantidade de dados, foi aplicada de várias formas, como o reconhecimento facial, carros inteligentes, segurança de dados na internet, *chatbots*, reconhecimento de padrões, *e-commerce* e outros (RUSSELL; NORVIG, 2015). Dessa forma, é importante compreender seus conceitos básicos e o impacto que causa na sociedade, portanto, o ensino de IA desde os primeiros anos escolares às crianças e jovens possibilita maior capacidade de desenvolvimento de soluções de problemas (CSTA, 2019).

A partir do aprendizado de computação, o estudante desenvolve raciocínio lógico, pensamento algorítmico, noções de design, compreensão de Inteligência Artificial e uma estruturada forma de resolução de problemas, conceitos e habilidades que poderiam ser utilizadas além das salas de aula (SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, 2017). Essas competências também auxiliam ao aluno a capacidade de desenvolver sistemas inteligentes que aprendem a partir de modelos já criados ou de desenvolver seus próprios modelos, implementar, testar e implantar uma solução, contornando as restrições do mundo real, e aplicando estas habilidades em diversos contextos (CSTA, 2019). Este cenário fortalece a necessidade de oferecer um ensino que envolva diversas competências de computação, de forma que computadores sejam vistos como um meio acessível de expressar ideias e expor a criatividade (CAMBRAIA; SCAICO, 2013).

Atualmente existem diversas iniciativas mundiais cujo foco está no ensino de computação explorando diversas maneiras de ensino (MIT, 2022c), focando tipicamente no ensino de computação via programação, usando ambientes de programação visual baseada em blocos como *App Inventor* (MIT, 2022a). O *App Inventor* permite a qualquer pessoa começar a programar e construir aplicativos completos para dispositivos Android (MIT, 2022a). Utilizando o *App Inventor*, os alunos aprendem a pensar de maneira criativa, a trabalhar de forma colaborativa e a pensar de forma sistemática na solução de problemas. Já existem diversas unidades

instrucionais voltadas ao ensino de programação de apps com App Inventor (PATTON; TISSENBAUM; HARUNANI, 2019). A maioria está voltada ao ensino de programação de um determinado tipo de app por meio de vídeos explicativos (MIT, 2022b) ou de tutoriais online (MIT, 2022d) ou por oficinas e/ou cursos de curta duração. Essas unidades instrucionais basicamente enfocam somente no ensino de pensamento computacional abordando conceitos básicos de programação como condicionais, *strings*, laços, etc. (GROVER; PEA, 2013). Alguns poucos abordam também conceitos de Engenharia de Software (PINHEIRO; GRESSE VON WANGENHEIM; MISSFELDT FILHO, 2018), Design de Interface (FERREIRA *et al.*, 2019) e/ou até IoT/Robótica (GRESSE VON WANGENHEIM *et al.*, 2017).

Observa-se que mesmo com a crescente importância e interesse na área de Inteligência Artificial, existem poucas unidades instrucionais voltadas ao ensino de conceitos básicos de IA na educação básica (MARQUES *et al.*, 2020) e, dessa forma, percebe-se uma necessidade de inserir de forma mais sistemática o ensino destes conceitos (SBC, 2017; TOURETZKY *et al.*, 2019).

Sendo um subcampo da Inteligência Artificial, o *Machine Learning* (ML) estuda o reconhecimento de padrões e teoria do aprendizado computacional. O ML explora o estudo e construção de algoritmos que possuem a capacidade de aprender de seus erros e fazer previsões sobre dados. Esses algoritmos constroem um modelo a partir de inserções de amostras a fim de fazer previsões ou decisões guiadas pelos dados (RUSSELL; NORVIG, 2015). O *Deep Learning* é um ramo do *Machine Learning*, que fornece um processo de aprendizado automático para a máquina utilizando Redes Neurais para o reconhecimento de fala, visão computacional e processamento de linguagem natural, focando na resolução de problemas, que valoriza e explora várias perspectivas de um problema (RUSSELL; NORVIG, 2015).

Segundo as diretrizes de currículo AI4K12 ML, espera-se que os estudantes participantes aprendam e/ou aprofundem seus conhecimentos sobre ML, aprendendo conceitos básicos sobre classificação de imagens com Redes Neurais, seguindo as diretrizes do ensino de IA da AI4K-12 (TOURETZKY *et al.*, 2019).

Tipicamente os conceitos de computação e de IA/ML são ensinados utilizando metodologias ativas seguindo o ciclo de aprendizagem *use-modify-create* (LEE, 2011). Esse ciclo se inicia no *use*, que inclui a utilização de algo que faz parte do contexto em um *app* com *ML*, possibilitando a compreensão de conceitos básicos. O próximo estágio é *modify*, que se refere a modificar um *app* afetando a parte de *ML*. E

o último estágio é o *create*, em que o estudante vai criar um *app* com *ML*. Combinando com a estratégia de ação computacional, incentiva o estudante a desenvolver *app* que solucione problemas do seu entorno, tanto na sua vida ou na sua comunidade, e mostra para o estudante que seu aprendizado é aplicável e útil (PATTON; TISSENBAUM; HARUNANI, 2019).

Contudo, observa-se que poucas unidades instrucionais voltadas ao ensino de IA/ML na educação básica apresentam o estágio *create*, deixando de ensinar ao aluno a criação do seu próprio modelo de ML e a implantação deste em um sistema de software, como um aplicativo móvel, por exemplo (MARQUES *et al.*, 2020).

Assim, esse trabalho visa desenvolver um curso para ensinar, conceitos de *Machine Learning* e habilidades do século XXI de forma dinâmica e motivadora para popularizar este conhecimento já no Ensino Médio. Espera-se que o curso desenvolvido a partir deste trabalho possa ser aplicado amplamente em escolas do Brasil, popularizando não só os conceitos de programação, mas também os conceitos de Inteligência Artificial e *Machine Learning* como elementos do aprendizado de computação.

1.1 Objetivos

O objetivo geral deste trabalho é desenvolver um curso online para estudantes do Ensino Médio que os ensine a criar seus próprios apps “inteligentes” para classificação de imagens usando *Machine Learning*. O curso está alinhado ao currículo de referência do ensino de computação da SBC (2017) e, principalmente, às diretrizes referentes ao ensino de IA, conforme proposto pelo AI4K-12 (TOURETZKY *et al.*, 2019; LONG; MAGERKO, 2020). Neste nível educacional é utilizado *fast.ai/jupyter/colab* para desenvolver o modelo de ML e a implantação do modelo criado no *App Inventor*. Este curso visa o desenvolvimento de aplicativos inteligentes no estágio *create* usando *Design Thinking* para criar soluções relacionadas a questões da comunidade.

Para isso, definiu-se os seguintes objetivos específicos:

- O1. Sintetizar a fundamentação teórica relativa ao ensino de ML no Ensino Médio, *Machine Learning* e desenvolvimento de apps com *Design Thinking*;
- O2. Levantar o estado da arte em relação a cursos similares;

O3. Analisar o contexto e definir o design do curso (seleção e sequenciamento do conteúdo, estratégias de ensino, etc.)

O4. Desenvolver o material didático para o curso.

1.2 Metodologia de pesquisa e trabalho

Para concluir o desenvolvimento dos objetivos definidos, é necessário efetuar uma combinação de métodos de pesquisa de acordo com os respectivos objetivos específicos a serem alcançados. Dessa forma, foram adotadas as seguintes etapas:

Etapa 1 - Fundamentação teórica: Nesta etapa são abordados os conceitos de ensino de computação no Ensino Médio, conceitos básicos de aprendizagem e ensino, além de conceitos de Inteligência Artificial, da área de *Machine Learning* e sobre o ambiente de programação visual baseado em blocos do *App Inventor usando Design Thinking*. Este estudo é feito por meio de uma análise e síntese da literatura. Desta forma, esta etapa pode ser subdividida em:

Atividade 1.1: Sintetizar conceitos de ensino de *Machine Learning* no Ensino Médio;

Atividade 1.2: Sintetizar conceitos de *Machine Learning (Deep Learning)*;

Atividade 1.3: Sintetizar conceitos de desenvolvimento de *apps* com *App Inventor* usando *design thinking*.

Etapa 2 - Levantamento do estado da arte: Nesta etapa é realizado o estudo dos processos proposto por Petersen, Vakkalanka e Kuzniarz (2015) para identificar, analisar e sistematizar cursos online e estratégias de ensino, atualmente sendo utilizadas e voltadas ao ensino de computação em escolas. Desta forma esta etapa pode ser subdividida em:

Atividade 2.1: Definir o protocolo de busca;

Atividade 2.2: Executar a busca;

Atividade 2.3: Extrair e analisar as informações.

Etapa 3 - Design do curso: Nesta etapa é desenvolvida a definição do design do curso online, seguindo a metodologia de design instrucional ADDIE (BRANCH, 2009), e pode ser subdividida em:

Atividade 3.1: Analisar o contexto em termos de necessidades, perfil dos aprendizes e instrutores e de ambiente em escolas brasileiras;

Atividade 3.2: Definir e sequenciar o conteúdo do curso e definir uma estratégia de aprendizagem, criando o plano de ensino.

Etapa 4 - Desenvolvimento do curso: nesta etapa é realizado o desenvolvimento do material didático para a aplicação da unidade instrucional. Esses materiais didáticos incluem atividades interativas usando h5p.org a partir de slides, um Caderno com os artefatos que serão desenvolvidos durante o curso e Caderno exemplo. Desta forma, esta etapa pode ser subdividida em:

Atividade 4.1: Desenvolver o material didático;

Atividade 4.2: Desenvolver avaliações da aprendizagem do aluno.

1.3 Estrutura do documento

Para melhor compreensão do desenvolvimento deste trabalho e visando o alcance pleno dos objetivos propostos, a estrutura está dividida em capítulos complementares entre si.

No capítulo 2 é apresentada a fundamentação teórica do conhecimento necessário para o desenvolvimento do curso. A seguir, o capítulo 3 apresenta o estado da arte sobre as unidades instrucionais existentes referente ao ensino de *Machine Learning* utilizando *App Inventor* no Ensino Médio. No capítulo 4, apresenta-se o desenvolvimento do curso proposto. Por fim, o capítulo 5 contém as considerações finais deste trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentados os conceitos básicos de ensino e aprendizagem, e, especificamente, do ensino de computação. São apresentados também conceitos básicos referentes ao *Machine Learning* e o desenvolvimento de aplicativos para smartphones utilizando o *App Inventor*.

2.1 Ensino de Inteligência Artificial na Educação Básica

Conforme a Lei nº 9.394 de Diretrizes e Bases da Educação Nacional (BRASIL, 1996, *online*), a educação básica tem como objetivo “desenvolver o educando, assegurar-lhe a formação comum indispensável para o exercício da cidadania e fornecer-lhe meios para progredir no trabalho e em estudos posteriores”.

O Ensino Médio, junto com a educação infantil e o ensino fundamental, faz parte do ciclo de educação básica no Brasil. O Ensino Médio tem duração de três anos, organizada para que o jovem inicie os estudos aos quinze anos de idade e conclua aos dezessete anos (MEC, 2013). Com o aprendizado dos conteúdos escolares, o jovem estará apto para continuar com o seu desenvolvimento escolar, seguindo para formação no ensino técnico ou superior, de forma autônoma e crítica. Além disso, a educação permite ao jovem o desenvolvimento de uma base como cidadão e a compreensão da cidadania como participação social e política, posicionando-se de maneira crítica, responsável e construtiva nas diferentes situações sociais, dentre vários outros objetivos (BRASIL, 2013).

A Base Nacional Comum Curricular (BNCC) estruturou o currículo comum da base de ensino nacional, e em relação ao Ensino Médio, divide-se nas seguintes competências:

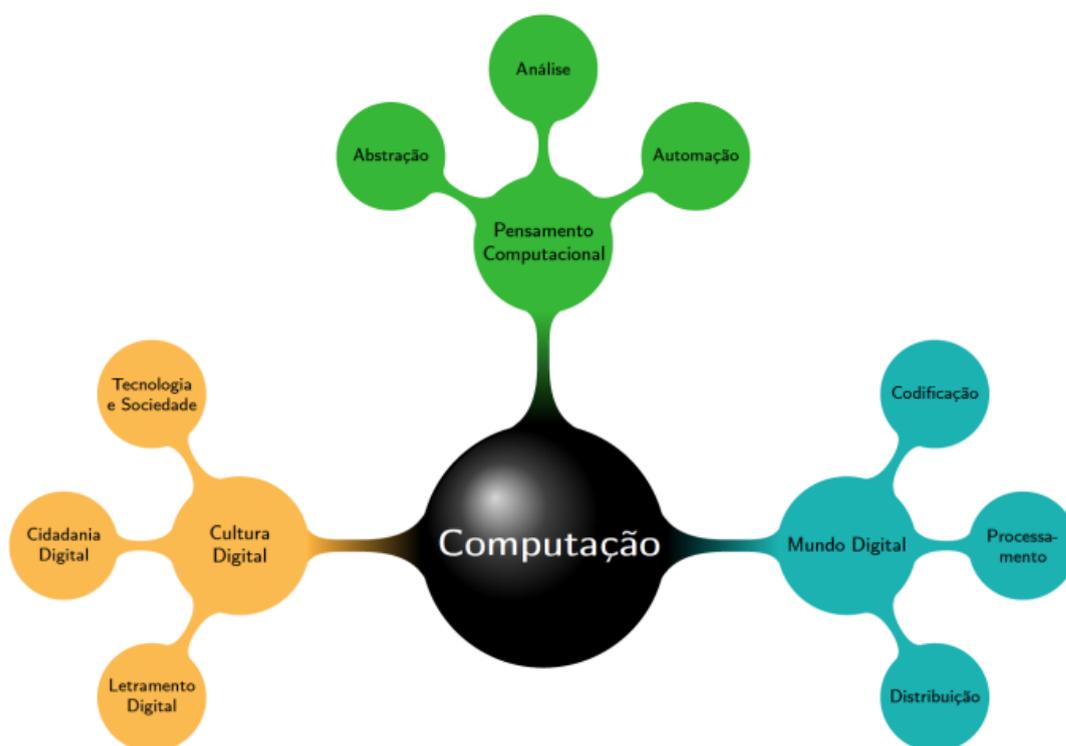
- Linguagens, representada a partir dos componentes:
 - Língua Portuguesa: Nessa competência aprofunda-se o conhecimento adquirido na etapa anterior do ensino escolar focando nos campos temáticos de vida pessoal, artístico-literário, práticas de estudo e pesquisa, jornalístico-midiático e atuação na vida pública.
 - Artes: Compreende aprofundamento na pesquisa e no desenvolvimento de processos de criação autorais nas linguagens das artes visuais, do audiovisual, da dança, do teatro, das artes circenses e da música.

- Educação Física: Busca a abordagem de novos jogos e brincadeiras, esportes, danças, lutas, ginásticas e práticas corporais de aventura.
- Língua Inglesa: O ensino da língua inglesa no Ensino Médio visa permitir aos estudantes explorar a presença da multiplicidade de usos da língua inglesa na cultura digital, nas culturas juvenis e em estudos e pesquisas, como também ampliar suas perspectivas em relação à sua vida pessoal e profissional.
- Matemática: O ensino da matemática no Ensino Médio tem como um dos objetivos a continuação do aprendizado desenvolvido da etapa anterior, focando em construir uma visão integrada da Matemática, aplicada à realidade em diferentes contextos.
- Ciências da Natureza: No ensino médio, a área de conhecimento em ciências da natureza é dividida nas disciplinas biologia, física e química, para o ensino do conhecimento nas áreas de Matéria e energia, Vida e evolução, Terra e Universo. Tendo com objetivo aprofundar os conhecimentos da etapa anterior e desenvolver uma base de conhecimentos que permite investigar, analisar e discutir situações-problema que emergem de diferentes contextos socioculturais, além de compreender e interpretar leis, teorias e modelos, aplicando-os na resolução de problemas individuais, sociais e ambientais.
- Ciências Humanas: Essa área é dividida nas disciplinas história, geografia, filosofia e sociologia para o ensino do conhecimento nas áreas de Tempo e Espaço, Território e Fronteira, Indivíduo, Natureza, Sociedade, Cultura e Ética, Política e Trabalho. Com objetivo de ampliar e aprofundar os aprendizados já iniciados no ensino fundamental. Além de ajudar a orientar na formação da base das ideias de justiça, solidariedade, autonomia, liberdade de pensamento e de escolha, compreensão e o reconhecimento das diferenças, o respeito aos direitos humanos e à interculturalidade, e o combate aos preconceitos de qualquer natureza.

Na BNCC também foi incluído o ensino da computação, a partir das três grandes diretrizes que são Pensamento Computacional, Mundo Digital e Cultura digital dentro da área de conhecimento da matemática (BRASIL, 2018). Consolidando o conhecimento da etapa anterior e ampliando novos conhecimentos mais complexos, que exigem maior reflexão e abstração, integrando mais a visão matemática.

Nesse contexto, a Sociedade Brasileira de Computação (SBC) detalhou diretrizes para ensino de Computação na Educação Básica. Estas diretrizes são definidas em três eixos, conforme Figura 1.

Figura 1 - Visão geral dos conhecimentos da área de Computação segundo a SBC



Fonte: SOCIEDADE BRASILEIRA DE COMPUTAÇÃO (2017)

O eixo de Pensamento Computacional trata sobre a capacidade de compreender, definir, modelar, comparar, solucionar, automatizar e analisar problemas e soluções de forma organizada e sistemática, através da construção de algoritmos utilizando abstrações e técnicas necessárias para a descrição e análise de informações (dados) e processos, automação de soluções.

Já o eixo Mundo Digital foi criado para armazenar, processar e distribuir informação. Para desenvolver suas habilidades e conseguir utilizar a tecnologia digital da melhor maneira possível, é necessário que cada pessoa compreenda o “mundo digital” e “mundo real” da mesma forma. O mundo digital tem 3 pilares principais: codificação, processamento e distribuição. A codificação diz respeito à representação do mundo real no mundo digital, o processamento se refere à capacidade de processamento dos dados codificados no mundo digital e o pilar de

distribuição compreende a capacidade de distribuição de informação no mundo digital. Esta capacidade é fator fundamental para tamanho impacto do mundo digital (SBC, 2017).

O eixo denominado Cultura Digital reflete a necessidade de um letramento em tecnologias digitais, necessário para o estabelecimento da comunicação e expressão por meio do Mundo Digital. Também faz parte da Cultura Digital uma análise dos novos padrões de comportamento e novos questionamentos morais e éticos na sociedade que surgiram em decorrência do Mundo Digital. A Cultura Digital compreende as relações interdisciplinares da Computação com outras áreas do conhecimento, buscando promover a fluência no uso do conhecimento computacional para expressão de soluções e manifestações culturais de forma contextualizada e crítica.

As diretrizes de currículo da SBC (2017) também abordam a IA junto com a robótica nos eixos Pensamento Computacional e Mundo Digital. Nesse nível educacional, se foca na compreensão dos fundamentos da IA abordando uma discussão sobre o que é o Homem e o que é a Máquina, quais as similaridades e diferenças e questões éticas envolvidas na IA (SBC, 2017).

Similar a um dos principais *frameworks* de currículo internacional, o *Framework K–12 Computer Science* da CSTA (2019), nesse estágio escolar, o *framework* prevê abordar o ensino de competências relacionados a cinco objetivos de aprendizagem relacionados à IA conforme apresentado no quadro 1.

Quadro 1 - Objetivos de aprendizagem do Framework CSTA K-12 CS relacionado a Inteligência Artificial

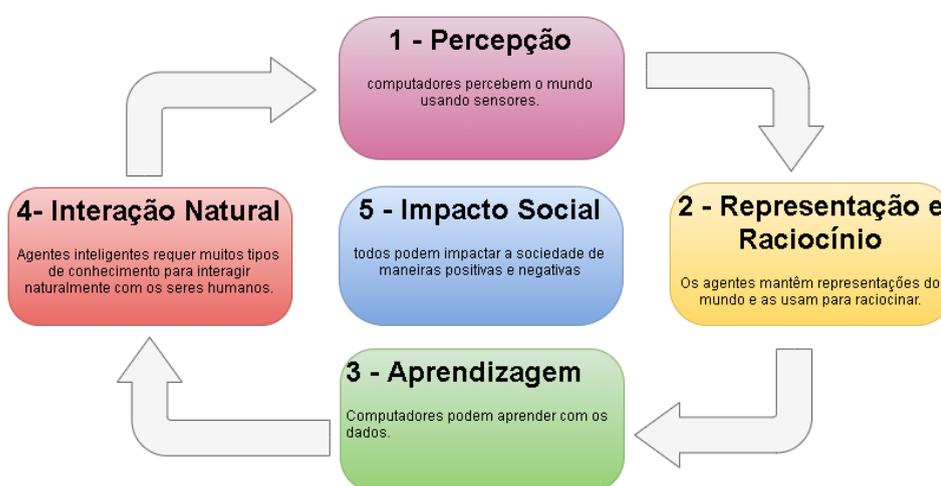
Framework CSTA K–12 CS		
Identificador	Ano escolar	Objetivo de aprendizagem
3A-AP13	14-16	Crie protótipos que usam algoritmos para resolver problemas computacionais, aproveitando o conhecimento prévio do aluno e seus interesses pessoais.
3B-AP-08	16-18	Descreva como a inteligência artificial impulsiona muitos softwares e sistemas físicos.
3B-AP-09	16-18	Implemente um algoritmo de inteligência artificial para jogar um jogo contra um oponente humano ou resolver um problema.
3B-AP-10	16-18	Use e adapte algoritmos clássicos para resolver problemas computacionais.

