

CENTRO UNIVERSITÁRIO UNIDADE DE ENSINO SUPERIOR DOM BOSCO
CURSO SISTEMAS DE INFORMAÇÃO

JORGE LUCAS ARAÚJO SALES

**ANALISE DE MODELO DE APLICAÇÃO COM BASE EM TÉCNICAS DE
INTELIGÊNCIA ARTIFICIAL PARA TESTAGEM DE SOFTWARE**

São Luís
2020

JORGE LUCAS ARAÚJO SALES

**ANALISE DE MODELO DE APLICAÇÃO COM BASE EM TÉCNICAS DE
INTELIGÊNCIA ARTIFICIAL PARA TESTAGEM DE SOFTWARE**

Monografia apresentada ao Curso de Sistemas de Informação do Centro Universitário Unidade de Ensino Superior Dom Bosco como requisito parcial para obtenção do grau de Bacharel em Sistemas de Informação.

Orientador: Prof. MSc. Allisson Jorge Silva Almeida

São Luís

2020

Sales, Jorge Lucas

Análise de modelo de aplicação com base em técnicas de inteligência artificial para testagem de software / Jorge Lucas Sales. - São Luís, 2020.

59f.

Orientador: Prof. Me. Allisson Jorge Silva Almeida.

Monografia (Graduação em Sistema de Informação) - Curso de Sistema da Informação – Centro Universitário Unidade de Ensino Superior Dom Bosco – UNDB, 2020.

1. Teste. 2. Software. 3. Inteligência Artificial. 4. Engenharia de Software. 5. Modelo. I. Título.

CDU 004.41

JORGE LUCAS ARAÚJO SALES

**ANALISE DE MODELO DE APLICAÇÃO COM BASE EM TÉCNICAS DE
INTELIGÊNCIA ARTIFICIAL PARA TESTAGEM DE SOFTWARE**

Monografia apresentada ao Curso de Sistemas de Informação do Centro Universitário Unidade de Ensino Superior Dom Bosco como requisito parcial para obtenção do grau de Bacharel em Sistemas de Informação.

Aprovada em: ____/____/____.

BANCA EXAMINADORA:

Prof. MSc. Allisson Jorge Silva Almeida

Mestre em Inteligência Artificial

Centro Universitário Unidade de Ensino Superior Dom Bosco (UNDB)

Prof. Me. Rodrigo Monteiro de Lima

Centro Universitário Unidade de Ensino Superior Dom Bosco (UNDB)

Prof. Me. Allan Kássio Beckman Soares da Cruz

Centro Universitário Unidade de Ensino Superior Dom Bosco (UNDB)

Dedico esse trabalho de conclusão de curso a todos os acadêmicos que se munirão dele.

AGRADECIMENTOS

Agradeço a Deus pela força, paciência e perseverança de passar todo o período de realização deste trabalho de conclusão de curso, obstinado a fazer e termina-lo na data certa e entrega-lo para obtenção do diploma.

Aos meus pais, familiares e amigos que ajudaram com dicas profissionais e de pesquisa para realização deste artigo.

Ao professor orientador mestre professor Allisson Jorge Silva Almeida pelo acompanhamento e direcionamento nos temas escolhidos, antes e durante a pesquisa bibliográfica deste presente trabalho.

"Cada época é salva por um pequeno punhado de homens que têm a coragem de não serem atuais."
(G.K Chesterton)

RESUMO

Este trabalho de conclusão de curso tem como meta identificar a existência de técnicas de inteligência artificial direcionadas a realização de testes de *softwares* diversos, englobando os gerais – de prateleira ou específicos para determinados mercados e empresas. Nesse sentido, foram apresentados três tópicos que esclareceram com propriedade sobre as técnicas existentes, sendo elas aplicadas de forma individual ou globalmente, isto é, uma técnica que seja capaz de realizar os testes vitais e assim averiguar a qualidade de um *software*, ou se terá que viabilizar o desenvolvimento de um modelo que realize as testagens. Diante das pesquisas bibliográficas desenvolvidas, que fora aplicada utilizando-se da metodologia exploratório qualitativa preditiva e explicativa foi possível elencar as técnicas já existentes para testagem. Ao final, pode-se verificar que a aplicação de técnicas de IA é uma boa oportunidade para analisar os códigos de software desenvolvidos em geral, bastando que os interessados, como a comunidade de desenvolvedores ou as empresas envolvidas, possam investir com conhecimento e fomento mais nessa área tão importante que é o teste de *software*.

Palavras-chave: Teste. Software. Inteligência Artificial. Engenharia de Software.

ABSTRACT

This work the conclusion work goals to identify the existence of artificial intelligence techniques directed to conduct various software tests, looking for the general – shelf or specific for certain markets and companies. It will go through three topics that will explain the existence of such techniques, which are used in each test or globally, that is, a technique that can already perform all the vital tests to ascertain the quality of a software, or whether it will have to enable the creation of a model that performs the tests. In view of the bibliographic research working as a predictive and explanatory qualitative exploratory way, the existing techniques for testing were listed, mainly visual and the results and guidelines for the creation of a new one, such as it is impacts on society. Therefore, it is possible to use existing techniques to perform isolated tests and create through them one that performs all kinds of tests.

Keywords: Test. Software. Artificial Intelligence. Software Engineering. Model.

LISTA DE FIGURAS

Figura 01 – 20 Bibliotecas de ciência de dados.	26
Figura 02 – Reconhecimento e mapeamento de alguns níveis do <i>Candy Crush Saga</i>	30
Figura 03 – Gráfico de 4.500 níveis percorridos pelo BAIT.	31
Figura 04 – Print da aplicação soliverdades?	40
Figura 05 – Código da aplicação	40

LISTA DE GRÁFICOS

Gráfico 01 - Linguagens mais usadas.....	25
Gráfico 02 – Economia com Teste de Software sem IA	49

LISTA DE QUADROS

Quadro 01 – Matriz de Confusão.....	36
--	----

LISTA DE TABELAS

Tabela 01 – Tipos de Testes de Software.....	32
---	----

LISTA DE ABREVIATURAS E SIGLAS

AG	Algoritmo Genético
AM	Aprendizado de Máquina
ES	Engenharia de Software
IA	Inteligência Artificial
IC	Inteligência Computacional
NBR	Norma Brasileira
RNA	Rede Neural Artificial
SI	Sistemas de Informação
UNDB	Unidade de Ensino Superior Dom Bosco
TCC	Trabalho de Conclusão de Curso
TI	Tecnologia da Informação
WOW	World of Warcraft

SUMÁRIO

1 INTRODUÇÃO.....	15
1.1 PROBLEMATIZAÇÃO.....	16
1.2 OBJETIVO GERAL	17
1.2.1 OBJETIVO ESPECÍFICO.....	17
1.3 JUSTIFICATIVA	17
1.4 SÍNTESES DA METODOLOGIA.....	18
2 INTELIGÊNCIA ARTIFICIAL	20
2.1 ANÁLISE DO CENÁRIO ATUAL.....	24
2.1.1 Teste de Software	32
2.2 TÉCNICAS DE INTELIGÊNCIA ARTIFICIAL.....	36
2.2.1 Matriz de Confusão	36
2.2.2 Rede Neural	42
2.2.3 Algoritmos Genéticos	44
2.2.4 Aprendizado de Máquina	45
2.2.4.1 Aprendizado Supervisionado.....	46
2.2.4.2 Aprendizado Não Supervisionado.....	46
2.2.5 Deep Learning.....	47
2.3 IMPACTOS DA APLICAÇÃO DE INTELIGÊNCIA ARTIFICIAL.....	48
3 RESULTADOS E DISCUSSÕES.....	52
4 CONSIDERAÇÕES FINAIS.....	54
REFERÊNCIAS	55

1 INTRODUÇÃO

Todo teste de *software* é o responsável pela validação da integridade de qualquer aplicação desenvolvida, seja ele de prateleira – genérico que é vendido em massa como diz (SOMMERVILLE, 2011, p.4) [...]em *softwares* genéricos, a organização que o desenvolve controla sua especificação. Para produtos sob encomenda, a especificação é normalmente desenvolvida e controlada pela empresa que está adquirindo o *software*.” – ou específico, que se adequa as necessidades diretas da empresa.

Nesse contexto, visa-se verificar se técnicas de Inteligência Artificial podem aperfeiçoar o processo dessa validação, passando pelos vários tipos metodológicos de testes que os *softwares* que são submetidos para verificação dos requisitos levantados em sua pré-elaboração.

[...]A Engenharia de Software evoluiu significativamente nas últimas décadas procurando estabelecer técnicas, critérios, métodos e ferramentas para a produção de software, em consequência da crescente utilização de sistemas baseados em computação em praticamente todas as áreas da atividade humana, o que provoca uma crescente demanda por qualidade e produtividade, tanto do ponto de vista do processo de produção como do ponto de vista dos produtos gerados. A Engenharia de Software pode ser definida como uma disciplina que aplica os princípios de engenharia com o objetivo de produzir software de alta qualidade a baixo custo[...]. (BARBOSA, Ellen, p.2)

Os problemas mais comuns encontrados no desenvolvimento de software, tem a ver como o versionamento, limite do tráfego de informações que determina se o sistema pode suportar sem apresentar erro ou lentidão, perda da comunicação e recarregamento da página, integração entre os módulos que deveriam estar se comunicando constantemente, integridade de cada módulo e funções do especificadas do projeto são característicos de *softwares* em desenvolvimento. Cada um dos pontos abordados é um possível ser identificado durante a validação por meio da testagem de *software*. Diante do exposto, buscou-se identificar como as técnicas de inteligência artificial podem contribuir com possíveis soluções para esses problemas, de forma evolutiva, com base nas características atualmente utilizadas no mercado de trabalho no desenvolvimento de *softwares*.

Como a migração das tecnologias para o desenvolvimento de técnicas de Inteligência Artificial. Isto é observado por exemplo pelo constante desenvolvimento de *chatbots*, aplicados no atendimento bancário 24 horas aos usuários de determinados bancos. Esse processo vem se tornando cada vez mais exigente, tendo como base a própria criação de um sistema que possa, por si só, aprimorar-se, o que exige um volume imenso de informações. Isso desencadeou a busca de uma forma inteligente e recursiva, alinhando-se aos fatores propiciados pela IA, podendo ajudar na efetividade da aplicação desses testes.