3B-AP-11	16-18	Avalie os algoritmos em termos de eficiência, correção e clareza.
----------	-------	---

Fonte: CSTA (2019)

O AI4K12.org coordenado pelo MIT está desenvolvendo um guia curricular para ensino de IA na educação básica em alinhamento com o CSTA, tanto no Ensino Médio quanto no ensino fundamental. Nesse sentido, a figura 2 apresenta uma proposta de visão geral das ideias que buscam tornar os alunos capazes de avaliar e aplicar a tecnologia de informação.

Figura 2 - Visão geral das 5 grandes ideias em IA da AI4K12



Fonte: TOURETZKY *et al.* (2019)

Assim, conforme Touretzky *et al.* (2019), a estrutura do currículo da AI4K12 está dividida em cinco grandes ideias em IA:

1. **Percepção:** Com o uso de sensores, a percepção é utilizada para extrair informação nos sinais sensoriais, assim recebendo a quantidade de informação necessária para ser usada na IA. Alguns exemplos de ideias que podem ser utilizadas na percepção:
 - Reconhecimento de fala
 - Visão computacional
 - Outras formas de percepção (por exemplo, sonar, reconhecimento de música).
2. **Representação e Raciocínio:** Na IA são usados agentes para representação do mundo e raciocínio. As representações são construídas usando estruturas de dados, e essas representações suportam raciocínio algorítmicos que

derivam novas informações do que já é conhecido. Enquanto na IA, os agentes podem raciocinar sobre os problemas, mas eles não são capazes de pensar da mesma maneira que humanos fazem.

3. **Aprendizagem:** Os computadores podem aprender com os dados. *Machine Learning* é um tipo de inferência estatística que usa padrões de dados nos dados. Muitos avanços na área da IA foram realizados pelo uso de algoritmos de aprendizado que criam novas representações. Para que a abordagem tenha sucesso, são necessárias enormes quantidades de dados e treinamento da rede.
4. **Interação Natural:** Agentes inteligentes requerem muitos tipos de conhecimento para interagir naturalmente com humanos. Os agentes devem poder conversar em línguas humanas, reconhecer expressões e emoções e recorrer a conhecimento de cultura e convenções sociais para inferir intenções do comportamento observado.
5. **Impacto Social:** A IA pode impactar a sociedade tanto em aspectos positivos quanto de maneiras negativas. Assim, é importante discutir os impactos e efeitos que a IA tem em nossa sociedade e desenvolver critérios para o desenho ético e implantação de aplicativos baseados em inteligência artificial sistemas.

Observa-se, então, que a grande ideia 3 se refere ao *Machine Learning*. No quadro 2, apresenta-se os objetivos de aprendizado específicos a esta ideia de aprendizagem conforme seu grau curricular segundo AI4K12.

Quadro 2 - Objetivos de aprendizado voltados à ideia de Aprendizagem

Grau curricular	Descrição dos conhecimentos ensinados
Grau K - K2. (5-8 anos)	<ul style="list-style-type: none"> ● Aprender a reconhecer padrões em dados com atividades. ● Aprender a utilizar um classificador que reconheça desenhos.
Grau K3-K5. (8-11 anos)	<ul style="list-style-type: none"> ● Descreve e compara as três diferentes abordagens de aprendizado de máquina. ● Modificar um projeto interativo de aprendizado de máquina treinando seu modelo. ● Descrever como algoritmos e aprendizado de máquina podem exibir vies.
Grau K6-K8 (11-14 anos)	<ul style="list-style-type: none"> ● Identificar o viés de um conjunto de dados de treinamento e ampliar o conjunto de treinamento para lidar com esse viés.

	<ul style="list-style-type: none"> • Simular manualmente o treinamento de uma simples rede neural.
Grau K9-K12 (14-18 anos)	<ul style="list-style-type: none"> • Treine uma rede neural (1-3 camadas) p.ex. <i>TensorFlow Playground</i>. • Rastrear e experimentar um simples algoritmo de ML.

Fonte: TOURETZKY *et al.* (2019)

Para atingir esses objetivos de aprendizagem a proposta de currículo da AI4K12 propõe a abordagem dos seguintes tópicos em unidades instrucionais:

- Aprendizagem de máquina: A aprendizagem de máquina é um subcampo da Inteligência Artificial, que trabalha com a premissa que as máquinas podem aprender sozinhas ao terem acesso a grandes volumes de dados. No quadro 3, estão demonstradas as categorias dos algoritmos de *ML*.

Quadro 3 - Categorias de algoritmos de ML

Categoria de algoritmo de ML	Definição
Algoritmos supervisionados	O conjunto de dados de treinamento tem entradas, bem como a saída desejada. Durante a sessão de treinamento, o modelo se ajusta às suas variáveis para mapear as entradas para a saída desejada.
Algoritmos não supervisionados	Não tem resultado pré-definido para o modelo. Os algoritmos agrupam o conjunto de dados para diferentes grupos.
Algoritmos de reforço	Esses algoritmos são treinados para tomar decisões. Então, com base nessas decisões, o algoritmo irá treinar com base no sucesso / erro das saídas do treinamento. E assim, possivelmente, o algoritmo de experiência poderá dar boas previsões.

Fonte: Autora (2020)

- Tipos de algoritmos de aprendizado: Algoritmo é um conjunto de regras e procedimentos/operações lógicas definidas que levam à solução de um problema em um número finito de etapas. Existem vários algoritmos voltados a *ML*. A partir das categorias de algoritmos de *ML* são apresentados alguns algoritmos de *ML* como: *Decision Tree*, *neural network*, *instance-based learning*, *Linear Regression*, *SVM (Support Vector Machine)*, *KNN (K-Nearest Neighbors)*, *Logistic Regression*, *Decision Tree*, *K-Means*, *Random Forest*, *Naive Bayes*, *Dimensional Reduction Algorithms*, *Gradient Boosting Algorithms*.

- Fundamentos de Redes Neurais: Redes Neurais são técnicas que tentam simular o funcionamento das Redes Neurais Biológicas do Corpo Humano para resolver problemas computacionais relacionados com Inteligência Artificial. Para as

Redes Neurais solucionarem o problema desejado, é necessário treinar a rede e para isso existem três categorias de algoritmos de treinamento: algoritmos supervisionados, algoritmos não supervisionados e algoritmos de reforço. O treinamento da rede pode ter várias camadas de treinamento, elas variam conforme a complexidade do problema. Redes Neurais funcionam a partir de nós e camadas, em que os nós são os locais que fazem os cálculos matemáticos para decidir se a informação deve ser enviada para o próximo nó, e as camadas são um conjunto de nós com um determinado objetivo. Quanto mais camadas, mais profunda é a rede neural. Os dados são inseridos através da camada de entrada, nas camadas ocultas ocorre o processamento dos dados através das conexões ponderadas. Os resultados são processados até chegar na camada de saída e assim tendo o resultado final.

- Preocupações éticas e carreira: Existe muita preocupação sobre a ética na área da tecnologia, sobre o que ela desenvolve ou informações que coleta. Este tópico se preocupa em discutir o que é ética e se é ético ou não desenvolver algo em específico com esse conhecimento. Além da discussão sobre as preocupações éticas, visa demonstrar em quais áreas é possível atuar com esse conhecimento, como: Cientista de Dados, Engenheiro de *Machine Learning*, Cientista de *Machine Learning*, Arquiteto de Dados entre outras carreiras. Dessa forma, amplia a visão do aluno para o que pode ser feito com esse conhecimento e quais carreiras ele pode seguir.

Em vista disso, este trabalho traz uma abordagem sobre como incluir o ensino de *Machine Learning* para os estudantes do ensino médio pensando em alinhar as diretrizes do currículo BNCC e SBC utilizando as diretrizes curriculares da AI4K12.

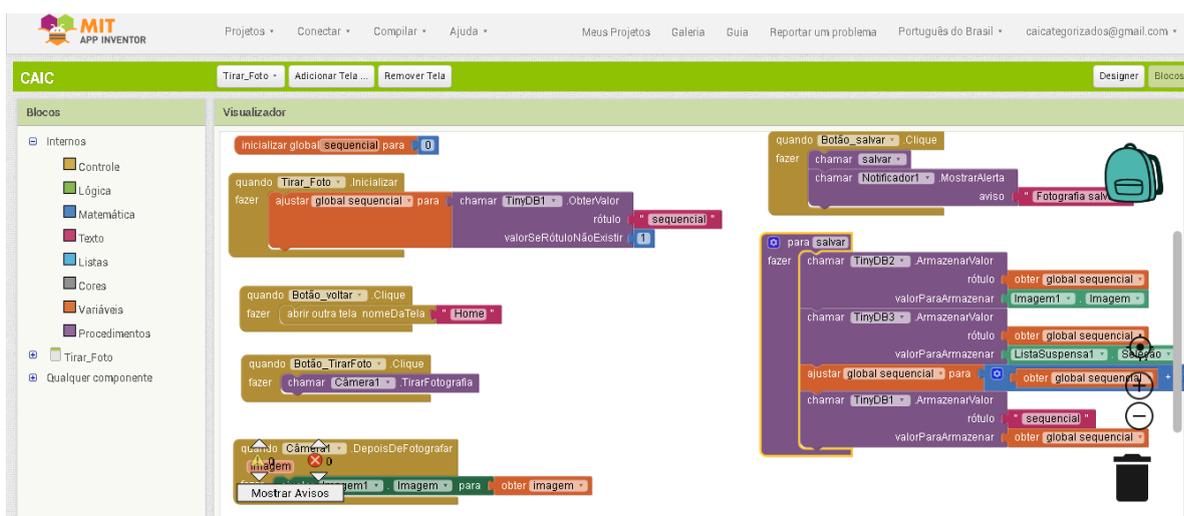
2.2 App Inventor

O *App Inventor* (MIT, 2022a) foi criado pela *Google LLC* e atualmente é mantido pela *Massachusetts Institute of Technology* (MIT). É um ambiente de programação visual que permite a usuários sem conhecimento em programação o desenvolvimento de aplicativos para *smartphone* com sistema operacional *Android*. O *App Inventor* é um ambiente de desenvolvimento de aplicativos móveis baseado em blocos.

O ambiente oferecido pelo *App Inventor* apresenta o editor de blocos, no qual o desenvolvedor programa a funcionalidade do aplicativo, e o editor de designer, utilizado para desenvolver a interface de usuário do aplicativo.

O editor de blocos fornece vários tipos de blocos quando combinados conforme a funcionalidade (Figura 3).

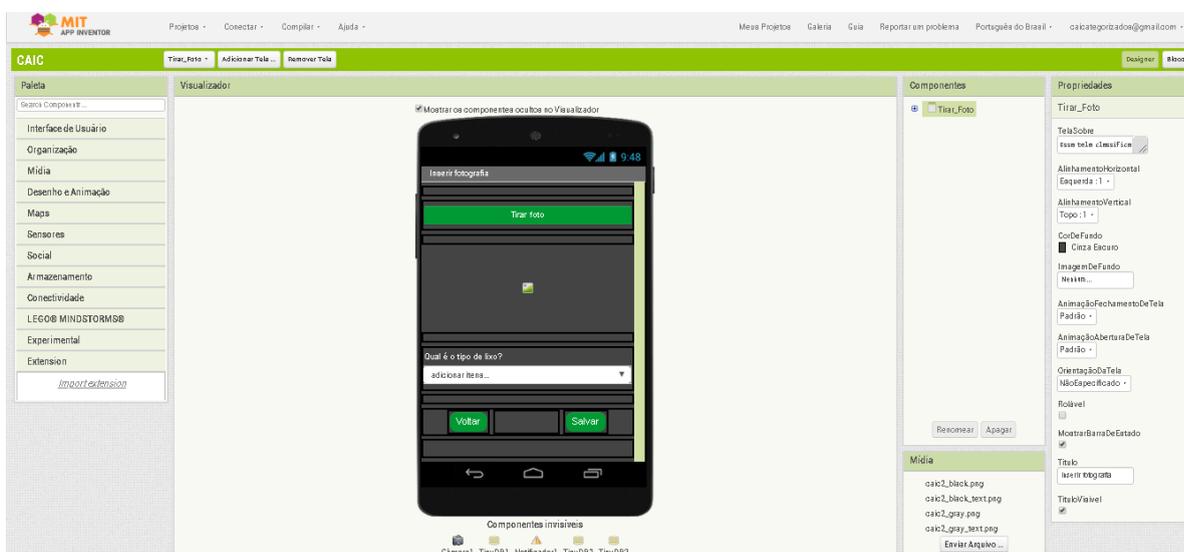
Figura 3 - Janela Editor de Blocos do App Inventor



Fonte: MIT APP INVENTOR (2019)

O editor de designer fornece vários componentes de interfaces de usuário que, quando combinados, apresentam tela de visualização prévia do aplicativo em um dispositivo móvel (Figura 4).

Figura 4 - Janela Designer do App Inventor



Fonte: MIT APP INVENTOR (2019)

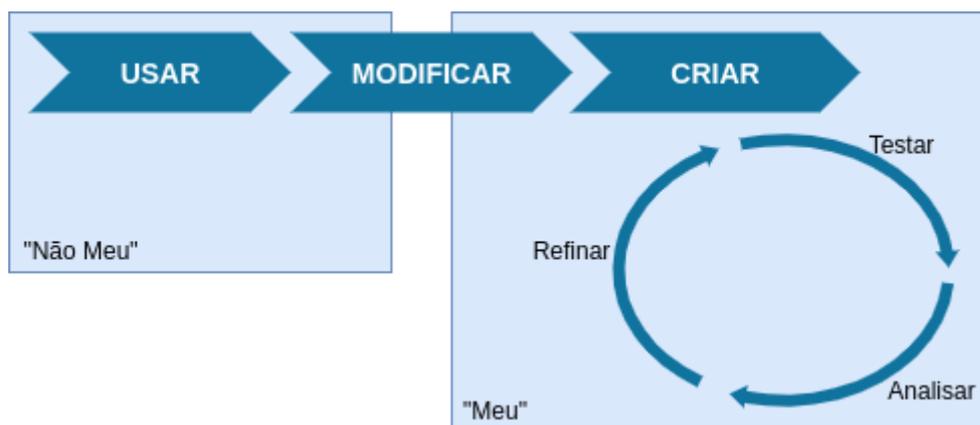
O aplicativo dispõe também de outros componentes como organizador, mídias, mapas, sensores, componentes para integração de redes sociais, entre outros. Também é possível compartilhar o aplicativo desenvolvido pelo compartilhamento do “.apk” ou exportando o código do arquivo “.aia”, que é um formato próprio do *App Inventor*.

O *App Inventor* tem uma funcionalidade que auxilia no processo de desenvolvimento, o “*Live Testing*”, que permite aos desenvolvedores testar o seu aplicativo no mesmo momento que se desenvolve. Esta funcionalidade está disponível por meio do aplicativo *MIT AI2 Companion* disponível na *Play Store*¹.

2.2.1 Ensino de computação por meio de desenvolvimento de apps

Levando os alunos ao nível de conhecimento de aplicação seguindo a taxonomia de Bloom (ANDERSON, 2001) são utilizados tipicamente metodologias ativas. O ensino da computação por meio de desenvolvimento de apps vem sendo realizada com a aplicação do ciclo *use-modify-create* (LEE, 2011), que facilita o aprendizado dos estudantes por consolidar o conhecimento e deixá-los mais seguros ao passar dos 3 estágios do ciclo: usar, modificar e criar.

Figura 5 - *Use-Modify-Create*



Fonte: LEE (2011, tradução nossa)

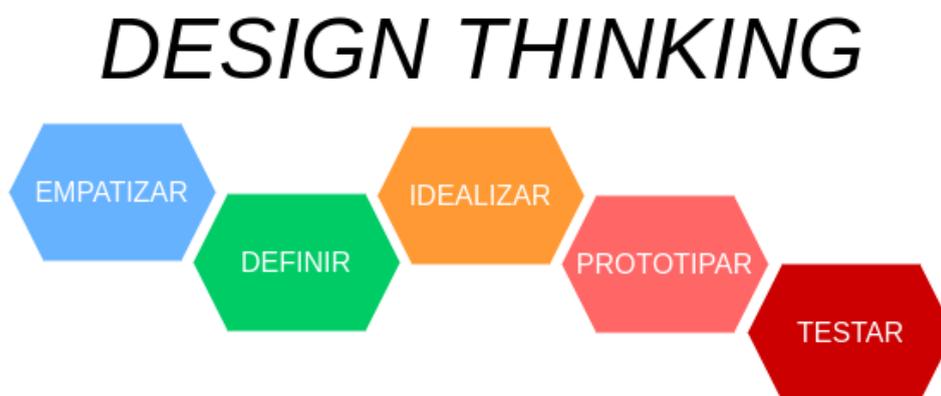
¹ Play Store é o serviço de catálogo de aplicativos e conteúdos de mídias digitais para *smartphones* e *tablets* com Sistema Operacional Android, atualmente pertence a Google LLC, segundo os Termos de Serviço da Google Play, Disponível em https://play.google.com/intl/pt_br/about/play-terms/index.html. Acesso em 10 de dezembro de 2019.

Dessa forma, Lee (2011) define o primeiro estágio do ciclo como *Use* (usar), cujo objetivo é a exploração do que será ensinado, deixando o estudante usar o programa para a compreensão de seu funcionamento. O segundo estágio é *Modify* (modificar), momento no qual os estudantes têm o conhecimento básico do aprendizado proposto, se sentindo mais confortáveis com o programa. Neste estágio são incentivados a modificar o programa, iniciando com alterações simples e evoluindo para alterações mais complexas. O último estágio desse ciclo é o *Create* (criar), no qual, a partir dos conhecimentos adquiridos nos estágios *Use* e *Modify*, o estudante passa para a fase chamada *mine* (meu), constituída por um fluxo de testar, analisar e redefinir o programa com frequência em seu desenvolvimento.

Este estágio permite também adotar a estratégia de ação computacional (TISSENBAUM *et al.*, 2019) para o desenvolvimento de apps. Essa estratégia tem como objetivo demonstrar ao estudante a aplicação e utilidade do conhecimento adquirido, por meio do ensino de computação com desenvolvimento de soluções que tenham impacto direto em suas vidas e comunidades.

Para ensinar o desenvolvimento de apps no estágio *create*, tentando assegurar a criação de apps úteis, pode ser adotado o *Design Thinking*, uma metodologia de *design* para resolver problemas (BROWN, 2008), procurando ao longo do processo *insights* mostrando maneiras novas e diferentes de enxergar os problemas e desenvolver soluções seguindo um processo, conforme apresentado na Figura 6.

Figura 6 - *Design Thinking*



Fonte: STANFORD (2016, tradução nossa)

A primeira etapa é a empatia, em que o objetivo é compreender o problema proposto, se aprofundando tanto no ambiente físico quanto no contexto e, especificamente, envolvendo a empatia para o público-alvo.

A segunda etapa é a definição, que consiste na reunião, organização e análise das informações obtidas na etapa da empatia pela ótica dos usuários para o objetivo de definir apropriadamente o problema identificado.

A etapa seguinte é a de idealizar, momento em que se começa a conceber ideias em relação ao problema identificado nas etapas anteriores. Neste momento, é possível utilizar várias técnicas para ajudar na geração de ideias, como *brainstorm*, momento empregado para estimular o pensamento livre e as divergências. Além do *brainstorm* há formas de ajudar na investigação e teste de ideias para encontrar a solução do problema. Ao final são criadas propostas de soluções que serão escolhidas uma ou mais para seguirem à etapa seguinte.

Prototipar é a quarta etapa, cujo objetivo é a criação de protótipos ou partes importantes do produto. É importante para testar as soluções pensadas na etapa anterior, podendo ser testadas pelo grupo idealizador da solução, com outras pessoas da empresa e até mesmo com grupo pequeno de pessoas externas. Assim a solução pode ser avaliada, rejeitada, aprovada, revisada e melhorada, guiando-se a partir dos resultados da experiência dos usuários.

Essa etapa compreende a prototipação do design de interface começando com *sketches*, *wireframes* até o design de alta fidelidade. Envolve também a prototipação da parte funcional do app.

A última etapa é testar o desempenho do produto em forma final, de forma ampla e completa via teste de usabilidade, por exemplo, adicionando as melhorias encontradas na etapa anterior.

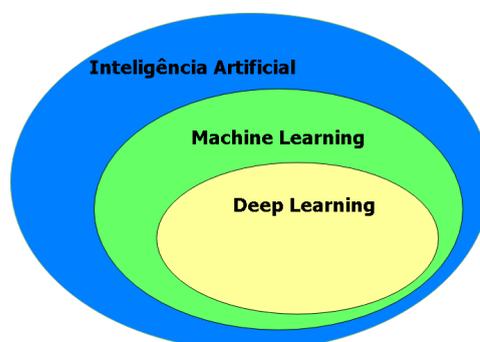
Essa metodologia permite voltar para etapas anteriores quando necessário, assim possibilitando melhorar os conhecimentos para facilitar o desenvolvimento das etapas seguintes.

2.3 Deep Learning

Inteligência Artificial (IA) é um ramo da ciência da computação com o objetivo de desenvolver a tecnologia que simula a inteligência humana utilizando alguns mecanismos ou software (GOODFELLOW *et al.*, 2016). O objetivo da inteligência

artificial é baseado na ideia de fazer o computador “pensar” como se fosse um humano.

Figura 7 -Áreas de pesquisa da Inteligência Artificial



Fonte: Autora (2019)

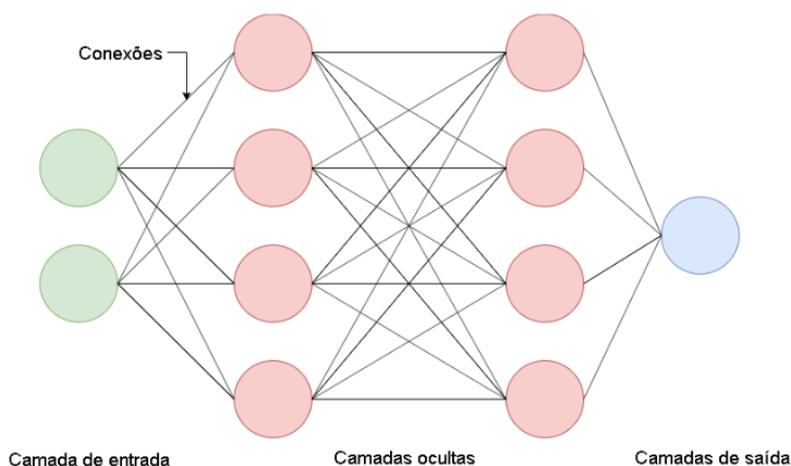
Machine Learning (ML) é uma área dentro da IA, que foca no estudo científico de algoritmos e modelos estatísticos para que os sistemas possam executar tarefas predeterminadas sem ter que utilizar instruções explícitas, tomando as decisões a partir de padrões e inferências (GOODFELLOW *et al.*, 2016). Os algoritmos utilizados na *ML* constroem um modelo matemático baseado em dados amostrais utilizados para treinar a máquina, assim sendo possível fazer previsões ou decisões sem ter comandos explícitos para a execução da tarefa (GOODFELLOW; BENGIO; COURVILLE, 2016).

Dentro da área de *ML*, *Deep Learning (DL)* é uma técnica baseada em conjunto de algoritmos que modela abstrações de alto nível de dados. São utilizadas várias camadas de processamento e treinando os computadores para realizar as tarefas a partir dos dados injetados na rede de treinamento. Desta forma reconhece padrões e se aperfeiçoa. Normalmente *DL* é utilizado para reconhecimento de fala, reconhecimento de imagem, processamento de linguagem natural, sistemas de recomendação e entre outros (GOODFELLOW; BENGIO; COURVILLE, 2016).

Uma das técnicas de *DL* é Redes Neurais Artificiais (RNA) que são modelos computacionais baseadas na estrutura neural de ser humano, que possuem capacidade de aprender com experiência, sendo capazes de fazer o aprendizado de máquina ou o reconhecimento de padrões (NICHOLSON, 2019). Assim, as RNAs simulam o comportamento de Redes Neurais biológicas. São utilizadas em diversas formas como: *perceptron*, *feed forward*, *radial basis network*, *deep feed forward*, *recurrent neural network*, *deep convolutional network* e etc (NICHOLSON, 2019).

Uma rede neural é organizada em camadas, tendo uma camada de entrada, uma de saída e camada oculta entre as camadas de entrada e saída. Essas camadas são conectadas por nós e assim formando a rede neural.

Figura 8 - Estrutura geral de uma rede



Fonte: SAS (2019)

Os nós são modelados como o comportamento de um neurônio humano. Esses nós são ativados por incentivos ou por entradas suficientes. Essa ativação se propaga através da rede, criando uma resposta ao incentivo. As conexões fazem os sinais serem transmitidos de um para o outro, sinalizando conforme via entre as camadas e são processadas durante o percurso. Quando há uma requisição ou problema para resolver, os nós executam cálculos matemáticos para saber se tem informação suficiente para mandar para o próximo nó. De maneira resumida, eles leem todos os dados e decidem onde estão as relações mais fortes. A partir dessas camadas é possível construir a rede treinando-a com os dados inseridos para emular tarefas humanas com precisão.

Em resumo, os dados são inseridos na rede neural através da camada de entrada, que se comunica com as camadas ocultas. Nessas camadas ocultas ocorre o processamento dos dados através das conexões ponderadas. Os nós combinam os dados da camada de entrada com um conjunto de coeficientes e, assim, são definidos diferentes pesos para as entradas. Os resultados são avaliados e somados, passando pela função de ativação de nó que determina extensão que o sinal vai ter, assim propagando na rede para afetar o resultado final. Na camada de saída tem o resultado final (NICHOLSON, 2019).

Para classificar a imagem são tipicamente utilizados modelos CNN (*Convolutional Neural Network*), que foi inspirado pelo padrão de conectividade entre os neurônios. Uma CNN é dividida em várias camadas pré-processando a menor demanda possível, isso possibilita que o algoritmo aprenda a classificar as imagens no decorrer das camadas (WIKISTAT, 2021).

2.3.1 Frameworks e bibliotecas de Deep Learning

Frameworks e bibliotecas são códigos desenvolvidos para resolver problemas comuns de desenvolvimento. Atualmente existem vários *frameworks* disponíveis para facilitar o desenvolvimento de Redes Neurais e do seu treinamento. No Quadro 4 são apresentados os principais *frameworks* atuais (SPADINI, 2021).

Quadro 4 - Frameworks predominantes mundialmente

Framework	Desenvolvido por	Principais linguagens compatíveis
Tensorflow	Google Brain Team	C++, Python
Keras	François Chollet	Python
PyTorch	Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan	C++, Python
Microsoft CNTK	Microsoft	Python, C # ou C ++
Apache MXNET	Apache	C++, Python, Java, Julia, Matlab, JavaScript, Go, R, Scala, Perl e Wolfram
Sonnet	<i>Google DeepMind</i>	Python
Chainer	<i>IBM, Intel, Microsoft, Nvidia e AWS</i>	Python
Eclipse Deeplearning4j	Skymind	Java, Scala, Clojure, Kotlin

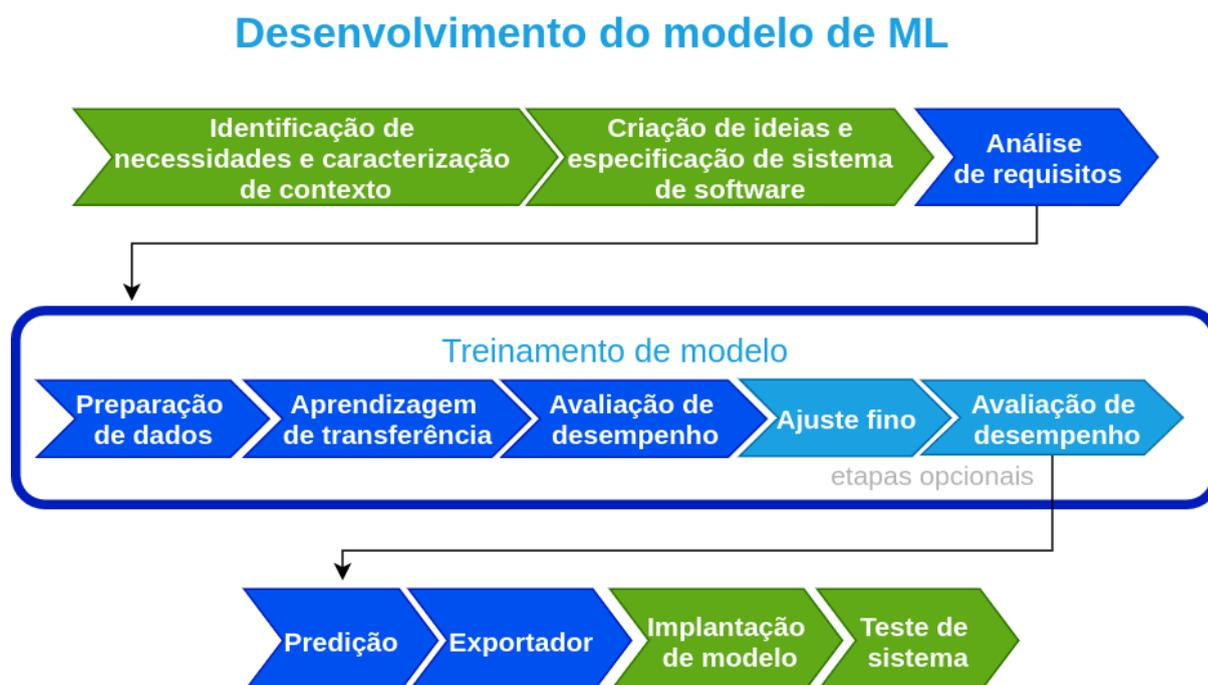
Fonte: SPADINI (2021)

Cada um desses *frameworks* tem suas vantagens e desvantagens, sendo possível utilizá-los dependendo da aplicação. Mesmo assim, ainda é necessário um esforço maior para a compreensão dos *frameworks*.

2.3.2 Processo de desenvolvimento de modelos de *Deep Learning*

A criação de um sistema de *Deep Learning* é um processo iterativo e exploratório que geralmente abrange a fase de desenvolvimento e implantação, como mostra a Figura 9.

Figura 9 - Etapas do desenvolvimento do modelo de Machine Learning



Fonte: Autora (2021)

A primeira etapa é definir qual é o objetivo do sistema de *Deep Learning*. Em seguida, coletam-se e preparam-se os dados. Os dados são validados e tratados, como remoção de duplicatas, correção dos dados, normalização, conversão de tipos de dados e etc. Os dados ainda podem ser pré-processados, assim transformando os dados brutos em algum formato para melhorar a aprendizagem.

Os dados são divididos tipicamente em conjuntos de dados para o treinamento do modelo, um conjunto de validação utilizado para selecionar o melhor candidato de todos os modelos e um outro conjunto de dados para testes para avaliar o desempenho do modelo escolhido utilizando dados não usados anteriormente.

Depois é escolhido ou construído o modelo de *ML*, ou normalmente escolhido a partir de modelos reconhecidos, utilizando *Frameworks* ou/e bibliotecas na sua

construção, tendo interface de programação de alto nível e, assim, facilitando a sua construção.

Após a construção do modelo de *ML*, inicia-se a fase de treinamento do modelo e comparação entre modelos de *Deep Learning* semelhantes utilizando os dados de treinamento nesta etapa. Após o treinamento de um modelo, avalia-se a sua qualidade para compreender como melhorar interativamente seu desempenho utilizando métricas, tais como: Acurácia, Taxa de Erro, Precisão, Recall e F1 Score. Em seguida, testa-se o modelo com os dados do conjunto de teste, esses dados não foram utilizados na etapa anterior. Nessa etapa o objetivo é ver o comportamento do modelo com dados não conhecidos, simulando a execução real.

Depois vem a fase de produção/implantação. O modelo é implantado em um ambiente de produção para ser utilizado no ambiente real, recebendo dados reais. É ideal que a implantação tenha testes para medir o desempenho do modelo e, assim, acionar alertas quando este diminui além de um certo limite aceitável (AMERSHI *et al.*, 2019).

2.3.3 Ambiente de desenvolvimento de modelo de ML

Existem diversos tipos de ambientes de desenvolvimento de modelos de ML sendo adotadas na educação Básica (GRESSE VON WANGENHEIM, 2021b).

Atualmente o *Jupyter Notebooks* é um dos ambientes mais populares para o desenvolvimento de modelos ML, por ser um ambiente de código aberto que permite aos seus usuários criar e compartilhar documentos do código executável, saída computacional, texto explicativo e recursos multimídia em um único arquivo. Nele é possível combinar várias linguagens de programação, sendo muito utilizado para desenvolver modelos de ML normalmente utilizando da linguagem Python (PROJECT JUPYTER, 2022). Nesse ambiente foi desenvolvido o VULCAN que é uma ferramenta visual de classificação de imagens para o ensino de *machine learning* (FRANZ, 2021).

Figura 10 - Interface do VULCAN no Google Colab

The image shows a JupyterLab interface with a terminal window at the top displaying the command `visual_layer()`. Below the terminal, a notification indicates that Google Drive is mounted at `/content/gdrive` and connected successfully. A navigation bar contains several tabs: 'Análise de requisitos', 'Preparação de dados' (which is selected and highlighted), 'Transfer learning', 'Avaliação transfer learning', 'Fine tuning (opcional)', 'Avaliação fine tuning (opcional)', 'Predição', and 'Exportação'. The main content area is divided into four sections: 1. 'Definição do Path' with a text input field for the Google Drive path and a 'Carregar Path' button. 2. 'Visualizar informações do conjunto de dados' with a 'Ver total por categoria' button. 3. 'Redimensionamento' with a checked checkbox labeled 'Padrão'. 4. 'Transformações e aumentações' with a checked checkbox labeled 'Desabilitar'.

Fonte: PROJECT JUPYTER (2022)

3 ESTADO DA ARTE

Com o objetivo de levantar o estado da arte, realiza-se um mapeamento sistemático da literatura para identificar quais são as unidades instrucionais que ensinam as competências de *Machine Learning* no contexto do Ensino Médio voltado ao estágio de *create* nos últimos 10 anos.

3.1 Definição do Protocolo de Revisão

Nessa pesquisa foram definidas as seguintes questões de análise:

QA1. Quais unidades instrucionais existem?

QA2. Quais competências ML são ensinadas nas unidades instrucionais?

QA3. Quais são as características instrucionais das unidades instrucionais?

QA4. Quais são as características de contexto das unidades instrucionais?

QA5. Como foram desenvolvidas as unidades instrucionais?

QA6. Como foi avaliada a qualidade das unidades instrucionais?

Para critérios de inclusão/exclusão de artigos foram considerados somente artigos e artefatos que ensinam Inteligência Artificial a adolescentes, publicados a partir de 2011. Não foram considerados os artigos/artefatos de ensino de Inteligência Artificial (e não especificamente ML) para outros níveis educacionais do ensino infantil, fundamental e superior. Somente cursos que aplicam conceitos de ML no nível *create* foram considerados.

Como critérios de qualidade foram considerados apenas artigos que apresentam informações substanciais sobre o ensino de competências de *ML*, indicando, no mínimo, conteúdo das lições, material instrucional etc.

As fontes de dados para responder a questão de pesquisa foram utilizadas as bases ACM Digital Library², IEEE Xplore Digital Library³, Sciencedirect⁴, Scopus⁵. Também foi utilizado o site do Google Scholar⁶ e Google⁷.

As *strings de busca* são compostas por palavras com conceitos relacionados às questões abordadas, incluindo sinônimos, conforme o Quadro 5. A partir dessas

² www.acm.org

³ www.ieeexplore.ieee.org

⁴ www.sciencedirect.com

⁵ www.scopus.com

⁶ www.scholar.google.com.br

⁷ www.google.com

palavras-chave, foi definida a *string* de pesquisa adaptando de acordo com a sintaxe específica de cada fonte de dados, como é mostrado no quadro 6.

Quadro 5 - Termos de busca e sinônimos

Termo	Sinônimos	Tradução (inglês)
Ensino	Instrução, Treinamento, Aprendizagem	<i>Education, Teach*, Learn*, Course, Tutorial, MOOC</i>
Deep Learning	Inteligência Artificial, Aprendizagem de máquina, aprendizado de máquina, aprendizagem profunda, aprendizado profundo	<i>Artificial intelligence, Machine Learning, Deep Learning, data science</i>
Educação básica	escolar, crianças, adolescentes,	<i>K-12, school, teen*</i>

Fonte: Autora (2020)

Quadro 6 - Strings de busca por base

Local de busca	String de busca
ACM	[Abstract: teach*] OR [Abstract: education] OR [Abstract: course] OR [Abstract: mooc] OR [Abstract: learn*] OR [Abstract: tutorial]] AND [[Abstract: "machine learning"] OR [Abstract: "data science"] OR [Abstract: "artificial intelligence"] OR [Abstract: "Deep Learning"]] AND [[Abstract: "k-12"] OR [Abstract: school*] OR [Abstract: teen*] AND [Publication Date: (01/01/2011 TO *)]
IEEE	((("Abstract":teach*) OR ("Abstract":education) OR ("Abstract":course) OR ("Abstract":MOOC) OR ("Abstract":learn*)) AND ((("Abstract":“machine learning”) OR (“Abstract”:“data science”) OR (“Abstract”:“artificial intelligence”) OR (“Abstract”:“Deep Learning”))) AND ((("Abstract”:“k-12”) OR (“Abstract”:school*) OR (“Abstract”:teen*))
Scimedirect	<i>“Machine Learning” “Deep Learning” “Artificial Intelligence” Education teach “K-12” “high-school”</i>
Scopus	<i>TITLE-ABS-KEY ((teach* OR education OR course OR mooc OR learn*) AND ("machine learning" OR "data science" OR "artificial intelligence" OR "Deep Learning") AND ("k-12" OR "high school" OR teen*)) AND (LIMIT-TO (PUBYEAR , 2021) OR LIMIT-TO (PUBYEAR , 2020) OR LIMIT-TO (PUBYEAR , 2019) OR LIMIT-TO (PUBYEAR , 2018) OR LIMIT-TO (PUBYEAR , 2017) OR LIMIT-TO (PUBYEAR , 2016) OR LIMIT-TO (PUBYEAR , 2015) OR LIMIT-TO (PUBYEAR , 2014) OR LIMIT-TO (PUBYEAR , 2013) OR LIMIT-TO (PUBYEAR , 2012) OR LIMIT-TO (PUBYEAR , 2011))</i>
Google	<i>“Machine Learning” “Deep Learning” “Artificial Intelligence” Education teach “K-12” “high-school”</i>
Google Scholar	<i>“Machine Learning” “Deep Learning” “Artificial Intelligence” Education teach “K-12” “high-school”</i>

Fonte: Autora (2020)

3.2 Execução da Busca

As buscas foram realizadas em agosto de 2021 pela autora do trabalho. A pesquisa foi dividida em duas etapas. Na primeira etapa foi aplicada a *string* de busca nas bases de dados já mencionadas e também em cada uma das bases. O Quadro 7 apresenta a quantidade dos resultados dessas buscas. A partir desses resultados foram aplicados os critérios de inclusão e exclusão. Foram analisados os 200 resultados mais relevantes. Na segunda etapa, todo o texto dos artigos pré-selecionados foi analisado, aplicando novamente os critérios de inclusão/exclusão e de qualidade. Como resultado foram identificados 5 artigos relevantes (Quadro 7).

Quadro 7 - Resultado da busca

Base de dados	Quantidade de artigos resultantes da busca	Quantidade analisada	Quantidade de artigos potencialmente relevantes	Quantidade de artigos relevantes
ACM	232	200	10	2
IEEE	717	200	5	0
Scopus	828	200	9	0
Sciencedirect	11	11	0	0
Google	104.000	200	5	1
Google Scholar	514	52	3	0

Fonte: Autora (2020)

3.3 Extração das Informações

De acordo com as perguntas de análise são identificadas informações extraídas dos materiais encontrados, conforme apresentado no quadro 8.