1.1 PROBLEMATIZAÇÃO

Problemas de versionamento, limite do tráfego de informações que determinado sistema pode suportar sem apresentar erro ou lentidão, perda da comunicação e recarregamento da página, integração entre os módulos que deveriam estar se comunicando constantemente, integridade de cada módulo e funções especificadas no projeto. Cada um dos pontos abordados é um possível problema a ser identificado durante o teste de *software*.

[...]Existem vários tipos de sistemas de software, desde os simples sistemas embutidos até os sistemas de informações complexos, de alcance mundial. Não faz sentido procurar notações, métodos ou técnicas universais para a engenharia de software, porque diferentes tipos de software exigem abordagens diferentes. Desenvolver um sistema de informações corporativo é totalmente diferente de desenvolver um controlador para um instrumento científico. Nenhum desses sistemas tem muito em comum com um jogo computacional com gráficos intensos. Todas essas aplicações precisam de engenharia de software, embora não necessitem das mesmas técnicas[...]. (SOMMERVILLE, 2011, p.2)

Tomando conhecimento dessa dificuldade e com a crescente migração das tecnologias para o desenvolvimento de técnicas de IA, esse processo vem se tornando cada vez mais exigente, tendo como base a própria criação de um sistema que possa, por si só, se aprimorar, o que exige uma quantidade imensa de informações. Isso desencadeou buscas na viabilidade da própria técnica de Inteligência artificial poder ajudar na efetuação desses testes de *softwares* em desenvolvimento.

Partindo desse pressuposto, questiona-se: como aplicar nesses vários tipos de testes, seja uma ou mais técnicas de Inteligência Artificial na realização dos testes de software?

1.2 OBJETIVO GERAL

Analisar a aplicação de técnicas de Inteligência Artificial aos variados tipos de testes de teste de *software* existentes.

1.2.1 OBJETIVOS ESPECÍFICOS

- A. Identificar a existência de técnicas de inteligência artificial que possa realizar teste em cada etapa do desenvolvimento da aplicação;
- B. Verificar a criação de uma técnica de inteligência artificial que empregue os conhecimentos de outras técnicas de inteligência;
- C. Impactos que a implementação de técnicas de inteligência artificial pode ocasionar no processo e desenvolvimento de teste de *software*.

1.3 JUSTIFICATIVA

Ao decorrer da história, se for traçado um comparativo entre a revoluções, notou-se que a que exerceu a maior velocidade de evolução foi a tecnológica. Com isso, e suas várias etapas, presenciemos atualmente a migração de um sistema onde toda aprendizagem era feita manualmente para o emprego de Inteligências Artificiais - ou suas técnicas - que tem como característica o auto incremento do seu código por meio de atualizações que o próprio sistema faz em si mesmo.

[...]A Primeira Revolução Industrial (1780) foi impulsionada pela concepção dos teares mecânicos dirigidos por motores a vapor, e culminou na centralização do processo de produção em fábricas; a Segunda Revolução ocorreu cerca de 100 anos depois, e tem como marca a inserção das linhas de produção e a construção do Ford T; a Terceira Revolução Industrial ocorreu no final da década de 1960, e é marcada pela apresentação do primeiro controlador lógico programável, que permite a programação de sistemas digitais[...]. (PEREIRA, 2018, p. 2)

Em vista dessa realidade e com a afinidade as implicações profissionais existentes de um estudo nessa área viram-se a necessidade de empregar esforços neste ramo de pesquisa. Em concordância com essa necessidade se tem o impacto no meio acadêmico e social, que com o emprego de técnicas de sistemas inteligentes na testagem de aplicações, se pode construir *softwares* mais robustos e com menor

possibilidade de erro ou roubo de informações privadas, isto é, com maior potencial e maior segurança na informação contido e gerada pelas técnicas.

Por fim, o resultado dessa linha de estudo tem seu desdobramento em uma nova linha de pesquisa que resultará desse desenvolvimento. As pesquisas e artigos voltados a essa área, podem proporcionar estudos na área de *big data*, redes neurais, aprendizado de máquina, mineração de dados, aprendizado profundo. Que já vem sendo realizada, entretanto em áreas limitadas do conhecimento.

1.4 SÍNTESES DA METODOLOGIA

O vigente trabalho de conclusão de curso busca elucidar a viabilidade na aplicação de técnicas de Inteligência artificial na realização das mais variadas bibliotecas de testes, necessários para verificar a qualidade de um *software*, de qualquer tipo, que for requisitado pelo cliente. A pesquisa foi feita por meio de uma pesquisa bibliográfica retirando conceitos, problemas, validações, análises, gráficos, tabelas, quadros e indicadores de desempenho. Como diz o autor:

[...]abrange toda bibliografia já tornada pública em relação ao tema de estudo, desde publicações avulsas, boletins, jornais, revistas, livros, pesquisas, monografias, teses, material cartográfico etc., até meios de comunicação orais: rádio, gravações em fita magnética e audiovisuais: filmes e televisão. Sua finalidade é colocar o pesquisador em contato direto com tudo o que foi escrito, dito ou filmado sobre determinado assunto, inclusive conferências seguidas de debates que tenham sido transcritos por alguma forma, quer publicadas, quer gravadas[...]. (LAKATOS, 2003, p.183)