Quadro 8 - Informações extraídas das unidades instrucionais encontrados

Perguntas de análise	Item	Descrição
Quais unidades instrucionais existem?	Descrição	Descrição da unidade instrucional
	Referência	Referência da unidade instrucional
	Objetivo(s) de aprendizado em relação à ML após a unidade	Objetivo(s) aprendido na unidade instrucional

Quais competências ML são ensinadas nas unidades instrucionais?	instrucional, os alunos devem ser capazes de:	
	Processo de ML	Conhecimentos abordados na unidade instrucional
	Técnicas de ML	Técnicas de <i>ML</i> utilizados na unidade instrucional
	Ferramenta(s) ML	Ferramentas utilizadas no ensino <i>ML</i> na unidade instrucional
Quais são as características instrucionais das unidades instrucionais?	Objetivo geral de aprendizagem da unidade instrucional	Objetivo geral da aprendizagem da unidade instrucional
	Descrição geral	Descrição geral da unidade instrucional
	Modo instrucional	Modo instrucional que é aplicado na unidade instrucional
	Método instrucional	Método utilizado na unidade instrucional
	Material instrucional	Material utilizado na unidade instrucional
	Método(s) de avaliação / instrumento(s)	Método de avaliação utilizado na unidade instrucional
Quais são as características de contexto das unidades instrucionais?	Nível escolar	Nível escolar da unidade instrucional
	Duração da unidade instrucionais	Duração da aplicação da unidade instrucional
	Pré-requisitos	Conhecimento necessário para participar da unidade instrucional
Como foram desenvolvidas as unidades instrucionais?	Método de desenvolvimento das unidades instrucionais	Metodologia de desenvolvimento da unidade instrucionais
Como foi avaliada a qualidade das unidades instrucionais?	Projeto de pesquisa	Projeto de pesquisa utilizado na unidade instrucionais
	Fatores avaliados	Fatores avaliados nas unidades instrucionais
	Métodos de coleta de dados	Métodos utilizadas para coletar dados na unidade instrucionais
	Tamanho da amostra	Quantidade de participantes na unidade instrucionais
	Estudos replicados	Quantidade de participantes que participaram da replicação da unidade instrucionais
	Método(s) de análise de dados	Maneira que os dados da unidade instrucional foram analisados
	Constatações	Constatação obtida após a aplicação na unidade instrucional

Fonte: Autora (2020)

3.4 Análise dos Resultados

A partir dos artigos selecionados, realiza-se a análise das perguntas anteriormente citadas.

3.4.1 Quais unidades instrucionais existem?

No quadro 9 são identificadas as unidades instrucionais.

Quadro 9 - Unidades instrucionais encontradas

Citação	Descrição
(SPERLING e LICKERMAN, 2012)	Curso introdutório e presencial que ensina as diversas áreas de conhecimento da <i>Machine Learning</i> .
(IBM, 2020)	Curso online que ensina os conceitos básicos de <i>Machine Learning</i> focado na classificação de imagens, desde preparação dos dados até avaliação do modelo.
(MIKE; HAZAN; HAZZAN, 2020)	Curso presencial que ensina sobre ciência de dados e seu fluxo de trabalho com a compreensão profunda da <i>Machine Learning</i> .

Fonte: Autora (2020)

Como resultado, foram identificadas três unidades instrucionais que ensinam IA no Ensino Médio no nível *create*. Isso demonstra que ainda existem poucas unidades instrucionais que abordam o ensino *ML* no médio no nível *create*.

3.4.2 Quais competências ML são ensinadas nas unidades instrucionais?

As unidades instrucionais ensinam as principais competências das áreas de conhecimento de *ML*, para a compreensão do processo desenvolvimento de *ML*. Entre as áreas mais frequentemente abordadas pelas unidades instrucionais estão as áreas relacionadas às principais fases do processo de *ML*: coleta de dados, preparação dos dados, treinamento e avaliação do modelo. As etapas mais ensinadas por essas unidades instrucionais são a coleta de dados e o treinamento do modelo. As unidades instrucionais ensinam *ML* e seu processo de desenvolvimento por meio de um cenário específico onde o aluno coleta os dados e treina um modelo já definido. Observa-se que a linguagem *DrRacket* é utilizada para o desenvolvimento das atividades e projeto final, mas não foi informado se foi utilizado algum *Framework* (SPERLING; LICKERMAN, 2012). Já a unidade instrucional proposta pelo IBM (2020), deixa livre para o aluno escolher, mas nos módulos anteriores foram apresentadas ferramentas para o desenvolvimento do

modelo de *ML*. Na unidade instrucional proposta por Mike, Hazan e Hazzan (2020), o aluno desenvolve usando o *Google Colab* como ambiente de programação e linguagem *Python*.

O processo do *create* normalmente começa quando a parte teórica dos cursos é concluída, deixando a área do tema aberta. No trabalho de Mike, Hazan e Hazzan (2020) o questionamento de qual tema será o projeto ocorre logo no início da etapa do projeto final, por necessitar que os alunos tenham os dados para o treinamento. Já no trabalho de Sperling e Lickerman (2012), os alunos iniciam escrevendo a proposta do trabalho final, dessa forma já estimulando a pensar nos problemas que aparecerão na solução para a sua proposta. No curso online da IBM (2020), no módulo do projeto final é apresentado *Design Thinking*, e assim ensinando o estudante como idealizar e desenvolver o projeto final.

Os dados extraídos referente a PA2 são apresentados no quadro 10.

Quadro 10 - Visão geral sobre as competências ML ensinadas nas unidades instrucionais

Identificador	Objetivo(s) de aprendizado Em relação à <i>ML</i> após a unidade instrucional, os alunos devem ser capazes de:	técnicas de <i>ML</i>	Processo de <i>ML</i>	Ferramenta(s) <i>ML</i>
(SPERLING e LICKERMAN, 2012)	Criar seu próprio sistema utilizando seu modelo <i>ML</i> .	Classificação de imagem Árvore de decisão Vizinho mais próximo K vizinhos mais próximos (KNN)	Preparação de dados Treinamento e avaliação de modelo	Não informado
(IBM, 2020)	Entenda sobre os conceitos básicos de <i>ML</i> e crie seus próprios sistemas que implementam esses conceitos por meio da classificação de imagens.	Classificação de imagem	Preparação de dados Treinamento e avaliação de modelo	Teachable machine
(MIKE; HAZAN; HAZZAN, 2020)	Introdução a ciência de dados; Introdução a Machine Learning; <i>criação das suas</i>	Classificação de imagem K vizinhos mais próximos (KNN) Dados e tabelas	Preparação de dados Treinamento e avaliação de modelo	Python em Notebooks Jupyter no Google Colab

	<i>ML</i> ;	O algoritmo perceptron O algoritmo de máquina de vetores de suporte (SVM)		
--	-------------	--	--	--

Fonte: Autora (2020)

3.4.3 Quais são as características instrucionais das unidades instrucionais?

O ensino das competências *ML* está nas abordagens de ensino a distância (IBM, 2020) e no ensino presencial (SPERLING e LICKERMAN, 2012; MIKE; HAZAN; HAZZAN, 2020). Na abordagem presencial, a unidade instrucional ensina por meio de oficinas presenciais que ensinam o básico de *ML* até o desenvolvimento do projeto final. Já a unidade instrucional da IBM (2020) ensina desde o básico através da divisão dos conhecimentos por módulos. Então, ao decorrer do módulo os estudantes passam da abstração para a parte prática. No quadro 11 são mostradas as características das unidades instrucionais.

Quadro 11 - Visão geral sobre as características das unidades instrucionais

Identificador	Objetivo geral de aprendizagem da unidade instrucional	Descrição geral	Modo instrucional	Método instrucional	Material instrucional	Método(s) de avaliação / instrumento(s)
(SPERLING e LICKERMAN, 2012)	Ensinar os conceitos básicos de <i>ML</i> .	As aulas iniciassem com a parte teórica e ao decorrer das aulas vão para parte prática até chegar no projeto final	Presencial	Aula teórica, aula prática a partir de um projeto em específico e projeto final livre	Não informado	Execução do projeto final
(IBM, 2020)	Ensinar as etapas de <i>ML</i> e seus conceitos básicos	As aulas são divididas em módulos, iniciando com a parte teórica, indo para a prática e terminado com projeto	online	Aula teórica, aula prática a partir de um objetivo em específico	atividades interativas e vídeos	Exame com 20 questões de múltipla escolha

		final no último módulo				
(MIKE; HAZAN; HAZZAN, 2020)	Ensinar sobre ciência de dados e seu fluxo de trabalho com a compreensão profunda do ML.	As aulas iniciam com a parte teórica e ao decorrer das aulas vai para parte prática até chegar no projeto final	Presencial	Aula teórica, aula prática a partir de um projeto em específico e projeto final livre	Não informado	Execução do projeto final e relatório final

Fonte: Autora (2020)

3.4.4 Quais são as características de contexto das unidades instrucionais?

Todas as unidades instrucionais têm foco específico no ensino de *IA/ML* exceto Sperling e Lickerman (2012) que tem como objetivo de ensinar engenharia de software e assim tendo um currículo mais denso, mas também abordando *ML* com umas das suas disciplinas. Já Mike, Hazan e Hazzan (2020) abordam *IA/ML* de uma forma mais detalhada e, assim, apresentam um currículo denso também, conforme apresentado no quadro 12.

Quadro 12 - Visão geral das características de contexto da unidade instrucional

Identificador	Nível escolar	Duração da unidade instrucional total	Pré-requisitos
(SPERLING e LICKERMAN, 2012)	Ensino Médio	450 horas	Não informado
(IBM, 2020)	Anos finais do Ensino Fundamental e Ensino Médio	15 horas	Não informado
(MIKE, 2020)	Ensino Médio	90 horas	Não informado

Fonte: Autora (2020)

3.4.5 Como foram desenvolvidas e avaliadas as unidades instrucionais?

Para atingir os resultados efetivos de aprendizado, as unidades instrucionais precisam ser desenvolvidas sistematicamente, seguindo os modelos de design instrucional. No entanto, foi observada uma falta geral de informações nos artigos em relação à forma como as unidades instrucionais foram desenvolvidas. Poucas publicações fornecem informações, e geralmente de forma escassa, sobre o

desenvolvimento das unidades instrucionais, por meio de cooperações entre escolas e/ou universidades contando com a participação dos professores e instrutores.

Quadro 13 - Método de avaliação das unidades instrucionais

Identificador	Método de avaliação da unidade instrucionais
(SPERLING e LICKERMAN, 2012)	Avaliação informal com base nas opiniões dos estudantes
(IBM, 2020)	Não informado
(MIKE; HAZAN; HAZZAN, 2020)	Não, é um dos objetivos futuros do trabalho.

Fonte: Autora (2020)

3.4.6 Como foi avaliada a qualidade das unidades instrucionais?

Uma forma de verificar a qualidade e/ou melhorar as unidades instrucionais é por meio da sua avaliação. A avaliação é normalmente realizada por meio de estudos empíricos em sala de aula. Dos três encontrados, dois têm avaliação das unidades instrucionais por meio de estudo de caso (SPERLING; LICKERMAN, 2012; MIKE; HAZAN; HAZZAN, 2020). Já no outro não foi informado se houveram avaliações.

No trabalho de Sperling e Lickerman (2012), ao finalizar o workshop, as opiniões dos estudantes sobre o workshop foram coletadas, perguntando se conseguiram concluir e qual foi a dificuldade, mas sem um questionário pensado para essa coleta de informações, concluindo que o workshop foi realizado com sucesso. No trabalho piloto de Mike, Hazan e Hazzan (2020), a forma dessa avaliação foi a verificação se os estudantes conseguiram concluir ou não o trabalho final. Enquanto isso, o trabalho da IBM (2020) não teve avaliação.

Uma visão geral das informações extraídas com relação à avaliação das unidades instrucionais é apresentada no quadro 14.

Quadro 14 - Visão geral sobre as avaliações das unidades instrucionais

Identificador	Projeto de pesquisa	Fatores avaliados	Métodos de coleta de dados	Tamanho da amostra	Estudos replicados	Método(s) de análise	Constatações
---------------	---------------------	-------------------	----------------------------	--------------------	--------------------	----------------------	--------------

						de dados	
(SPERLING e LICKERMAN, 2012)	Estudo de caso	Aprendizagem do conteúdo	Opiniões dos estudantes	Não informado	Não informado	Análise das opiniões	Realizado com sucesso
(IMB, 2020)	Não informado	Não informado	Não informado	Não informado	Não informado	Não informado	Não informado
(Mike, 2020)	Estudo de caso	Aprendizagem do conteúdo	Não tem	Não informado	Não informado	Não informado	Piloto realizado com sucesso

Fonte: Autora (2020)

3.5. Discussão

Ainda não há muitas unidades instrucionais voltadas ao ensino da ML, e esse número diminui muito mais quando a pesquisa aponta para o ensino de ML voltado ao nível create no Ensino Médio, com apenas três unidades instrucionais encontradas. Essas unidades instrucionais focam em ensinar o processo de desenvolvimento ML e com a criação dos seus próprios projetos, da maneira possível encontrada por eles. As unidades instrucionais focaram em ensinar as principais fases do desenvolvimento de ML, incluindo preparação de dados, treinamento e avaliação de modelos.

Observou-se, também, que há iniciativas que podem ser ensinadas no Ensino Fundamental (IBM, 2020), mas há mais unidades institucionais no Ensino Médio (SPERLING; LICKERMAN, 2012; MIKE; HAZAN; HAZZAN, 2020). Nenhum dos autores relatou dificuldade ou baixa aprendizagem no ensino de ML. Nos trabalhos analisados notou-se que os estudantes conseguiram desenvolver as atividades propostas nos cursos presenciais e no *online*.

As unidades instrucionais se preocupam em ensinar os conceitos básicos de ML, como classificação de imagem, mas também é ensinado a outros (SPERLING; LICKERMAN, 2012; MIKE; HAZAN; HAZZAN, 2020). Assim utilizando outros tipos de algoritmos de aprendizagem, mostrando as principais áreas possíveis na utilização da ML.

As unidades instrucionais dividem as aulas em três partes: na primeira parte da aula é focada no conteúdo teórico, já a segunda parte da aula é reservada para a parte prática e, por fim, a terceira parte para o desenvolvimento do projeto final.

Com o objetivo de disseminar as unidades instrucionais observamos, em geral, a falta de disponibilidade de informações detalhadas e/ou material instrucional. Nenhuma das unidades instrucionais tem material em português, mas notou-se a diversidade das nacionalidades das unidades. Com essa limitação, o material criado pelas unidades instrucionais podem ser descartados ou demandar maior esforço para adequar o material para o português, assim prejudicando uma aplicação em maior escala.

Outra característica identificada foi a utilização das ferramentas *Python*, *Jupyter Notebooks*, *Google Colab* e *Teachable Machine* para as atividades no ensino de ML (IBM, 2020; MIKE; HAZAN; HAZZAN, 2020).

Em resumo, foi possível verificar a existência de poucas unidades instrucionais que ensinam o processo de desenvolvimento de *ML* no Ensino Médio no nível *create*, mas todos os autores relataram que não tiveram dificuldade em ensinar.

Existem, ainda, alguns motivos que podem ameaçar a validade do mapeamento. Entre eles, está não ter encontrado todas as unidades instrucionais ou ter deixado de fora alguma relevante. Para minimizar este risco, utilizou-se as bases *ACM Digital Library*, *IEEE Xplore Digital Library*, *Scimedirect*, *Scopus* e também o site da Google na busca. Também aplicou-se uma estratégia de busca que usa palavras-chave na busca, junto com os seus sinônimos e traduções para a língua inglesa, desta forma, reduzindo o risco de não ter encontrado um artigo de relevância para a pesquisa.

4 CURSO “DESENVOLVA SEU APLICATIVO INTELIGENTE PARA CLASSIFICAÇÃO DE IMAGENS USANDO VULCAN”

Neste capítulo, apresenta-se uma análise do contexto focado pela unidade instrucional. Também é apontado o design da unidade instrucional por meio dos objetivos de aprendizagem, plano de ensino e materiais didáticos para a unidade instrucional.

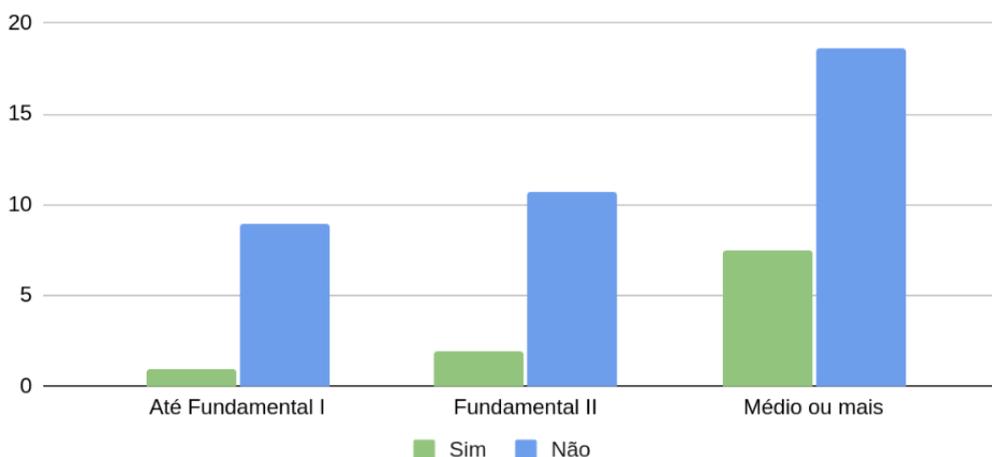
4.1. Análise do Contexto

Este curso evidencia estudantes do Ensino Médio como público-alvo, pois leva em consideração que, atualmente, muitos adolescentes nessa idade (15 aos 17 anos) já têm contato com tecnologia e até mesmo possuem seus próprios dispositivos eletrônicos, como smartphone, tablet e computador. Ainda assim, segundo a pesquisa TIC Kids online (CETIC.BR, 2019), 7,4 milhões de estudantes possuem computadores com acesso à internet em seus domicílios, enquanto 18,6 milhões não têm, conforme apresentado na figura 11.

Figura 11 - Quantidade de estudantes que apresentam computadores no domicílio com acesso à internet

ESTUDANTE QUE APRESENTAM COMPUTADORES NO DOMICÍLIO COM ACESSO A INTERNET (2019)

Estimativa em milhões de estudantes de 9 a 17 anos



Fonte: CETIC.BR (2019)

Os estudantes do Ensino Médio apresentam conhecimentos mais consolidados, já alfabetizados e com conhecimento básico em inglês, além das

outras áreas de conhecimentos. No Ensino Médio, muitos estudantes já têm acesso diário à internet através de dispositivos eletrônicos, e na escola, de forma interdisciplinar, os alunos têm contato com os computadores no laboratório de informática.

Segundo o QEDu, utilizando os dados do censo escolar de 2020, cerca de 82,2% das escolas públicas brasileiras que oferecem Ensino Médio, tem laboratório de informática, sendo que 73,2% dessas escolas possuem laboratório com acesso à internet para os alunos (QEDU, 2021).

A BNCC não prevê ensino de computação em uma matéria em específico para o Ensino Médio, e o laboratório de informática é utilizado pelos professores para relacionar conteúdo visto em sala de forma interdisciplinar.

Em geral, o professor elabora uma atividade que utiliza editor de texto, editor de apresentação e navegador web. Desta forma, o professor conecta a atividade para ensinar/reforçar o conteúdo proposto com o uso da tecnologia, como: escrever o resumo da aula ou apresentar para sala sobre um tema específico utilizando um editor de apresentação (CETIC.BR, 2019).

O Ensino Médio pode contar, além do ensino comum, com ensino integral profissional e técnico. Os períodos são divididos em matutino e vespertino, compostos por cinco aulas de 45 minutos de duração cada (BRASIL, 2019).

4.1.1 Objetivos de Aprendizagem

Os objetivos de aprendizagem do curso foram criados a partir do *Framework CSTA K-12 CS* (CSTA, 2019) e *The Artificial Intelligence for K-12* (AI4K12, 2019), também tendo objetivos de *Design Thinking*, algoritmo e programação. O curso, em sua forma completa, visa ensinar a identificação de um problema, idealização de uma solução por meio de um app inteligente, o desenvolvimento de um modelo de ML e a sua implantação em um app. Porém, como atualmente o desenvolvimento de uma extensão do *App Inventor* para permitir a implantação dentro de um app está ainda em andamento, esta versão do curso se enfoca na parte inicial voltada aos passos de identificação de problema, idealização de solução e desenvolvimento do modelo de ML, a ser complementado no futuro pelas demais partes do curso.