Para melhor averiguação do estudo, foi feito também pesquisas em ambientes de desenvolvimento de aplicações multiplataformas, onde há profissionais voltados no desenvolvimento de aplicações a serem utilizadas pelos clientes e na criação de algoritmos inteligentes. Coletando assim, ocasionalmente, dados do progresso e resultados, em ambientes de desenvolvimento de *softwares* inteligentes e técnicas de IA, incubadores de startups físicos ou virtuais que tenham como foco a mesma modalidade. Essas informações coletadas são analisadas buscando comparativos com a literatura existente e validando a pesquisa feita nos ambientes de coletas de dados ou, em contrapartida, elencando os defeitos na realização dos testes. Por isso será feita de forma exploratória qualitativa preditiva e explicativa com cita (LAKATOS, 2003, p.239) [...]Quando reúne e relaciona material obtido de

diferentes fontes, expondo o assunto com fidedignidade e demonstrando habilidade não só de levantamento, mas também de organização”.

Tendo já definido o objetivo e o local a ser usado como fonte de informação experimental se providenciou-se toda a documentação necessária para que a pesquisa possa ser realizada. Permissões para entrevistas, filmagens, reprodução em artigo de informações dadas e fichamento das referências e material autoral usada durante o período de pesquisa do vigente trabalho.

2 INTELIGÊNCIA ARTIFICIAL

A inteligência artificial, em seu conceito mais puritano – atualmente - não pode ser construída em decorrência da limitação do conhecimento humano, e da lógica a ser construída para realizar tal feito. Como afirma Pessis (2001), o professor Heinz von Forester, um dos fundadores da cibernética, não acredita que o pensamento humano possa ser programado – da mesma forma que não se consegue mapear e definir o quanto de capacidade cerebral se pode usar, em vista que ninguém sabe como medir a totalidade da mente humana – logo não há como se programar algo em que se não tem certeza de como funciona em sua plenitude.

Uma das formas de elaborar um algoritmo conciso, que atenda às necessidades para qual ele foi proposto, é ter uma ideia real do que ele deve fazer e que objeto real ele vai automatizar, como por exemplo uma emissão de nota fiscal, que será automatizada e todo o processo será digital. Uma vez que não se tem toda a capacidade neuronal elucidada a criação literal de uma inteligência artificial fica prejudicada. Na comunidade científica até esse âmbito mais básico ainda está em discussão.

Em outro aspecto de uma simulação de inteligência humana se tem a memória que como explica a (ADES, 1993, p.9) “[...]A memória é, basicamente, está intrusão do passado no presente, seja sob a forma de imagens, seja como instruções, implícitas o explícitas, de como agir” isto diz que, para o cérebro, não existe noção de tempo e espaço ela só é percebida como algo passado quando comparada com o presente, ou, muitas das vezes, a própria noção de que ele estava em outro lugar e não no que está no presente, em suma, a memória só é compreendida com tal quando se confronta duas realidades e se define, através de características sensoriais qual é a do passado e qual a presente.

Muito similar ao sonho, em que você acredita que é a realidade e só se tornar consciente do ato de estar sonhando, quando desperta. Na computação isso se dá mais facilmente pela datação de cada informação obtida e também que, para um computador tudo é presente, em sua forma mais literal.

Outro processo similar ao da memória também descrito por (ADES, 1993, p. 9) “[...]As lembranças podem surgir de forma mais ou menos vívida, espécie de cópia de eventos vividos anteriormente, cópia quase sensorial; podem aparecer como relatos, histórias que contamos para nós mesmo ou para os outros.” pode ser

construído de forma a imitar um reforço de memória, podendo até classifica-los em negativo ou positivo para proporcionar uma maneira de diferenciar a entonação quando implementarem um simulador de voz; situações associados a catástrofes naturais ou eventos extremamente violentos em épocas de guerra, pode ser reproduzido em uma tonalidade de campo harmônico menor e fatos positivos com grande descoberta científicas - como curas – em uma entonação mais animada.

Esse reforço no ser humano, geralmente, se dá quando é constantemente consultada determinada memória ou quando essa está associada a uma carga emocional levada. Nas IA's podem ser caracterizadas pela divisão em conjuntos hierarquizados em ordem de importância ou apenas pela numeração por intensidade de cada evento.

Outra denominação dada a IA é de inteligência computacional, linguagem que toma como força motriz a tentativa de construção de sistemas baseados em conhecimento, que são as IA's. Nesses sistemas temos as 3 classes mais populares que são: redes neurais, aprendizado de máquina e algoritmos genéticos. Para (SÁ, 2010) as técnicas de IA são algoritmos ou sistemas que procuram simular e reproduzir comportamentos intelectuais humanos, buscando assim obter a capacidade de gerar um raciocínio através da percepção junto com o aprendizado e adaptação, ou seja, evolução, resultando assim no desenvolvimento do conhecimento que é o caminho natural e estágio final de todo dado.