Quadro 15 - Objetivos de aprendizagem do curso
(todos os objetivos marcados em cinza se referem a uma parte futura do curso tendo como pré-requisito a disponibilidade de uma extensão para App Inventor)

ID	Objetivo de aprendizagem	Área de conhecimento	Fonte
OA1	Identificar e resolver problemas criando sistemas de software inteligentes.	Algoritmo e programação / Engenharia de Usabilidade	(CSTA, 2019: 1B-AP-09, 1B-AP-10, 2-AP-11) (CSTA, 2017: 1BCS-03, 3A-AP13), (AIGA, 2013)
OA2	Compreender o processo do <i>Design Thinking</i> .	<i>Design Thinking</i>	
OA3	Criar protótipos que usam algoritmos para resolver problemas computacionais.	Algoritmo e Programação	(CSTA, 2019: 3A-AP13)
OA4	Preparar um conjunto de dados para o treinamento do modelo de <i>ML</i> , passados pelas suas etapas.	<i>ML</i>	(AI4K12, 2019)
OA5	Treinar um modelo de <i>ML</i> de classificação de imagem.	<i>ML</i>	(AI4K12, 2019)
OA6	Avaliar o desempenho do modelo de <i>ML</i> treinado.	<i>ML</i>	(AI4K12, 2019)
OA7	Exportar o modelo de <i>ML</i> treinado.		
OA8	Explicar o conceito de um processo de <i>ML</i> e fornecer um exemplo, ilustrando suas fases, incluindo as entregas que são produzidas.	Engenharia de Software/ Engenharia de Usabilidade	ACM/IEEE (2013), UXQB (2018)
OA9	Analisar o contexto de sistemas de software inteligente em termos de usuários, tarefas, dispositivos e ambientes de uso	Engenharia de Usabilidade	(AIGA, 2013), (ISO/IEC 9241-220:2019, 2019) (CSTA, 2017: 1BIC-19), ACM/IEEE (2013)
OA10	Especificar requisitos de sistemas de software inteligente em termos de funcionalidade e usabilidade.	Engenharia de Software / Engenharia de Usabilidade	(CSTA, 2017: 2-AP-19), ACM/IEEE (2013)
OA11	Criar protótipos de sistemas de software inteligente em diferentes níveis (esboços, baixa fidelidade, alta fidelidade, funcional).	Engenharia de Usabilidade	(CSTA, 2017: 2-AP-19, 1B-AP-13, 2-AP-13), ACM/IEEE (2013)
OA12	Projetar o design visual (cores, tipografia, ícones, imagens, etc.) do sistema de software inteligente.	Engenharia de Usabilidade	ACM/IEEE (2013), (CSTA, 2017: 2-IC-21), (GARRET,

			2011)
OA13	Testar e refinar um sistema de software inteligente para funcionalidade e usabilidade.	Engenharia de Software/ Engenharia de Usabilidade	(CSTA, 2017: 2-AP-17, 1B-AP-15, 3A-AP-21)
OA14	Compartilhar o sistema de software inteligente desenvolvido	Algoritmo e programação	(CSTA, 2017: 1BAP-12, 1B-AP-17, 2-AP-16), (LEE <i>et al.</i> , 2007)

Fonte: Autora (2022)

4.2. Projeto do Curso

O projeto deste curso ocorreu a partir da análise do contexto e da verificação das necessidades referentes ao ensino de *Machine Learning* no Ensino Médio com base no *Framework* CSTA K-12 CS e diretrizes de AI4K12 (Touretzky *et al.*, 2019) e nas competências de Long e Magerko (2020) e AI4K12 (AI4K12, 2019), que é guiado para o ensino de computação com seus objetivos de aprendizagem.

4.2.1 Contexto de Aplicação

O curso foi projetado para ser ofertado de forma online ou presencial (com ou sem instrutor) com 23 horas-aulas, e busca auxiliar o estudante a definir e desenvolver seu próprio aplicativo com seu próprio modelo de ML de classificação de imagem. Voltado a um nível avançado, espera-se que o estudante já tenha as competências de algoritmos e programação no nível básico, que consiga desenvolver modelos de classificação de imagens utilizando *Jupyter Notebooks* no *Google Colab*, usando uma linguagem textual *Python* e saiba desenvolver app com *App Inventor*. Este curso integra a iniciativa Computação na escola, fazendo parte da trilha de Machine Learning, e tem como pré-requisito o curso ML4Teens (CARDOZO, 2022).

4.2.2 Plano de ensino

O plano de ensino foi desenvolvido para definir os objetivos do curso, levando em consideração o nível escolar dos alunos do Ensino Médio. Adota-se, principalmente, uma metodologia ativa para levar os estudantes a aplicar os conceitos de computação e ML. Enfocando o nível de *create*, o curso guia o aluno passo-a-passo a desenvolver o seu próprio aplicativo inteligente voltado a

classificação de imagens empregando *design thinking*. Adotando a ação computacional, o curso de forma interdisciplinar também orienta o aluno a desenvolver um aplicativo para resolver uma necessidade identificada na sua comunidade. Como exemplo, é apresentado um app que classifica imagens de larvas de mosquito visando a prevenção de focos do mosquito *aedes aegypti*, com o objetivo de diminuir os casos de dengue.

Para as atividades práticas, foi utilizada a ferramenta visual VULCAN dentro do Jupyter Notebook (FRANZ, 2021).

Quadro 16 - Plano de ensino do curso

Aula	Conteúdo	Objetivos de Aprendizagem	Método Instrucional	Material Instrucional	Avaliação
Introdução					
1 - (30 min)	Motivação	OA2		Slides	--
Análise					
2 - (1h)	Descoberta	OA1, OA2	Atividade prática, discussão	Slides, Google drive	--
3 - (1h)	Idealização da proposta	OA1, OA2, OA9	Atividade prática, discussão	Slides, Google drive	--
Desenvolvimento do modelo ML					
4 - (10h)	Preparação dos dados (coleta, rotulação e pré-processamento)	OA4	Atividade prática	Slides, VULCAN, Google drive	Rubrica de ML
5 - (1h)	Treinamento, avaliação do desempenho o modelo	OA5, OA6	Atividade prática	Slides, VULCAN, Google drive	Rubrica de ML
6 - (1h)	Treinamento, avaliação na etapa de ajuste fino e exportação	OA5, OA6, OA7	Atividade prática	Slides, VULCAN, Google drive	Rubrica de ML
Construção e teste do app inteligente com App Inventor					
7 - (1h)	Criar e testar sketches	OA11	Atividade prática, Vídeo, Lápis e papel	Slides, Google drive	--
8 - (1h)	Programar e testar o wireframe	OA11, OA12	Atividade prática, Vídeo, Sketch2aia, Wireframe app Xô Dengue	Slides, Google drive	--
9 - (3h)	Criar o design visual	OA12	Atividade prática	Slides, Google drive	Codemaster - design

					visual
10	Implantar o modelo de ML	OA11, OA12, OA13	Atividade prática, App Inventor	Slides, Google drive	--
11 - (2h e 30min)	Programar e testar o app funcional	OA13	Atividade prática, Vídeos	Slides, Google drive	CodeMaster - Algoritmos & programação
12 - (1h)	Compartilhamento do aplicativo (via .apk)	OA14	Atividade prática	Slides, Google drive	--

Fonte: Autora (2022)

4.2.3 Desenvolvimento do Material Didático

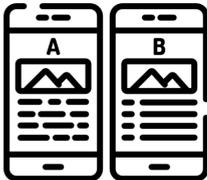
Com base no plano de ensino foi desenvolvido o material didático, composto por *slides*, vídeos para apresentar os conceitos e guiar o estudante a desenvolver o seu app inteligente passo-a-passo.

Quadro 17 - Exemplo de slides do curso

Slides	
Aula 1	Aula 2
<p>O que é Machine Learning?</p> <p>Machine Learning é uma área dentro da Inteligência Artificial, que estuda formas para que os sistemas possam executar tarefas predeterminadas sem ter que utilizar instruções explícitas, tomando as decisões a partir de padrões e inferências aprendidas pelo conjunto de dados</p> 	<p>DESCOBERTA</p> <p>Descubra mais sobre o contexto para qual você estará produzindo uma solução</p> <p>Quem são os usuários? Quais objetivos/tarefas eles fazem? Eles usam celulares Android? Para quê? Em quais ambientes? Quais problemas encontram? Quais são as suas necessidades?</p> 
Aula 3	Aula 4
<p>HISTÓRIA DO USUÁRIO</p> <p>História do usuário é uma explicação informal e geral sobre uma funcionalidade do App ou sobre o próprio App que será desenvolvido, usando a perspectiva do usuário.</p> <p>COMO [quem?], EU QUERO [o que?], PARA que eu possa [por quê].</p>	<p>Análise dos requisitos</p> <p>A etapa de análise dos requisitos tem como objetivo identificar e especificar o objetivo do modelo e que problemas ele resolve, assim como a caracterização das entradas e saídas esperadas.</p> 
Aula 5	Aula 6

<h3 style="background-color: #8e44ad; color: white; padding: 5px;">Epochs</h3> <p>Epochs é a quantidade de vezes que o algoritmo se passa pelo conjunto de dados treinando, analisando os padrões e atualizando os parâmetros internos. Quanto mais epochs melhor será seu modelos, mas custará mais tempo de treinamento</p> <p>Basta definir quantos epochs e clicar em "Treinar Modelo"</p>  <p>Sugestão: Inicie o treinamento do modelo com 15 epochs</p>	<h3 style="background-color: #8e44ad; color: white; padding: 5px;">Ajuste Fino</h3> <p>Na etapa de ajuste fino (Fine Tuning) é o momento onde ocorre adequações no modelo para aumentar o desempenho do modelo. Buscando possíveis vícios nos dados de treinamento (overfitting)</p> <p>Na etapa seguinte é Avaliação do ajuste fino é a etapa onde avalia a performance do modelo após o ajuste fino</p>
--	---

Aula 7	Aula 8
---------------	---------------

<h3 style="background-color: #e74c3c; color: white; padding: 5px;">Interagindo com um app inteligente</h3> <p>Apps inteligentes são baseados em probabilidade e incerteza</p> <div style="display: flex; align-items: center; gap: 20px;"> <div style="text-align: center;">  <p>Imagem de larva de aedes-aegypti</p> </div> <div style="text-align: center;">  <p>Modelo de ML</p> </div> <div style="text-align: left;"> <p>90% probabilidade que é larva de aedes aegypti</p> <p>7% probabilidade que não é aedes aegypti</p> <p>3% probabilidade que não é larva de mosquito</p> </div> </div>	<h3 style="background-color: #e74c3c; color: white; padding: 5px;">Wireframe</h3> <p>Wireframe é o esqueleto do protótipo da tela do app, sendo uma versão bem simple das telas</p> 
---	---

Aula 9	Aula 10
---------------	----------------

<h3 style="background-color: #e74c3c; color: white; padding: 5px;">Design visual</h3> <p>O design visual se preocupa com a estética visual e usabilidade do aplicativo, escolhendo as cores, fontes tipográficas e imagens da interface</p> <div style="display: flex; flex-direction: column; align-items: center; gap: 10px;"> <div style="display: flex; align-items: center; gap: 10px;">  Cor </div> <div style="display: flex; align-items: center; gap: 10px;">  Tipografia </div> <div style="display: flex; align-items: center; gap: 10px;">  Imagem </div> </div>	<h3 style="background-color: #e74c3c; color: white; padding: 5px;">Teste funcionais</h3> <p>Com base na atividade Sketches, onde foi relembrado as histórias de usuários e definida as funcionalidades, avale se seu app funciona ou não</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 10px;"> <tr> <td>História de usuário</td> <td colspan="2"><xx história de usuário xx></td> </tr> <tr> <td>Caso de teste funcional</td> <td>Funcionou?</td> <td style="text-align: center;"> <input checked="" type="checkbox"/> ✓ <input type="checkbox"/> ✗ </td> </tr> <tr> <td rowspan="2">Funcionalidade(s)</td> <td><xx qual é a funcionalidade xx></td> <td></td> </tr> <tr> <td><xx qual é a funcionalidade xx></td> <td></td> </tr> </table> <p>RESULTADO o app <funciona/não funciona> corretamente</p>	História de usuário	<xx história de usuário xx>		Caso de teste funcional	Funcionou?	<input checked="" type="checkbox"/> ✓ <input type="checkbox"/> ✗	Funcionalidade(s)	<xx qual é a funcionalidade xx>		<xx qual é a funcionalidade xx>	
História de usuário	<xx história de usuário xx>											
Caso de teste funcional	Funcionou?	<input checked="" type="checkbox"/> ✓ <input type="checkbox"/> ✗										
Funcionalidade(s)	<xx qual é a funcionalidade xx>											
	<xx qual é a funcionalidade xx>											

Aula 11

Compartilhar os seu app

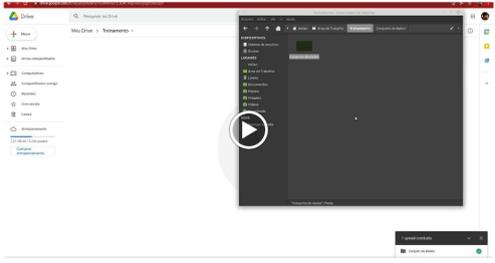
Agora vamos aprender como gerar o app no formato do **arquivo em APK**

Mas o que é **APK?** é um arquivo que instala um app em celulares Android

Assim você pode compartilhar o seu app neste formato para os seus amigos e familiares



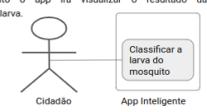
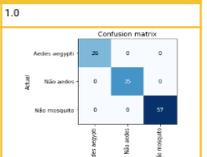
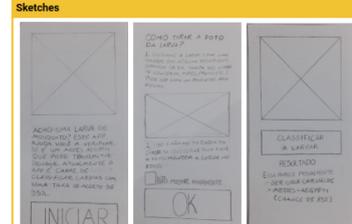
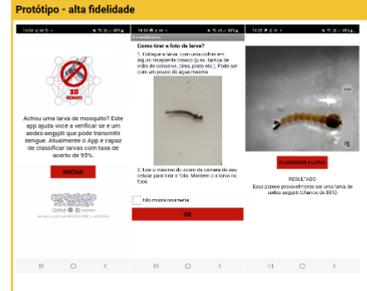
Quadro 18 - Exemplo de vídeos do curso

Vídeos	
<p>Instrução de como utilizar o Google Drive</p>	<p>Coleta de dados</p> <p>Na sua conta do Google Drive crie uma pasta para o conjunto de dados. Adicione no conjunto de dados no Google Drive, para isso basta arrastar a pasta com os dados para o Google Drive.</p> 
<p>Teste de usabilidade</p>	<p>Teste do sketches</p> 
<p>Tutorial Sketch2aia</p>	<p>Como vamos fazer os wireframes?</p> <p>Utilizaremos o Sketch2aia para criar os <i>wireframes</i></p> <p>Assista o vídeo para conhecer o Sketch2aia</p> 
<p>Tutorial do App Inventor</p>	<p>Tutorial do App Inventor</p>  <p>Canal Krishna Raghavendran Canal The Coding Bus</p>

Fonte: Autora (2022)

Foram projetados também modelos (*templates*) de artefatos a serem criados pelos alunos em forma de Caderno (*Google Docs*) e Caderno exemplo apresentando o desenvolvimento do app Xô Dengue (*Google Docs*). O Caderno projetado apresenta os templates de todos os artefatos a serem criados ao longo do curso.

Quadro 19 - Caderno exemplo

DESCOBERTA Identificação da necessidade		IDEAÇÃO Requisitos funcionais e de usabilidade		DESENVOLVIMENTO DO MODELO DE ML																																			
<p>Quais são as categorias de usuário?</p> <p>No uso do aplicativo móvel será envolvido somente uma categoria de usuário: o cidadão, interessado em descobrir se a larva encontrada é de um mosquito <i>aedes aegypti</i> ou não. O cidadão irá capturar uma foto da larva com seu celular e após o processamento o app irá visualizar o resultado da classificação da larva.</p> 		<p>Requisitos funcionais</p> <p>História de usuário COMO cidadão EU QUERO descobrir se uma larva encontrada é do mosquito <i>aedes aegypti</i> ou não PARA reduzir o risco da dengue</p> <p>Funcionalidade(s) Tirar foto de uma larva de mosquito Visualizar resultados da classificação</p>		<p>Avaliação do desempenho</p> <p>Acurácia 1.0</p> <p>Matriz de confusão</p> 																																			
<p>Persona</p>  <p>Biografia Joana adora cultivar plantas na sua casa. Ultimamente começou a surgir larvas nos vasos e não sabe se essas larvas são do mosquito <i>aedes-aegypti</i></p> <p>Qual seu objetivo? Saber se suas plantas são um ponto de reprodução do mosquito <i>aedes-aegypti</i> no seu bairro</p> <p>O que gosta? Em seu tempo livre, gosta de cultivar suas plantas. Assistir filmes e séries</p> <p>Conhecimento em TI Smartphone Android: intermediário, usando Whatsapp para se comunicar e Instagram para ver o feed e postar fotos</p> <p>O que não gosta? Ficar sem internet. Plantas murchas. Lugares com muitas pessoas</p>		<p>Requisitos de usabilidade</p> <p>História de usuário Como cidadão, eu quero diferenciar se as larvas que estão nos meus vasos de plantas são do mosquito <i>aedes-aegypti</i>, para que consiga eliminá las e prevenir a epidêmico</p> <p>Eficácia 85% dos usuários conseguem completar a tarefa</p> <p>Eficiência Realizar a tarefa em até 1 minuto</p> <p>Satisfação Todos os usuários acham fácil realizar a tarefa</p>	<p>Métricas</p> <table border="1"> <thead> <tr> <th></th> <th>precision</th> <th>recall</th> <th>f1-score</th> <th>support</th> </tr> </thead> <tbody> <tr> <td><i>aedes aegypti</i></td> <td>1.00</td> <td>1.00</td> <td>1.00</td> <td>26</td> </tr> <tr> <td>Não aedes</td> <td>1.00</td> <td>1.00</td> <td>1.00</td> <td>37</td> </tr> <tr> <td>Não mosquito</td> <td>1.00</td> <td>1.00</td> <td>1.00</td> <td>37</td> </tr> <tr> <td>accuracy</td> <td>1.00</td> <td>1.00</td> <td>1.00</td> <td>118</td> </tr> <tr> <td>macro avg</td> <td>1.00</td> <td>1.00</td> <td>1.00</td> <td>118</td> </tr> <tr> <td>weighted avg</td> <td>1.00</td> <td>1.00</td> <td>1.00</td> <td>118</td> </tr> </tbody> </table>		precision	recall	f1-score	support	<i>aedes aegypti</i>	1.00	1.00	1.00	26	Não aedes	1.00	1.00	1.00	37	Não mosquito	1.00	1.00	1.00	37	accuracy	1.00	1.00	1.00	118	macro avg	1.00	1.00	1.00	118	weighted avg	1.00	1.00	1.00	118	<p>Interpretação A acurácia do modelo treinado atingiu no mínimo a taxa definido nos requisito (0.95)</p> <p>Aceitação do modelo Sim</p> <p>Predição</p> <p>Quantidade de testes realizados 10 imagens novas testadas</p> <p>Acurácia dos testes Modelo acertou 10 das 10 imagens utilizadas no teste</p> <p>Exportação</p> <p>Link do modelo exportado</p>
	precision	recall	f1-score	support																																			
<i>aedes aegypti</i>	1.00	1.00	1.00	26																																			
Não aedes	1.00	1.00	1.00	37																																			
Não mosquito	1.00	1.00	1.00	37																																			
accuracy	1.00	1.00	1.00	118																																			
macro avg	1.00	1.00	1.00	118																																			
weighted avg	1.00	1.00	1.00	118																																			
<p>DESIGN DE INTERFACE Sketches</p>  <p>Teste de sketches</p> <p>O que funcionou Ideia geral do app</p> <p>Dúvidas do usuário Não sabia se tinha terminado a classificação</p> <p>O que pode ser melhorado Incluir texto explicativo para qual ação o usuário deve fazer quando o app classifica a imagem como <i>aedes-aegypti</i></p> <p>Ideias Permitir utilizar as fotos da galeria</p>		<p>DESIGN DE INTERFACE Prototipo de alta fidelidade</p> 																																					

Fonte: Autora (2022)

Quadro 20 - Caderno para documentação dos artefatos criados pelo estudante

DESCOBERTA	IDEAÇÃO	DESENVOLVIMENTO DO MODELO DE ML																																
Identificação da necessidade	Requisitos funcionais e de usabilidade	DESENVOLVIMENTO DO MODELO DE ML																																
<p>Quais são as categorias de usuário?</p>	<p>Requisitos funcionais</p> <table border="1"> <tr> <td>História de usuário</td> <td></td> </tr> <tr> <td>Funcionalidade(s)</td> <td></td> </tr> </table> <p>Requisitos de usabilidade</p> <table border="1"> <tr> <td>História de usuário</td> <td></td> </tr> <tr> <td>Eficiência</td> <td></td> </tr> <tr> <td>Eficiência</td> <td></td> </tr> <tr> <td>Satisfação</td> <td></td> </tr> </table>	História de usuário		Funcionalidade(s)		História de usuário		Eficiência		Eficiência		Satisfação		<p>Avaliação do desempenho</p> <table border="1"> <tr> <td>Acurácia</td> <td></td> </tr> <tr> <td>Matriz de confusão</td> <td></td> </tr> <tr> <td>Métricas</td> <td></td> </tr> <tr> <td>Interpretação</td> <td></td> </tr> <tr> <td>Aceitação do modelo</td> <td></td> </tr> <tr> <td>Predição</td> <td></td> </tr> <tr> <td>Quantidade de testes realizados</td> <td></td> </tr> <tr> <td>Acurácia dos testes</td> <td></td> </tr> <tr> <td>Exportação</td> <td></td> </tr> <tr> <td>Link do modelo exportado</td> <td></td> </tr> </table>	Acurácia		Matriz de confusão		Métricas		Interpretação		Aceitação do modelo		Predição		Quantidade de testes realizados		Acurácia dos testes		Exportação		Link do modelo exportado	
História de usuário																																		
Funcionalidade(s)																																		
História de usuário																																		
Eficiência																																		
Eficiência																																		
Satisfação																																		
Acurácia																																		
Matriz de confusão																																		
Métricas																																		
Interpretação																																		
Aceitação do modelo																																		
Predição																																		
Quantidade de testes realizados																																		
Acurácia dos testes																																		
Exportação																																		
Link do modelo exportado																																		
<p>Persona</p> <table border="1"> <tr> <td style="text-align: center;">[FOTO]</td> <td>Biografia</td> <td>Qual seu objetivo?</td> </tr> <tr> <td>[Nome] [Idade] [Cidade-UF]</td> <td>O que gosta?</td> <td>Conhecimento em TI</td> </tr> <tr> <td></td> <td colspan="2">O que não gosta?</td> </tr> </table>	[FOTO]	Biografia	Qual seu objetivo?	[Nome] [Idade] [Cidade-UF]	O que gosta?	Conhecimento em TI		O que não gosta?																										
[FOTO]	Biografia	Qual seu objetivo?																																
[Nome] [Idade] [Cidade-UF]	O que gosta?	Conhecimento em TI																																
	O que não gosta?																																	