[...]ligação entre o conhecimento com os diferentes meios e tecnologias para seu registro e processamento, faz-se necessário uma distinção entre dado, informação e conhecimento no qual o dado, é um elemento puro e quantificável sobre um determinado evento; a informação representa um dado analisado e contextualizado; a interpretação de um conjunto de dados e o conhecimento que se refere à habilidade de criar um modelo mental que descreva o objeto e indique as ações a serem implementadas e as decisões a serem tomadas[...]. (SÁ, 2010, p. 44)

Para (Arariboia, 1988, n/p) “[...]a Inteligência Artificial é um campo que usa técnicas de programação que procuram, por intermédio de máquinas, resolver problemas do mesmo modo que um ser humano os resolveria.” Essa resolução se dará por meio de várias simulações, que colheram dados, ou receberem dados que manualmente a ser enviados por um especialista, e através de regras lógicas os trataram - dependendo do objetivo - para o qual a técnica foi construída. Vamos trabalhar neste artigo científico com essa definição, daqui por diante, de forma a facilitar o entendimento do conteúdo a ser abordado e explanado.

Existem 5 correntes filosóficas para discutir os ramos da Inteligência artificial Segundo (CONAI, 1994, p.35):

- A. Estruturalista: há mecanismos que realizam as tarefas, basta descobri-los.
- B. Conexionista: há processos simples que se auto-organizam, basta descobri-los.
- C. Genética: há a inteligência e o caos, basta separá-los por seleção natural, como faz a natureza, porém em escala de tempo adequada.
- D. Fenomenológica: tudo é caótico, em avanço paralelo. Ao se dar uma sintonia momentânea, dá-se a comunicação. Não se conhece chave para repetir a inteligência.
- E. Metafísica: só é possível compreender o intelecto. A inteligência pertence a outra dimensão, inacessível por meios intelectuais.

Como podemos observar o campo da inteligência artificial ainda é vago, o mesmo acontece na questão aplicável aos testes de *softwares*. Embora se tenha várias abordagens a se considerar na hora decisória somente a mais adequada a necessidade, seja ela de bibliotecas, tratamento de dados estatísticos ou a busca na rede mundial de computadores por aprendizagem profunda. Por mais que tenhamos uma ideia do tipo de problema a ser testado nas 8 modalidades mais comuns, ao se testar uma aplicação, o seu uso total e em conjunto ainda terá que ser descoberto. Podemos mapear todo o campo de teste e através de construções de modelos isolados e tentar elaborar um geral, como toda resolução de um problema grande, dividir em micro tarefas e realizar uma etapa de cada vez.

Dentre as abordagens a mais pessimista é a metafísica, onde o assunto é lidado de forma a impossibilitar qualquer esforço otimista na construção de um IA pura, que é a que consegue simular todos os aspectos da inteligência humana. Nessa questão de uma IA pura, podemos considerar como certa a projeção metafísica se nos questionar se ela só poderia simular, de forma perfeita, se tivesse emoção, uma vez que o ser humano toma decisão considerando o contexto emocional, não no âmbito de a emoção eliminar o racional – como recorrentemente ocorre em pessoas menos esclarecidas – mas a complementando. Essa hipótese seria plausível neste caso.

A IA teria que contrair emoção para replicar a inteligência descrita no modelo metafísico ou a decisão com ausência desse aspecto emocional ainda seria uma decisão inteligente e, de certa forma, inteligentemente humana. Nas outras dimensões filosóficas de abordagem podemos influir a conclusão da Inteligência computacional ultrapassar o limite humano, ou seja, atingir um nível de inteligência humanamente impossível, a superinteligência.

[...]O termo "singularidade" se refere ao momento em que as máquinas ultrapassam os humanos na inteligência. Desde que os seres humanos conseguiram criar algo mais inteligente do que eles próprios, esse novo tipo de cérebro pode produzir algo mais esperto do que ele mesmo e assim sucessivamente, em grande velocidade. Haverá limites para quanto tempo isso pode continuar. Mas como os poderes computacionais aumentaram rapidamente ao longo das décadas, os limites para o que uma inteligência superior pode fazer estão além do que podemos imaginar agora. A singularidade e a superinteligência exercitam bastante alguns participantes no debate da IA, [...]. (RISSE, 2018, p. 22)

Nesse contexto entra vários problemas e dilemas, tal como o do bonde, que diz que se uma IA estiver na direção de um automóvel e estiver nas pistas por onde está seguindo uma pessoa de um lado e 3 da outra, qual ela iria escolher atropelar. Essa discussão pode ser projetada para casos de testes de se a IA deveria variar – em relação a moral imbuída nela – de acordo com a região. Cito a Índia, que tem como animal sagrada a vaca, ela devia atropelar as 3 pessoas ou vaca. No teste visa-se contemplar o maior número de hipóteses possíveis, já verificando esses aspectos e criando uma árvore de decisão baseada.

Dentre os problemas mais preocupantes está a questão do limite, como descrito por (RISSE, 2018) que o temos que garantir – antes de chegar perto de qualquer singularidade na criação de uma entidade composta de inteligência artificial – que os valores dela seja compatível com os nossos, evitando assim todo e qualquer cenário, até mesmo os apocalípticos descritos no ambiente cinematográfico norte americano. Para tal, foi criado as leis da robótica que consiste em 3 regras básicas a serem impressas em uma IA para preservação da vida humana, independentemente do nível de consciência que a IA atingiu em determinada época.

Essas leis eram baseadas em não matar ou não agir, quando um humano estiver em perigo e por fim preservar a integridade da IA. Já vemos que este item por mais direto que seja não evita o problema do bonde, muito menos o do paradoxo que se evidencia quando se nota que se uma IA no decorrer de sua construção e aprimoramento terá a capacidade de melhorar a si mesma, ou seja, de reescrever seu

próprio código, logo terá a possibilidade de mudar a leis fundamentais da robótica e assim desenvolver-se sem limite.

Portando de antemão se deveria procurar falhas nesse aspecto essencial, verificar se a reescrita de código pode atingir até características “bloqueadas” ou até colocar um limitador de domínio, como se faz em redes de compartilhamento de arquivos, a cada usuário definido níveis de permissões estruturais, parametrizando assim limites ao qual cada acesso pode chegar.

2.1 ANÁLISE DO CENÁRIO ATUAL

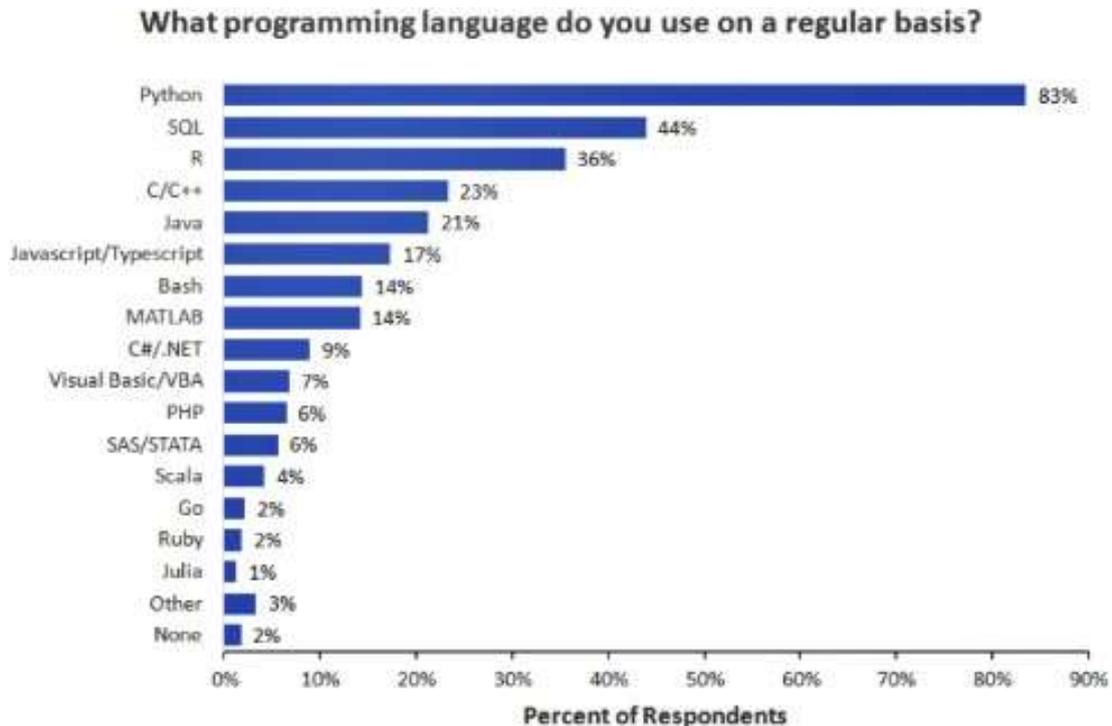
Atualmente não existe nenhuma inteligência artificial propriamente dita, como já foi explanado, mas o caminho até lá continua a ser trilhado de modo obstinado. Dentre a preparação para tal, o engenheiro de *software*, analista ou o profissional encarregado do projeto tende a escolher uma linguagem de criação da técnica de IA que atenda as especificações levantadas na coleta dos requisitos. As mais usadas eram as que os programadores estavam mais habituados e que se adequavam ao maior número de sistemas.

Antes do desenvolvimento de linguagens estritamente dedicadas a determinada tarefa, como a linguagem de programação R, o jeito mais prático era usar de bibliotecas e linguagens multiplataformas deixando aquelas que exigiam um maior detalhamento ou que exigia uma alta curva de aprendizagem por ser muito próxima da linguagem de máquina.

No gráfico a seguir veremos o uso das linguagens no desenvolvimento de uma técnica de Inteligência Artificial - de modo geral – isto é, independente da especificidade do projeto, tendo em vista que alguns necessitam exclusivamente serem elaborados por essa ou aquela linguagem por proporcionar o maior acervo de opções de bibliotecas dedicadas.

Essas linguagens específicas para projetos podendo até ultrapassar a classificação atual à medida que os desenvolvedores vão migrando e construindo suas bibliotecas de acordo com a exigência do mercado, podemos citar o próprio uso de estatísticas com forma de prever eventos ou analisar suas causas, através de modelos de regressão, com a finalidade de limita a recorrência dos eventos, tal como a pandemia do vírus em 2020.