Fonte: Autora (2022)

Junto com o curso, é apresentado como exemplo um app inteligente que visa a classificação de larvas de mosquitos *aedes-aegypti*. O app “Xô Dengue” apresenta três categorias de classificação: É *aedes-aegypti*, Não é *aedes-aegypti* e Não é mosquito, utilizando imagens capturadas pela câmera do celular. Os objetivos do modelo de ML são documentados conforme o Quadro 21. Como parte deste trabalho foi desenvolvido um protótipo desse modelo de ML, a ser aperfeiçoado no trabalho de conclusão de curso de Manarin (2023).

Quadro 21 - Objetivo do modelo de ML

Objetivo do modelo de ML	
Tarefa	Classificar a categoria da larva do mosquito por imagens capturada pela câmera de um celular Android
Tipo da tarefa	<i>Single-label</i> classificação de imagens
Categorias	2 categorias de mosquitos e uma para imagens não representando mosquitos
Contexto de uso	O modelo é utilizado como exemplo no contexto de ensino de na Educação Básica.
Público alvo	Cidadãos (15+ anos) Foco em alunos do Ensino Médio

Fonte: Autora (2022)

Quadro 22 - Conjunto de dados

Conjunto de dados		
Fonte de dados (coleta própria/uso de conjunto de dados pré-existente, p.ex. do kaggle)	Conjunto de dados de larvas coletadas no LACEN e Departamento de Biologia/UFSC.	
Padronização das imagens	Formato: .jpeg	Tamanho: 224x224 pixels
Rotulação dos dados	Por biólogos	

Fonte: Autora (2022)

Figura 12 - É aedes-aegypti



Fonte: Autora (2022)

Figura 13 - Não é aedes-aegypti



Fonte: Autora (2022)

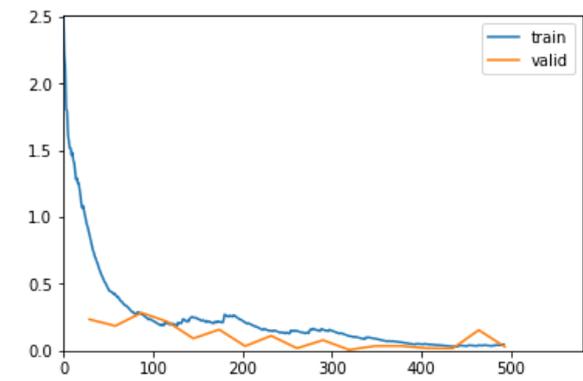
Figura 14 - Não é mosquito



Fonte: Autora (2022)

O treinamento do modelo foi realizado no VULCAN, implementado no Google Colab. Os parâmetros utilizados são apresentados no quadro 23. Foi obtida uma acurácia de 100% como resultado do treinamento. Uma análise mais detalhada desse desempenho atingido é prevista como parte do TCC de Manarin (2023).

Quadro 23 - Treinamento

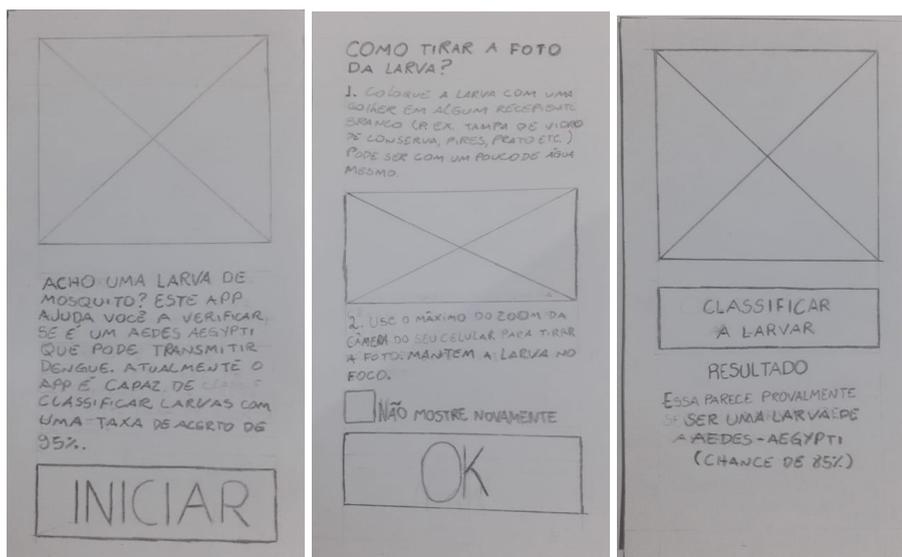
Treinamento																								
Quantidade de épocas	15																							
Taxa de aprendizagem	0.994																							
Tamanho do conjunto de dados	174 Aedes-aegypti 168 Não Aedes 250 Não é mosquito																							
Curva de loss																								
Avaliação do desempenho																								
Acurácia	1.00																							
Matriz de confusão	<p style="text-align: center;">Confusion matrix</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tbody> <tr> <td rowspan="3" style="vertical-align: middle;">Actual</td> <td>Aedes aegypti</td> <td style="background-color: #4682b4; color: white;">26</td> <td style="background-color: #e6f2ff;">0</td> <td style="background-color: #e6f2ff;">0</td> </tr> <tr> <td>Não aedes</td> <td style="background-color: #e6f2ff;">0</td> <td style="background-color: #4682b4; color: white;">35</td> <td style="background-color: #e6f2ff;">0</td> </tr> <tr> <td>Não mosquito</td> <td style="background-color: #e6f2ff;">0</td> <td style="background-color: #e6f2ff;">0</td> <td style="background-color: #000080; color: white;">57</td> </tr> <tr> <td></td> <td></td> <td style="writing-mode: vertical-rl; transform: rotate(180deg);">Aedes aegypti</td> <td style="writing-mode: vertical-rl; transform: rotate(180deg);">Não aedes</td> <td style="writing-mode: vertical-rl; transform: rotate(180deg);">Não mosquito</td> </tr> <tr> <td></td> <td></td> <td colspan="3" style="text-align: center;">Predicted</td> </tr> </tbody> </table>	Actual	Aedes aegypti	26	0	0	Não aedes	0	35	0	Não mosquito	0	0	57			Aedes aegypti	Não aedes	Não mosquito			Predicted		
Actual	Aedes aegypti		26	0	0																			
	Não aedes		0	35	0																			
	Não mosquito	0	0	57																				
		Aedes aegypti	Não aedes	Não mosquito																				
		Predicted																						

Métricas				
	precision	recall	f1-score	support
Aedes aegypti	1.00	1.00	1.00	26
Não aedes	1.00	1.00	1.00	35
Não mosquito	1.00	1.00	1.00	57
accuracy			1.00	118
macro avg	1.00	1.00	1.00	118
weighted avg	1.00	1.00	1.00	118

Fonte: Autora (2022)

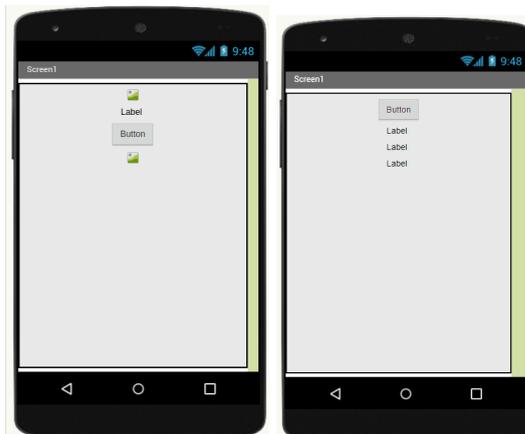
Na construção do aplicativo inteligente, foram criados *sketchs* para cada tela, e, a partir deles, foram feitos os *wireframes* utilizando a ferramenta Sketch2aia. Definindo a paleta de cores, tipografia e imagens, foram projetadas também as telas de alta fidelidade.

Figura 15 - Sketchs do app “Xô Dengue”



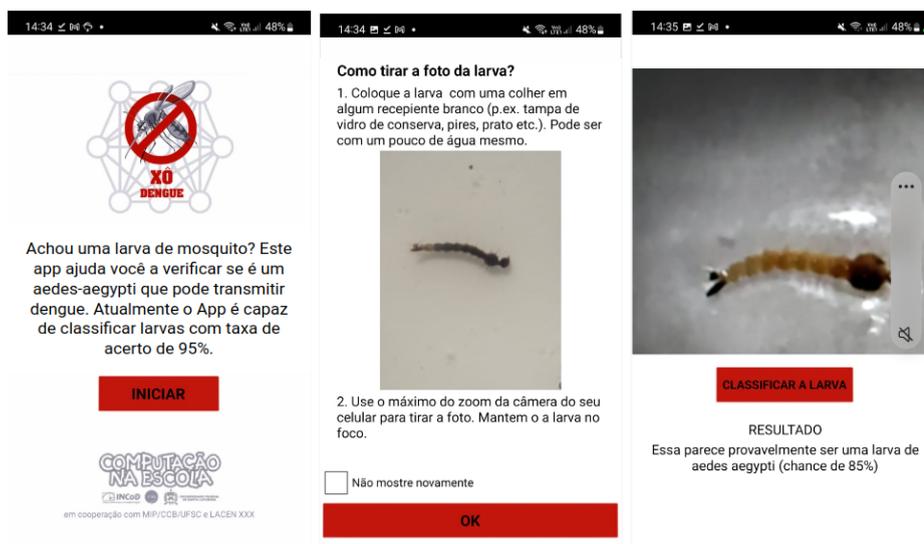
Fonte: Autora (2022)

Figura 16 - Wireframe do app “Xô Dengue”



Fonte: Autora (2020)

Figura 17 - Protótipo de alta fidelidade do app “Xô Dengue”



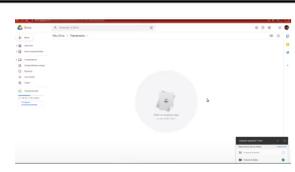
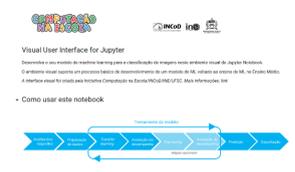
Fonte: Autora (2020)

4.2.4 Ferramentas utilizadas no curso

Este curso utilizou ferramentas online que são gratuitas e/ou de código aberto, o que facilita o processo de aprendizagem dos alunos, conforme apresentado no quadro 24.

Quadro 24 - Ferramentas utilizadas no curso

Ferramenta	Função	Exemplo	Link
------------	--------	---------	------

Google Drive	Armazenamento de conjunto de dados		https://www.google.com/intl/pt-BR/drive/
VULCAN	Treinamento de rede neural para classificação de imagens		https://colab.research.google.com/drive/14wfLb8ZmCM8lqjM67bjSo-xKMUBeXzg7
Sketch2aia	Prototipagem das interfaces de usuário		http://apps.computacaonaescola.ufsc.br/s2a/howto
App Inventor	Ambiente de programação visual para programar apps		https://appinventor.mit.edu/
CodeMaster	é uma ferramenta, que analisa e avalia projetos de programação desenvolvidos com App Inventor de forma automatizada		http://apps.computacaonaescola.ufsc.br/code/master
Extensão do App Inventor para implantação do modelo de ML	é uma extensão para classificar objetos utilizando modelos treinados de ML no VULCAN		Em desenvolvimento

Fonte: Autora (2022)

4.2.5 Avaliação da Aprendizagem

A avaliação do desempenho em relação às principais atividades práticas do curso do processo de ML, realiza-se por meio do modelo proposto por Gresse von Wangenheim *et al.* (2021a).

Quadro 25 - Rubrica de avaliação de aprendizagem com base no desempenho

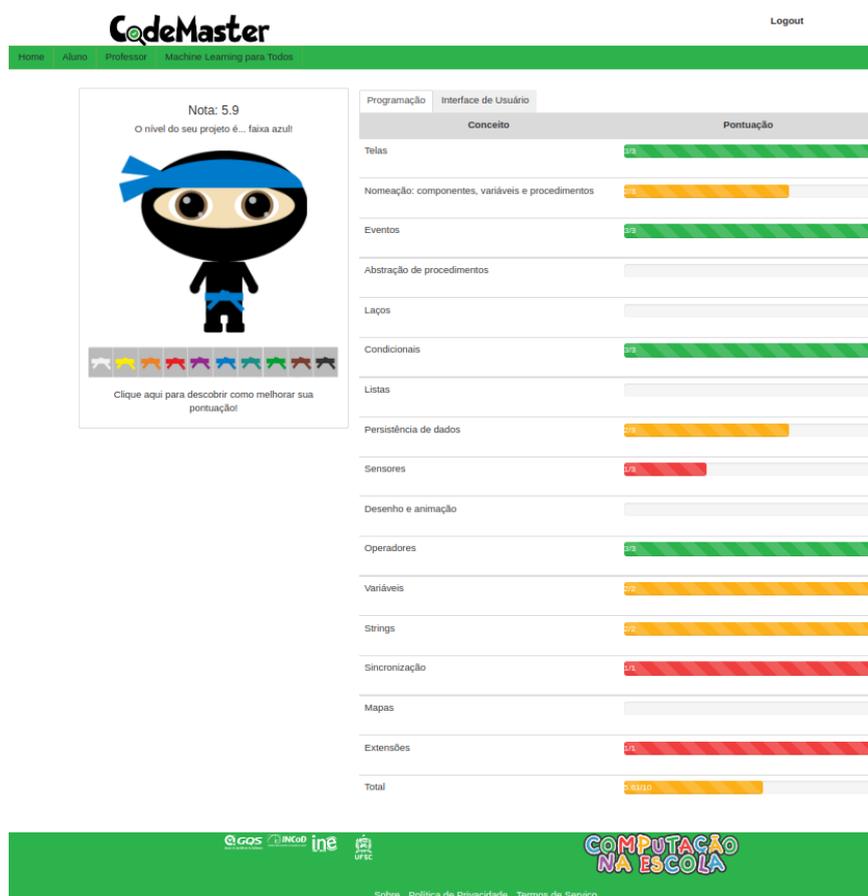
ID	Critério	Níveis de desempenho		
		Baixo - 0 pt	Aceitável - 1 pt.	Bom - 2 pt.
Preparação de dados				
C1	Quantidade de imagens	Menos de 5 imagens por categoria	6 de 10 imagens por categoria	Mais de 10 imagens por categoria
C2	Distribuição do conjunto de dados	Quantidade de imagens por categoria varia muito	Quantidade de imagens por categoria varia pouco	Todas as categorias possuem a mesma quantidade de imagens
Preparação de dados				
C3	Rotulagem das imagens	Menos de 20% das imagens rotuladas corretamente	De 20% a 99% das imagens rotuladas corretamente	Todas as imagens rotuladas corretamente
Treinamento de modelo de ML/Transfer Learning e Fine-Tuning				
C4	Treinamento - <i>Transfer Learning</i>	O modelo não foi treinado (<i>transfer learned</i>)	O modelo foi treinado com os parâmetros padrão	O modelo foi treinado com parâmetros ajustados (arquitetura, época e taxa de aprendizagem)
C5	Treinamento - Ajuste Fino	O modelo não sofreu o ajuste fino	Foi feito <i>unfreeze</i> das camadas e melhor taxa de aprendizagem não encontrada ou modelo não treinado	Foi feito <i>unfreeze</i> das camadas, a melhor taxa de aprendizagem foi encontrada e o modelo sofreu o ajuste fino
Avaliação e interpretação do desempenho de um modelo de ML (Transfer Learning e Fine-Tuning)				
C6	Interpretação de acurácia	Categorias com baixa acurácia não identificadas	Categorias com baixa acurácia identificadas e interpretação incorreta em relação ao modelo	Categorias com baixa acurácia identificadas corretamente e interpretação correta em relação ao modelo
C7	Interpretação da matriz de confusão	Classificações incorretas não identificadas e interpretação incorreta em relação ao modelo	Classificações incorretas identificadas e interpretação incorreta em relação ao modelo	Classificações incorretas identificadas e interpretação correta em relação ao modelo
C8	Ajustes/melhorias feitas	Sem novas iterações de desenvolvimento	Uma nova iteração com alterações no conjunto de dados e/ou parâmetros de treinamento	Diversas novas iterações com alterações no conjunto de dados e/ou parâmetros de treinamento
Avaliação e interpretação do desempenho de um modelo de ML/Testar e aperfeiçoar programas				
C9	Testes com	Nenhum novo objeto	1-2 novos objetos	Mais de dois novos

	novos objetos	testado	testados	objetos testados
C10	Interpretação dos testes	Interpretação errada	---	Interpretação correta

Fonte: Adaptado de Gresse von Wangenheim *et al.* (2021a)

O app a ser desenvolvido pode ser avaliado em relação aos conceitos de algoritmos e programação utilizando o CodeMaster com base na rubrica proposta por Alves *et al.* (2020).

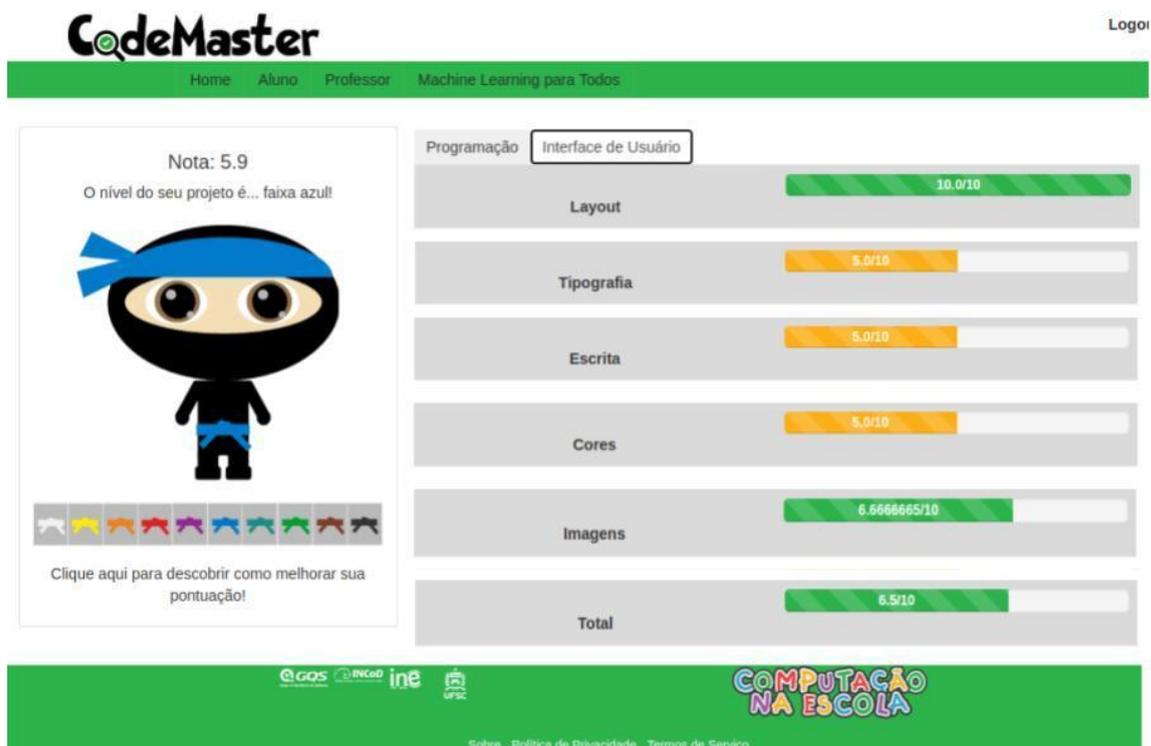
Figura 18 - Exemplo do módulo Avaliação de conceitos de algoritmos e programação no CodeMaster



Fonte: ALVES (2020)

Também pode ser avaliado o design de interface, utilizando o módulo de design visual do CodeMaster, a partir da rubrica proposta por Solecki *et al.* (2020).