Gráfico 01: Linguagens mais usadas

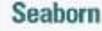
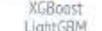
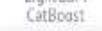
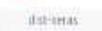
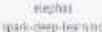


Fonte: Blog.geekhunter (2020)

Mas o que leva a esse grande aumento da migração dos programadores a linguagem é sua simplificação, e o fato de ser multiplataforma. Como diz (ANTUNES, 2020, p. 54182) “[...]focada em produtividade e legibilidade, ou seja, para produzir um código fácil para o programador entender: baixo uso de caracteres especiais; baixo índice de palavras-chave; gerenciador de memória eficiente que evita vazamento de memória”. No *Python*, por ser focado na simplificação do código e encomia de processamento as regras criadas na IA e a lógica já difícil de ser formulada fica mais objetiva ao ser lida, uma vez que a maior preocupação na linguagem é a indentação, ou seja, a organização da estrutura.

Para a começar a escolha de uma técnica em detrimento de outra, primeiramente, tem-se que elencar as melhores linguagens de aprendizado inteligente que podem melhor desenvolver uma técnica de IA. Em fóruns ou repositórios pode-se achar várias especificações onde se pode basear na escolha de uma linguagem para programação ou desenvolvimento de uma nova técnica de inteligência computacional, sempre levando em consideração o objetivo da técnica, sua finalidade, levantada no momento da coleta de requisitos.

Figura 01 – 20 Bibliotecas de ciência de dados.

Github data Python 2018										
Library Name	Type	Commits	Contributors	Releases	Watch	Star	Fork	Commits/ Contributors	Commits/ Releases	Star/ Contributors
 matplotlib	Visualization	25 747	725	70	498	7 292	398	36	368	10
 Bokeh	Visualization	16 983	294	58	363	7 615	2 000	58	293	26
 plotly	Visualization	2 906	48	8	198	3 444	850	61	363	72
 Seaborn	Visualization	2 044	83	13	205	4 856	752	25	157	59
 pydot	Visualization	169	12	12	17	193	80	14	14	16
 learn	Machine learning	22 753	1 084	86	2 114	28 098	14 005	21	265	26
 XGBoost	Machine learning	3277	280	9	868	11 991	5 425	12	364	43
 LightGBM	Machine learning	1083	79	14	363	5 488	1 467	14	77	69
 CatBoost	Machine learning	1509	61	20	157	2 780	369	25	75	46
 eli5	Machine learning	922	6	22	39	672	89	154	42	112
 SciPy	Data wrangling	19 150	608	99	301	4 447	2 318	31	193	7
 NumPy	Data wrangling	17 911	641	136	390	7 215	2 766	28	132	11
 pandas	Data wrangling	17 144	1 165	93	858	14 294	5 788	15	184	12
 Statsmodels	Statistics	10 067	153	21	234	2 868	1 240	66	479	19
 TensorFlow	Deep learning	33 339	1 469	58	7 968	99 664	62 952	23	575	68
 PyTorch	Deep learning	11 306	635	16	816	15 512	3 483	18	707	24
 Keras	Deep learning	4 539	671	41	1 673	29 444	10 954	7	1111	44
 dist-keras	Distributed deep learning	1125	5	7	41	431	106	225	161	86
 elephas	Distributed deep learning	170	13	5	97	913	189	13	34	70
 spark-deep-learning	Distributed deep learning	67	11	3	116	920	206	6	22	84
 Natural Language Toolkit	NLP	13 041	236	24	467	6 405	1 804	55	543	27
 spaCy	NLP	8 623	215	56	425	9 258	1 446	40	154	43
 gensim	NLP	3 603	273	52	415	6 995	2 689	13	69	26
 Scrapy	Data scraping	6 625	281	81	1 723	27 277	6 469	24	82	97

Last reviewed: 11.02.2018

Created by ActiveWizards

Fonte:Datascienceacademy (2018)

Na figura acima temos somente as bibliotecas destinadas a área de ciência de dados ou mineração de dados - elencadas em ordem de importância no ano de 2018 - um ramo das técnicas de IA que lida com a descoberta de padrões em um extenso volume de dados. Ele também é muito recomendado para, principalmente, trabalhar com aprendizado de máquina.

[...]O Python também é multiplataforma, e caso apareça alguma que ainda ele não suporte, os desenvolvedores podem modificar o código para que seja possível compilar para aquela linguagem, pois o Python também é uma linguagem livre. Assim, todos esses fatores fazem com que o Python seja uma das preferidas linguagens para o trabalho com inteligência artificial, data mining e machine learning[...]. (ANTUNES, 2020, p. 54183)

Em contrapartida, tem como desvantagem que, por ser uma linguagem interpretada, necessita de um tradutor para rodar e também há pouca documentação existente, já que como as linguagens relacionadas a inteligência, estão em desenvolvimento e consolidação o processo documental está em andamento. Todo o material, de certa forma, pseudo-documental está situado em repositórios como o *GIT* ou *GITHUB*. Em virtude disso o fortalecimento das comunidades online de programação tem criado um novo parâmetro de aprendizagem onde usuários se conectam e resolvem seus problemas, algumas vezes sem nenhuma intermediação profissional – no *brutal force* – usando uma expressão da disciplina de redes de computadores. Um exemplo disso é a lista de 2020 onde já se nota mudanças na importância das listas das bibliotecas mais usadas nos ambientes de programação, como mostra (COUTINHO, 2020, n/p)

- A. Arrow;
- B. Numpy;
- C. Pandas;
- D. Bokeh;
- E. NLTK;
- F. Pytil;
- G. Poetry;
- H. Theano;
- I. Scikit Learn;
- J. Tensor Flow;
- K. Keras.