Figura 19 - Exemplo do módulo Avaliação do design visual no CodeMaster



Fonte: SOLECKI *et al* (2020)

O curso está atualmente disponível no espaço de professor no moodle UFSC, e será publicado de forma gratuita no site de cursos da iniciativa Computação na Escola/INCoD/INE/UFSC.

5 CONCLUSÃO

Neste trabalho foi desenvolvido um curso de criação de aplicativo inteligente, utilizando o ensino de ML com ênfase em classificação de imagens para alunos do Ensino Médio em escolas brasileiras. Na primeira etapa do projeto, foi definido a fundamentação teórica sobre o tema deste projeto (O1). Na etapa seguinte foi levantado o estado da arte de unidades instrucionais que ensinam como criar sistemas inteligentes neste estágio educacional, mostrando que atualmente ainda existe lacuna no ensino de *ML* (O2). Na terceira etapa, com as informações levantadas no estado da arte, foi desenvolvido o curso, no qual a partir da análise de contexto, definiram-se os objetivos de aprendizagem e plano de ensino (O3). Com base no plano de ensino foi desenvolvido o material didático do curso, incluindo slides, vídeos, material necessário para que os alunos pudessem aprender a desenvolver o seu próprio aplicativo inteligente para resolver algum problema da sua comunidade.

Com esse curso, espera-se atingir um público amplo no ensino de ML na educação brasileira e incentivar o aluno a desenvolver suas habilidades em áreas de conhecimento relacionadas à ML.

Para trabalhos futuros, sugere-se aplicar esse curso com objetivo de avaliar a sua qualidade e melhorá-lo com base no feedback dessas aplicações.

REFERÊNCIAS

AI4K12. Disponível em: <https://ai4k12.org/>. Acesso em: 17 jun. 2022.

ALVES, N. da C., GRESSE VON WANGENHEIM, C., HAUCK, J. C. R., BORGATTO, A. F. A **Large-scale Evaluation of a Rubric for the Automatic Assessment of Algorithms and Programming Concepts**. In: Proc. of the 51st ACM Technical Symposium on Computer Science Education, Portland/USA, 2020, Pages 556–562.

ANDERSON, Lorin W.; Krathwohl, David R., eds. (2001). Uma taxonomia para aprender, ensinar e avaliar: Uma revisão da taxonomia de objetivos educacionais de Bloom. Nova York: Longman. ISBN 978-0-8013-1903-7.

AMERSHI, S. *et al.* Software Engineering for Machine Learning: A Case Study. In: Proc. of the International Conference on Software Engineering: Software Engineering In Practice, Montreal, Qc, Canada. 2019, 291 - 300.

BRANCH, R. M. **Instructional Design: The ADDIE Approach**. New York: Springer Science & Business Media, 2009.

BRASIL. **Lei nº 9394**, de 20 de dezembro de 1996. Estabelece as diretrizes e bases da educação nacional. Brasília, DF, Disponível em: http://www.planalto.gov.br/ccivil_03/leis/L9394compilado.htm. Acesso em: 01 set. 2019.

BRASIL. Ministério da Educação. **Base Nacional Comum Curricular**. 2018. Disponível em: http://basenacionalcomum.mec.gov.br/images/BNCC_EI_EF_110518_versaofinal_site.pdf. Acesso em: 26 jul. 2019.

BRASIL. Ministério da Educação. **Diretrizes Curriculares Nacionais da Educação Básica**. 2013. Disponível em: http://portal.mec.gov.br/index.php?option=com_docman&view=download&alias=13448-diretrizes-curriculares-nacionais-2013-pdf&Itemid=30192. Acesso em: 10 set. 2019.

BROWN, T. Design Thinking. **Harvard Business Review**, 1(1), 2008, 1-10.

CAMBRAIA, A.C.; SCAICO, P. D. Os desafios da Educação em Computação no Brasil: um relato de experiências com Projetos PIBID no Sul e Nordeste do país. **Revista Espaço Acadêmico**, 148(13), 2013, 01-09.

CARDOZO, J. **Desenvolvimento de um Curso On-line para o Ensino de Machine Learning voltado à Classificação de Imagens no Ensino Médio**. 2022. TCC (Graduação) - Curso de Sistemas de Informação, INE, UFSC, Florianópolis, 2022.

CETIC.BR. **Microdados TIC Kids Online Brasil - 2019 - Pais e responsáveis**. São Paulo: Cetic, 2019. Disponível em: <https://cetic.br/pt/arquivos/kidsonline/2019/pais/>. Acesso em: 05 jul. 2020.

COMPUTER SCIENCE TEACHERS ASSOCIATION. **CSTA K–12 CS Standards**. 2019. Disponível em: <https://csteachers.org/page/standards/>. Acesso em: 10 ago. 2019.

FERREIRA, M. *et al.* Ensinando Design de Interface de Usuário na Educação Básica: um mapeamento sistemático do estado da arte e prática. In: Anais do Workshop de Informática na Escola, Porto Alegre, Brasil, 2019, 511-520.

FRANZ, A. C. M.. **Desenvolvimento de uma ferramenta visual de classificação de imagens para o ensino de machine learning no ensino médio**. TCC (Graduação) - Curso de Sistemas de Informação, INE, UFSC, Florianópolis, 2021.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep learning**. Cambridge, MA: MIT Press, 2016.

GRESSE VON WANGENHEIM, C.; VON WANGENHEIM, A.; PACHECO, F. S.; FERREIRA, M. N.; HAUCK, J. C. R. **5 Minutos de Programação em Exibições**. In: Anais do 25° Workshop sobre Educação em Computação, São Paulo, Brazil, 2017.

GRESSE VON WANGENHEIM, C. *et al* (a). A Proposal for Performance-based Assessment of the Learning of Machine Learning Concepts and Practices in K-12. **Informatics In Education**, online, 2021, 1-21.

GRESSE VON WANGENHEIM, C.; HAUCK, J. C. R.; PACHECO, F. S.; BERTONCELI BUENO, M. F (b). Visual Tools for Teaching Machine Learning in K-12: A Ten-Year Systematic Mapping, **Education and Information Technologies**, 26(5), 5733-5778, 2021.

GROVER, S.; PEA, R. Computational Thinking in K–12. **Educational Researcher**, 42(1), 2013, 38-43.

IMB SKILLSBUILD FOR STUDENTS. Module 5: Bringing It All Together. 2020. Disponível em: https://ole03.yourlearning.ibm.com/mod/scorm/player.php?a=64¤torg=d2l_org&scoId=329&display=popup&mode=normal&lang=en. Acesso em: 22 ago. 2021.

LEE, I. *et al.* Computational thinking for youth in practice. **ACM Inroads**, 2(1), 2011.

LONG, D; MAGERKO, B. What is AI Literacy?: competencies and design considerations. In: **Proc. of the Chi Conference on Human Factors in Computing Systems**, ACM, 2020, 1-16.

MANARIN , B. **Desenvolvimento de um Modelo de Classificação de Imagens de Larvas de Mosquitos usando Deep Learning**. em desenvolvimento 2023. TCC (Graduação) - Curso de Sistemas de Informação, INE, UFSC, Florianópolis, 2023.

MARQUES, L. S.; GRESSE VON WANGENHEIM, C.; HAUCK, J. C. R.. **Introdução à Machine Learning**: desenvolvimento de uma unidade instrucional para ensino de ML na educação básica. Relatório Técnico INCoD/INE/UFSC, Florianópolis, Brasil, 2020.

MIKE, K.; HAZAN, T.; HAZZAN, O. Equalizing Data Science Curriculum for Computer Science Pupils. In: Proc. of the 20th Koli Calling International Conference On Computing Education Research, ACM, 2020, 1-5.

MIT (a). **About us**. Disponível em: <http://appinventor.mit.edu/explore/about-us.html>. Acesso em: 17 jun. 2022.

MIT (b). **Beginner Video Tutorials**. Disponível em: <http://appinventor.mit.edu/explore/ai2/beginner-videos.html>. Acesso em: 17 jun. 2022.

MIT (c). **Events**. Disponível em: <https://appinventor.mit.edu/explore/events>. 17 jun. 2022.

MIT (d). **Magic 8-Ball Tutorial**. Disponível em: <http://appinventor.mit.edu/explore/teach/magic-8-ball.html>. Acesso em: 17 jun. 2022.

NICHOLSON, C. **A beginner's guide to neural networks and deep learning**. 2019. Disponível em: <https://pathmind.com/wiki/neural-network>. Acesso em: 10 maio 2019.

PATTON, E W.; TISSENBAUM, M.; HARUNANI, F. MIT App Inventor: objectives, design, and development. **Computational Thinking Education**, 1(1), 31-49.

PETERSEN, K.; VAKKALANKA, S.; KUZNIARZ, L. Guidelines for conducting systematic mapping studies in software engineering: an update. **Information And Software Technology**, 64(1), 2015, 1-18.

PINHEIRO, F. d. C.; GRESSE VON WANGENHEIM, C.; MISSFELDT FILHO, R. Teaching software engineering in K-12 education: a systematic mapping study. **Informatics In Education**, 17(2), 2018, 167-206.

PROJECT JUPYTER. **About Us**: Project Jupyter's origins and governance. 2022. Disponível em: <https://jupyter.org/about>. Acesso em: 17 jun. 2022.

QEDU. **QEDU Analítico**. Disponível em: <https://analitico.qedu.org.br/>. Acesso em 04 de nov. 2021.

RUSSELL, S; NORVIG, P. **Artificial intelligence: a modern approach**. 3. ed. Chennai, TN, India: Pearson Education India, 2015.

SAS. **Redes neurais**: o que são e qual sua importância?. 2019. Disponível em: https://www.sas.com/pt_br/insights/analytics/neural-networks.html#technical. Acesso em: 10 nov. 2019.

SOCIEDADE BRASILEIRA DE COMPUTAÇÃO. **Referenciais de formação em computação**: educação básica. Porto Alegre: SBC, 2017. Disponível em: <http://www.sbc.org.br/files/ComputacaoEducacaoBasica-versaofinal-julho2017.pdf>. Acesso em: 10 ago. 2019.

SOLECKI, I.; PORTO, J. A.; ALVES, N. d. C., GRESSE VON WANGENHEIM, C., HAUCK, J. C. R., BORGATTO, A. F. **Automated Assessment of the Visual Design of Android Apps Developed with App Inventor**. In: Proc. of the 51st ACM Technical Symposium on Computer Science Education, Portland, USA, 2020, 51–57.

SPADINI, A. S. **Principais frameworks de deep learning em 2021**. 2021.

Disponível em:

<https://www.alura.com.br/artigos/principais-frameworks-de-deep-learning-em-2021>.

Acesso em: 15 jul. 2021.

SPERLING, A.; LICKERMAN, D.. Integrating AI and machine learning in software engineering course for high school students. In: Proc. of the Annual Conference on Innovation and Technology in Computer Science Education, ACM, 2012.

TISSENBAUM, M.; SHELDON, J.; ABELSON, H. From computational thinking to computational action. **Communications Of The ACM**, 62(3), 2019, 34-32.

TOURETZKY, D. S. *et al.* **K-12 Guidelines for Artificial Intelligence**: what students should know. 2019. Disponível em:

https://github.com/touretzkyds/ai4k12/raw/master/documents/ISTE_2019_Presentation_website_final.pdf. Acesso em: 06 set. 2019.

WIKISTAT. **Neural Networks and introduction to Deep Learning**. Disponível em:

<https://www.math.univ-toulouse.fr/~besse/Wikistat/pdf/st-m-hdstat-rnn-deep-learning.pdf> Acesso: 15 jul. 2021.

YU, Y.; CHEN, Y. Design and Development of High School Artificial Intelligence Textbook Based on Computational Thinking. **Open Access Library Journal**, 5(9), 2018, 1-15.

APÊNDICE A - ARTIGO NO FORMATO SBC

Ensinando *Machine Learning* no Ensino Médio no nível Create com VULCAN

**Mirian de França Santos Pereira¹, Ramon Mayor Martins¹,
Christiane Gresse von Wangenheim¹**

¹Departamento de Informática e Estatística – Universidade Federal de Santa Catarina (UFSC)
Caixa Postal 476 – 88.040-900 – Florianópolis – SC – Brazil

mirian.f@grad.ufsc.br, ramon.mayor@posgrad.ufsc.br, c.wangenheim@ufsc.br

Abstract. *Artificial intelligence (AI) technology has become increasingly used today, helping and improving people's lives, which also requires the understanding of such technologies by the population. Thus, it is important to include the teaching of AI concepts in public schools. In this respect this article presents a course to teach ML concepts on the create stage in High School. The results of this research can be used in the context of K-12 with the objective of popularizing computing and AI among high school students.*

Resumo. *A tecnologia de inteligência artificial (IA) tem se tornado cada vez mais utilizada na atualidade, auxiliando e melhorando a vida das pessoas, o que exige também a compreensão de tais tecnologias pela população. Assim, é importante incluir o ensino de conceitos de IA nas escolas públicas. Nesse sentido, este artigo apresenta um curso para ensinar conceitos de ML na etapa de criação no Ensino Médio. Os resultados desta pesquisa podem ser usados no contexto do K-12 com o objetivo de popularizar a computação e a IA entre alunos do ensino médio.*

1. Introdução

Com o avanço tecnológico ocorrido principalmente no final do século XX e no início do século XXI, contando com tecnologias voltadas para o uso de Inteligência Artificial (IA) que vêm tornando-se mais presentes no dia a dia na sociedade em geral, surge a necessidade do ensino de IA no currículo educacional [Yu and Chen, 2018]. A Sociedade Brasileira de Computação (SBC) e a Base Nacional Comum Curricular (BNCC) vêem a necessidade de ensinar sobre IA na educação básica [Sociedade Brasileira de Computação, 2017] [Brasil, 2019]. No Ensino Médio é um momento que os estudantes estão preparados para solucionar problemas complexos na área de IA, tendo a possibilidade de incentivar os estudantes a seguir na área da tecnologia.

Sendo um subcampo da Inteligência Artificial, o Machine Learning (ML) estuda o reconhecimento de padrões e a teoria do aprendizado computacional. O ML explora o estudo e construção de algoritmos que possuem a capacidade de aprender a partir de seus erros e fazer previsões sobre dados [Russell and Norvig, 2015].

Assim, esse trabalho visa desenvolver um curso para ensinar, conceitos de Machine Learning e habilidades do século XXI de forma dinâmica e motivadora para popularizar este conhecimento já no Ensino Médio. Espera-se que o curso desenvolvido a partir deste trabalho possa ser aplicado amplamente em escolas do Brasil,

popularizando não só os conceitos de programação, mas também os conceitos de Inteligência Artificial e Machine Learning como elementos do aprendizado de computação.

2. Trabalhos correlatos

Como resultado da análise do estado da arte, observou-se que não existem muitas unidades instrucionais voltadas ao ensino da ML, e esse número diminui ainda mais quando focamos no ensino de ML voltado para o nível *create* no Ensino Médio, com apenas três unidades instrucionais encontradas no período de 2020 - 2021. Essas unidades instrucionais focam em ensinar o processo de desenvolvimento ML concentrando-se nas principais fases do processo e com a criação dos seus próprios projetos.

Observou-se, também, que há iniciativas que podem ser ensinadas no Ensino Fundamental [IBM, 2020], porém existem mais unidades institucionais voltadas ao Ensino Médio [Sperling et al. 2012] [Mike et al. 2020]. Nenhum dos autores relatou dificuldade ou baixa aprendizagem no ensino de ML. Nos trabalhos analisados notou-se que os estudantes conseguiram desenvolver as atividades propostas nos cursos presenciais e no online.

Outra característica identificada foi a utilização das ferramentas Python, Jupyter Notebooks, Google Colab e Teachable Machine para as atividades no ensino de ML [IBM, 2020] [Mike et al. 2020].

Em resumo, foi possível identificar a existência de poucas unidades instrucionais que ensinam o processo de desenvolvimento de ML no Ensino Médio no nível *create*, no entanto todos os autores relataram que não tiveram dificuldade em ensinar.

3. Metodologia de pesquisa

O objetivo geral deste trabalho é desenvolver um curso online para estudantes do Ensino Médio que os ensine a criar seus próprios apps “inteligentes” para classificação de imagens usando Machine Learning. Foi utilizado o modelo ADDIE [Branch, 2009] para o desenvolvimento do método de design instrucional.

A primeira etapa foi analisar o contexto em termos de necessidades, características de aprendizes e o ambiente. Seguindo para a etapa foi definido o conteúdo do curso, as estratégias instrucionais e o plano de ensino. E em seguida foi desenvolvido o material didático e a avaliação de aprendizagem do aluno.

4. O Curso MLCreate

O curso MLCreate visa ensinar a aplicar os conceitos básicos de Machine Learning, levando os estudantes do ensino médio a criar o seu próprio aplicativo inteligente de classificação de imagens. O curso é voltado para estudantes do ensino médio que já têm o conhecimento básico de *Machine Learning*, sendo destinado à intermediários.

4.1. Objetivos de Aprendizagem

Os objetivos de aprendizagem do curso foram criados a partir do *Framework* CSTA K-12 CS [CSTA, 2019] e The Artificial Intelligence for K-12 [AI4K12, 2019], também tendo objetivos de *Design Thinking*, algoritmo e programação. O curso, em sua forma completa, visa ensinar a identificação de um problema, idealização de uma solução por meio de um app inteligente, o desenvolvimento de um modelo de ML e a sua implantação em um app. Porém como atualmente o desenvolvimento de uma extensão do App Inventor para permitir a implantação dentro de um app está ainda em desenvolvimento, esta versão do curso se enfoca na parte inicial voltada aos passos de identificação de problema, idealização de solução e desenvolvimento do modelo de ML, a ser complementado no futuro pelas demais partes do curso.

Tabela 1. Objetivos de aprendizagem do curso MLCreate

(todos os objetivos marcados em cinza se referem a uma parte futura do curso tendo como pré-requisito a disponibilidade de uma extensão para App Inventor)

Objetivo de aprendizagem	Referência(s)
OA1. Identificar e resolver problemas criando sistemas de software inteligentes	[CSTA, 2019][CSTA, 2017][AIGA, 2013]
OA2. Compreender o processo do Design Thinking	
OA3. Criar protótipos que usam algoritmos para resolver problemas computacionais.	[CSTA, 2019]
OA4. Preparar um conjunto de dados para o treinamento do modelo de ML, passados pelas suas etapas	[AI4K12 2019]
OA5. Treinar um modelo de ML de classificação de imagem	[AI4K12 2019]
OA6. Avaliar o desempenho do modelo de ML treinado	[AI4K12 2019]
OA7. Exportar o modelo de ML treinado	
OA8. Explicar o conceito de um processo de ML e fornecer um exemplo, ilustrando suas fases, incluindo as entregas que são produzidas	[ACM/IEEE, 2013][UXQB, 2018]
OA9. Analisar o contexto de sistemas de software inteligente em termos de usuários, tarefas, dispositivos e ambientes de uso	[AIGA, 2013] [ISO/IEC, 2019] [CSTA, 2017] [ACM/IEEE 2013]
OA10. Especificar requisitos de sistemas de software inteligente em termos de funcionalidade e usabilidade	[CSTA, 2017] [ACM/IEEE, 2013]
OA11. Criar protótipos de sistemas de software inteligente em diferentes níveis (esboços, baixa fidelidade, alta fidelidade, funcional)	[CSTA, 2017] [ACM/IEEE, 2013]
OA12. Projetar o design visual (cores, tipografia, ícones, imagens, etc.) do sistema de software inteligente	[ACM/IEEE, 2013] [CSTA, 2017] [GARRET, 2011]
OA13. Testar e refinar um sistema de software inteligente para funcionalidade e usabilidade	[CSTA, 2017]
OA14. Compartilhar o sistema de software inteligente desenvolvido	[CSTA, 2017] [LEE et al., 2007]

4.2. Design do Curso

Com base na análise do contexto e nos objetivos de aprendizagem, o curso foi projetado para o nível intermediário. O curso visa ensinar a identificação de um problema, idealização de uma solução por meio de um app inteligente, o desenvolvimento de um modelo de ML e a sua implantação em um app. Focado para estudantes do ensino médio com conhecimento básico sobre *Machine Learning* que tenham acesso a internet e *smartphone*. Projetado para ser um curso de curta duração, sendo planejado a duração de 23 horas. O curso foi projetado para ser aplicado à distância ou presencial.

Adota-se principalmente uma metodologia ativa para levar os estudantes a aplicar os conceitos de computação e ML. Enfocando ao nível de *create* o curso guia o aluno passo-a-passo a desenvolver o seu próprio aplicativo inteligente voltado a classificação de imagens adotando *design thinking*. Adotando a ação computacional o curso de forma interdisciplinar também orienta o aluno a desenvolver um aplicativo para resolver uma necessidade identificada na sua comunidade. Como app exemplo é apresentado um app para a classificação de imagens de larvas de mosquito visando a prevenção de focos do mosquito *aedes aegypti* para diminuir os casos de dengue. Para as atividades práticas, foi utilizada a ferramenta visual VULCAN dentro do Jupyter Notebook [Franz, 2021].

Tabela 2. Plano de ensino do curso MLCreate.

Aula	Conteúdo	Objetivos de Aprendizagem	Método Instrucional	Material Instrucional	Avaliação
Introdução					
1	Motivação	OA2		Slides	
Análise					
2	Descoberta	OA1, OA2	Atividade prática, discussão	Slides, Google Drive	
3	Idealização da proposta	OA1, OA2, OA9	Atividade prática, discussão	Slides, Google Drive	
Desenvolvimento do modelo de ML					
4.1	Preparação dos dados (coleta, rotulação e pré-processamento)	OA4	Atividade prática	Slides, VULCAN, Google Drive	Rubrica de ML
4.2	Treinamento, avaliação do desempenho do modelo	OA5, OA6	Atividade prática	Slides, VULCAN, Google Drive	Rubrica de ML
4.3	Treinamento, avaliação na etapa de ajuste fino e exportação	OA5, OA6, OA7	Atividade prática	Slides, VULCAN, Google Drive	Rubrica de ML
Construção e teste do app inteligente com App Inventor					
5.1	Criar e testar sketches	OA11	Atividade prática, Vídeo	Slides, Google drive	
5.2	Programar e testar o wireframe	OA11, OA12	Atividade prática, Vídeo	Slides, Google drive	
5.3	Criar o design visual	OA12	Atividade prática	Slides, Google drive	Codemaster - design visual
5.4	Implantar o modelo de ML	OA11, OA12, OA13	Atividade prática, App Inventor	Slides, Google drive	

5.5	Programar e testar o app funcional	OA13	Atividade prática, Vídeos	Slides, Google drive	CodeMaster - Algoritmos & programação
6	Compartilhamento do aplicativo (via .apk)	OA14	Atividade prática	Slides, Google drive	

4.3. Material Didático

Com base no plano de ensino foi desenvolvido o material didático composto por slides, vídeos para apresentar os conceitos e guiar o estudante a desenvolver o seu app inteligente passo-a-passo.

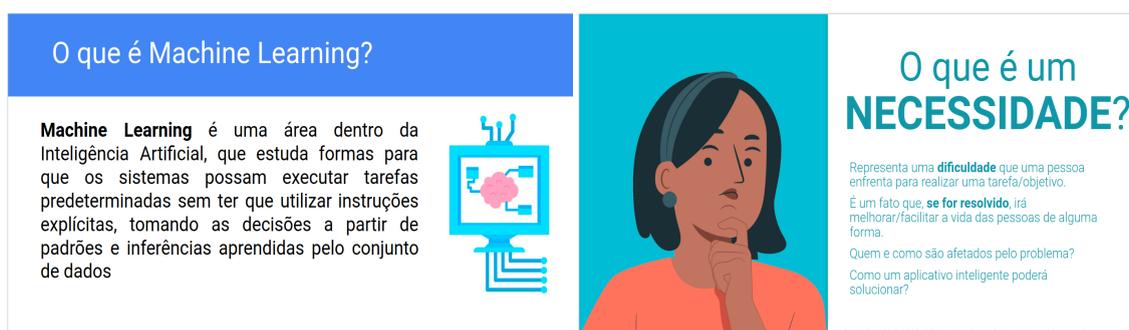


Figura 1. Exemplos de slides

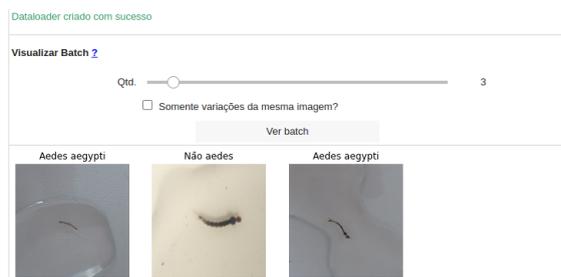


Figura 2. Exemplo do ambiente visual VULCAN dentro de um Jupyter Notebook no Google Colab

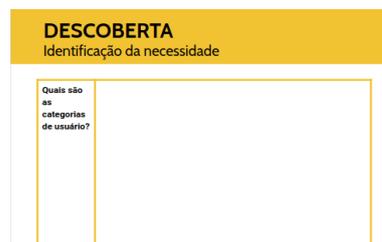


Figura 3. Caderno do estudante

Foram projetados também modelos (*templates*) de artefatos a serem criados pelos alunos em forma de Caderno (Google Docs) e Caderno exemplo apresentando o desenvolvimento do app Xô Dengue (Google Docs). O Caderno projetado apresenta os templates de todos os artefatos a serem criados ao longo do curso.

4.4. Avaliação da aprendizagem

Para a avaliação do desempenho em relação às principais atividades práticas do curso do processo de ML, realiza-se por meio do modelo de avaliação de desempenho proposto por Gresse von Wangenheim et al. (2021a).

Tabela 3. Rubrica de avaliação de aprendizagem com base no desempenho

ID	Critério	Níveis de desempenho		
		Baixo - 0 pt.	Aceitável - 1 pt.	Bom - 2 pt.
Preparação de dados				
C1	Quantidade de imagens	Menos de 5 imagens por categoria	6 de 10 imagens por categoria	Mais de 10 imagens por categoria

C2	Distribuição do conjunto de dados	Quantidade de imagens por categoria varia muito	Quantidade de imagens por categoria varia pouco	Todas as categorias possuem a mesma quantidade de imagens
C3	Rotulagem das imagens	Menos de 20% das imagens rotuladas corretamente	De 20% a 99% das imagens rotuladas corretamente	Todas as imagens rotuladas corretamente
Treinamento de modelo de ML/Transfer Learning e Fine-Tuning				
C4	Treinamento - Transfer Learning	O modelo não foi treinado (transfer learned)	O modelo foi treinado com os parâmetros padrão	O modelo foi treinado com parâmetros ajustados (arquitetura, época e taxa de aprendizagem)
C5	Treinamento - Ajuste Fino	O modelo não sofreu o ajuste fino	Foi feito unfreeze das camadas e melhor taxa de aprendizagem não encontrada ou modelo não treinado	Foi feito unfreeze das camadas, a melhor taxa de aprendizagem foi encontrada e o modelo sofreu o ajuste fino
Avaliação e interpretação do desempenho de um modelo de ML (Transfer Learning e Fine-Tuning)				
C6	Interpretação de acurácia	Categorias com baixa acurácia não identificadas	Categorias com baixa acurácia identificadas e interpretação incorreta em relação ao modelo	Categorias com baixa acurácia identificadas corretamente e interpretação correta em relação ao modelo
C7	Interpretação da matriz de confusão	Classificações incorretas não identificadas e interpretação incorreta em relação ao modelo	Classificações incorretas identificadas e interpretação incorreta em relação ao modelo	Classificações incorretas identificadas e interpretação correta em relação ao modelo
C8	Ajustes/melhorias feitas	Sem novas iterações de desenvolvimento	Uma nova iteração com alterações no conjunto de dados e/ou parâmetros de treinamento	Diversas novas iterações com alterações no conjunto de dados e/ou parâmetros de treinamento
Avaliação e interpretação do desempenho de um modelo de ML/Testar e aperfeiçoar programas				
C9	Testes com novos objetos	Nenhum novo objeto testado	1-2 novos objetos testados	Mais de dois novos objetos testados
C10	Interpretação dos testes	Interpretação errada		Interpretação correta

Para avaliar o design de interface e os conceitos de algoritmos e programação foi utilizado o CodeMaster que utiliza a rubrica do Alves et al. (2020).

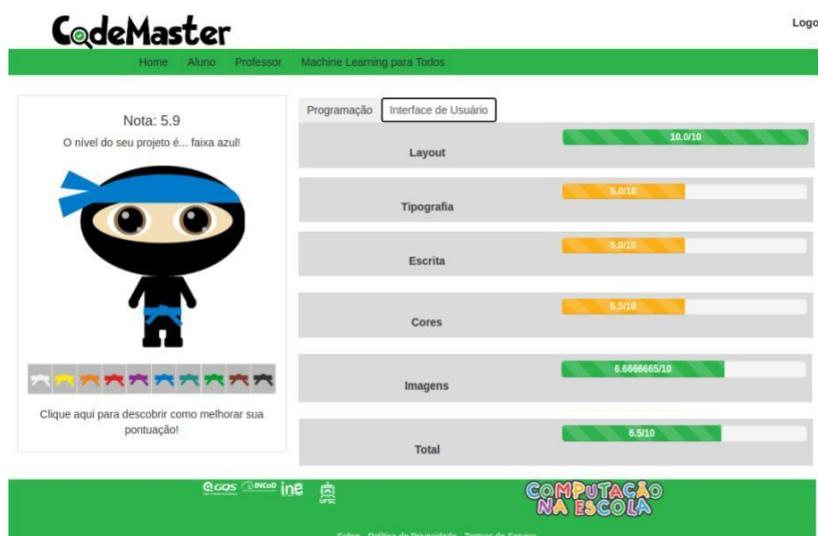


Figura 4. Exemplo da avaliação do CodeMaster.

O curso está atualmente disponível no espaço de professor no moodle UFSC e será em breve publicado de forma gratuita no site de cursos da iniciativa Computação na Escola/INCoD/INE/UFSC.

6. Conclusão

Neste trabalho foi desenvolvido um curso de criação de aplicativo inteligente, utilizando o ensino de ML com ênfase em classificação de imagens para alunos do Ensino Médio em escolas brasileiras. Para o curso foi analisado o contexto, definido os objetivos de aprendizagem, plano de ensino e o material didático do curso.

Com esse curso, espera-se atingir um público amplo no ensino de ML na educação brasileira e incentivar o aluno a desenvolver suas habilidades em áreas de conhecimento relacionadas à ML.

Para trabalhos futuros, sugere-se aplicar esse curso com objetivo de avaliar a sua qualidade e melhorá-lo com base no *feedback* dessas aplicações.

Referências

- AI4K12. Disponível em: <https://ai4k12.org/>. Acesso em: 17 jun. 2022.
- Alves, N. da C., Gresse Von Wangenheim, C., Hauck, J. C. R. and Borgatto, A. F. A (2020) Large-scale Evaluation of a Rubric for the Automatic Assessment of Algorithms and Programming Concepts. In: Proc. of the 51st ACM Technical Symposium on Computer Science Education, Portland/USA, Pages 556–562.
- Anderson, Lorin W., Krathwohl, David R., eds. (2001). Uma taxonomia para aprender, ensinar e avaliar: Uma revisão da taxonomia de objetivos educacionais de Bloom. Nova York: Longman. ISBN 978-0-8013-1903-7.
- Amershi, S. et al. (2019) Software Engineering for Machine Learning: A Case Study. In: Proc. of the International Conference on Software Engineering: Software Engineering In Practice, Montreal, Qc, Canada, 291 - 300.
- Branch, R. M. (2009) Instructional Design: The ADDIE Approach. New York: Springer Science & Business Media.
- Brasil. (1996) Lei nº 9394, de 20 de dezembro de 1996. Estabelece as diretrizes e bases da educação nacional. Brasília, DF, Disponível em: http://www.planalto.gov.br/ccivil_03/leis/L9394compilado.htm. Acesso em: 01 set. 2019.
- Brasil. (2018) Ministério da Educação. Base Nacional Comum Curricular. Disponível em: http://basenacionalcomum.mec.gov.br/images/BNCC_EI_EF_110518_versaofinal_site.pdf. Acesso em: 26 jul. 2019.

- Brasil. (2013) Ministério da Educação. Diretrizes Curriculares Nacionais da Educação Básica. Disponível em: http://portal.mec.gov.br/index.php?option=com_docman&view=download&alias=13448-diretrizes-curriculares-nacionais-2013-pdf&Itemid=30192. Acesso em: 10 set. 2019.
- Brown, T. (2008) Design Thinking. *Harvard Business Review*, 1(1), 1-10.
- Cambraia, A.C. and Scaico, P. D. (2013) Os desafios da Educação em Computação no Brasil: um relato de experiências com Projetos PIBID no Sul e Nordeste do país. *Revista Espaço Acadêmico*, 148(13), 01-09.
- Cardozo, J. (2022) Desenvolvimento de um Curso On-line para o Ensino de Machine Learning voltado à Classificação de Imagens no Ensino Médio. TCC (Graduação) - Curso de Sistemas de Informação, INE, UFSC, Florianópolis.
- CETIC.BR. (2019) Microdados TIC Kids Online Brasil - 2019 - Pais e responsáveis. São Paulo: Cetic. Disponível em: <https://cetic.br/pt/arquivos/kidsonline/2019/pais/>. Acesso em: 05 jul. 2020.
- Computer Science Teachers Association. (2019) CSTA K–12 CS Standards. Disponível em: <https://csteachers.org/page/standards/>. Acesso em: 10 ago. 2019.
- Ferreira, M. et al. (2019) Ensinando Design de Interface de Usuário na Educação Básica: um mapeamento sistemático do estado da arte e prática. In: *Anais do Workshop de Informática na Escola*, Porto Alegre, Brasil, 511-520.
- Franz, A. C. M.. (2021) Desenvolvimento de uma ferramenta visual de classificação de imagens para o ensino de machine learning no ensino médio. TCC (Graduação) - Curso de Sistemas de Informação, INE, UFSC, Florianópolis.
- Goodfellow, I., Bengio, Y.; Courville, A. (2016) *Deep learning*. Cambridge, MA: MIT Press.
- Gresse Von Wangenheim, C., Von Wangenheim, A., Pacheco, F. S., Ferreira, M. N. and Hauck, J. C. R. (2017) 5 Minutos de Programação em Exibições. In: *Anais do 25º Workshop sobre Educação em Computação*, São Paulo, Brazil.
- Gresse Von Wangenheim, C. et al. (a) (2021) A Proposal for Performance-based Assessment of the Learning of Machine Learning Concepts and Practices in K-12. *Informatics In Education*, online, 1-21.
- Gresse Von Wangenheim, C., Hauck, J. C. R., Hauck, J. C. R, F. S., Bertenceli Bueno, M. F. (b) (2021) Visual Tools for Teaching Machine Learning in K-12: A Ten-Year Systematic Mapping, *Education and Information Technologies*, 26(5), 5733-5778.
- Grover, S., Pea, R. (2013) Computational Thinking in K–12. *Educational Researcher*, 42(1), 38-43.
- Imb Skillsbuild For Students. (2020) Module 5: Bringing It All Together. Disponível em: https://ole03.yourlearning.ibm.com/mod/scorm/player.php?a=64¤torg=d21_or_g&scoId=329&display=popup&mode=normal&lang=en. Acesso em: 22 ago. 2021.

- Lee, I. et al.(2011) Computational thinking for youth in practice. ACM Inroads, 2(1).
- Long, D., Magerko, B. (2020) What is AI Literacy?: competencies and design considerations. In: Proc. of the Chi Conference on Human Factors in Computing Systems, ACM, 1-16.
- Manarin, B. (2023) Desenvolvimento de um Modelo de Classificação de Imagens de Larvas de Mosquitos usando Deep Learning. em desenvolvimento. TCC (Graduação) - Curso de Sistemas de Informação, INE, UFSC, Florianópolis.
- Marques, L. S., Gresse Von Wangenheim, C., Hauck, J. C. R.. (2020) Introdução à Machine Learning: desenvolvimento de uma unidade instrucional para ensino de ML na educação básica. Relatório Técnico INCoD/INE/UFSC, Florianópolis, Brasil.
- Mike, K., Hazan, T., Hazzan, O. (2020) Equalizing Data Science Curriculum for Computer Science Pupils. In: Proc. of the 20th Koli Calling International Conference On Computing Education Research, ACM, 2020, 1-5.
- MIT (a). About us. Disponível em: <http://appinventor.mit.edu/explore/about-us.html>. Acesso em: 17 jun. 2022.
- MIT (b). Beginner Video Tutorials. Disponível em: <http://appinventor.mit.edu/explore/ai2/beginner-videos.html>. Acesso em: 17 jun. 2022.
- MIT (c). Events. Disponível em: <https://appinventor.mit.edu/explore/events>. 17 jun. 2022.
- MIT (d). Magic 8-Ball Tutorial. Disponível em: <http://appinventor.mit.edu/explore/teach/magic-8-ball.html>. Acesso em: 17 jun. 2022.
- Nicholson, C. (2019) A beginner's guide to neural networks and deep learning. Disponível em: <https://pathmind.com/wiki/neural-network>. Acesso em: 10 maio 2019.
- Patton, e W., Tissenbaum, M., Harunani, F. MIT App Inventor: objectives, design, and development. Computational Thinking Education, 1(1), 31-49.
- Petersen, K., Vakkalanka, S., Kuzniarz, L. (2015) Guidelines for conducting systematic mapping studies in software engineering: an update. Information And Software Technology, 64(1), 1-18.
- Pinheiro, F. D. C., Gresse Von Wangenheim, C., Missfeldt Filho, R. (2018) Teaching software engineering in K-12 education: a systematic mapping study. Informatics In Education, 17(2), 167-206.
- Project Jupyter. About Us: Project Jupyter's origins and governance. 2022. Disponível em: <https://jupyter.org/about>. Acesso em: 17 jun. 2022.
- QEDU. QEDU Analítico. Disponível em: <https://analitico.qedu.org.br/>. Acesso em 04 de nov. 2021.
- Russell, S. and Norvig, P. (2015) Artificial intelligence: a modern approach. 3. ed. Chennai, TN, India: Pearson Education India.

- SAS. (2019) Redes neurais: o que são e qual sua importância? Disponível em: https://www.sas.com/pt_br/insights/analytics/neural-networks.html#technical. Acesso em: 10 nov. 2019.
- Sociedade Brasileira de Computação. (2017) Referenciais de formação em computação: educação básica. Porto Alegre: SBC. Disponível em: <http://www.sbc.org.br/files/ComputacaoEducacaoBasica-versaofinal-julho2017.pdf>. Acesso em: 10 ago. 2019.
- Solecki, I., Porto, J. A., Alves, N. D. C., Gresse Von Wangenheim, C., Hauck, J. C. R., Borgatto, A. F. (2020) Automated Assessment of the Visual Design of Android Apps Developed with App Inventor. In: Proc. of the 51st ACM Technical Symposium on Computer Science Education, Portland, USA, 51–57.
- Spadini, A. S. (2021) Principais frameworks de deep learning em 2021. Disponível em: <https://www.alura.com.br/artigos/principais-frameworks-de-deep-learning-em-2021>. Acesso em: 15 jul. 2021.
- Sperling, A.; Lickerman, D. (2012) Integrating AI and machine learning in software engineering course for high school students. In: Proc. of the Annual Conference on Innovation and Technology in Computer Science Education, ACM.
- Tissenbaum, M., Sheldon, J., Abelson, H. (2019) From computational thinking to computational action. *Communications Of The ACM*, 62(3), 34-32.
- Touretzky, D. S. et al. (2019) K-12 Guidelines for Artificial Intelligence: what students should know. Disponível em: https://github.com/touretzkyds/ai4k12/raw/master/documents/ISTE_2019_Presentation_website_final.pdf. Acesso em: 06 set. 2019.
- Wikistat. Neural Networks and introduction to Deep Learning. Disponível em: <https://www.math.univ-toulouse.fr/~besse/Wikistat/pdf/st-m-hdstat-rnn-deep-learning.pdf> Acesso: 15 jul. 2021.
- Yu, Y., Chen, Y. (2018) Design and Development of High School Artificial Intelligence Textbook Based on Computational Thinking. *Open Access Library Journal*, 5(9), 1-15.