Está ramificação de estudo das técnicas de inteligência artificial, como os demais, se obtém pouco referencial teórico por sua difícil construção de modelos e até de aplicações em si que atenda a todos aspectos da testagem. Todo o estudo nessa ramificação consiste em desenvolver uma técnica de IA focada na identificação de padrões e sanar a questões apresentadas em pesquisas no ramo médico.

[...]No entanto, apesar dos resultados da aplicação de algoritmos deep learning, existem problemas que ainda não resolvemos. Uma aplicação recente de deep learning para detecção de câncer de pele descobriu que o algoritmo era mais preciso que um dermatologista certificado. Porém, enquanto dermatologistas podem enumerar os fatores que os levaram a um diagnóstico, não há como identificar quais fatores um programa deep learning utilizou em sua classificação. Isso é chamado de problema de caixa preta de deep learning[...]. (JONES, 2017, n/p)

Como no exemplo acima citado pelo autor os investimentos na área de inteligência computacional são dedicados mais as áreas médicas em virtude da necessidade de identificação prévia e do longo tempo de preparo desse profissional. Mas todo o investimento feito retorna resultados melhores que o previsto como descritos em que o algoritmo conseguiu uma identificação do câncer de pele melhor que os profissionais dermatológicos. Embora o fato de as correlações feitas não serem usuais o algoritmo pode a partir daí - através de sua análise - extrair melhores caminhos de identificação de doenças, isso deixa os testes mais rápidos e mais eficazes.

Todo o problema está na identificação dos padrões encontrados e usados pelo algoritmo, uma vez que eles podem ser escritos em uma lógica não compreensível a mente humana. Superada essa barreira de tradução a técnica já usada no mercado pode ser aplicada e melhorada por meio dos testes periódicos ou até se for impossível com a tecnologia atual, testa-la para identificar somente tendo como base os padrões pré-estabelecidos.

[...]Outro aplicativo, chamado Deep Patient, podia prever uma doença com sucesso, dados os registros médicos do paciente. O aplicativo se provou muito melhor no prognóstico de doenças do que os médicos — mesmo para esquizofrenia, que é notoriamente difícil de diagnosticar. Portanto, mesmo que os modelos funcionem bem, nenhum pode alcançar grandes redes neurais para identificar o porquê[...]. (JONES, 2017, n/p)

Este outro exemplo aponta, não somente que o aprendizado é eficaz em doenças que seu prognóstico envolve apenas o tratamento de imagem como sua forma motriz de identificação, como também casos mais psicológicos. Analisando o histórico médico do paciente a técnica de IA é capaz de inferir correlações tão eficazes quanto as outras.

Ao se tratar dos sistemas de IA existem os 2 tipos que são os baseados em conhecimento e baseados em estatísticas que como explica (BARANAUSKAS, 2005) tem como objetivo a busca e implementação de modelos estatísticos para encontrar, de forma aproximada um, uma aproximação do conceito induzido. Isto é feito por meio da transformação, onde um classificador modifica uma classe, e a expressa através de uma combinação linear de valores dos atributos que então busca outra combinação linear que possa ser próxima a sua amostra base. Um desses métodos é o aprendizado Bayesiano que se furta do meio probabilístico baseados em um conhecimento predeterminado.

Já os baseados em conhecimento são criados de para imitar, através de regras, o conhecimento de um perito para resolver problemas que somente uma pessoa especializada em determinado ramo de atividade conseguiria, como na área médica supracitada. Um sistema que pudesse reconhecer através de imagens de ressonância se determinada pessoa tem ou não câncer ou outro tipo de enfermidade, poderia agilizar os processos e até extinguir as filas de espera em unidade de pronto atendimento públicas (MENDES, 1997).

Esses dois padrões de desenvolvimento podem ser aplicados na criação de uma técnica de IA, através de algum sistema especialista, boa parte dos códigos e bibliotecas usadas – ou regras de raciocínio – como as contidas na criação de uma rede neural artificial poderá ser reusada até para simular um usuário acessando determinado *software*. Como também no teste de usabilidade simular um usuário portador de deficiência teria o mesmo padrão de resposta

Em virtude disso pode-se afirmar que sua existência é bastante escassa. Mesmo assim a empresa King criou uma Inteligência para testar o mapa de seu jogo que é extenso e levaria ao testador humano uma fadiga física e psicológica devido a repetição.

[...]Para tornar esse esforço atraente para os testadores, nos concentramos no desenvolvimento de um bot de controle de qualidade chamado BAIT - Bot for AI Testing (robô para teste com IA), desenvolvido pela AI, que daria suporte aos testadores com tarefas tediosas e entediadas de teste, e também para cobrir áreas não testadas devido à falta de recursos, e testar áreas que antes eram consideradas impraticáveis ou impossíveis de testar manualmente[...]. (ANDELKOVIC, 2020, n/p)

Segundo (ANDELKOVIC, 2020, n/p) a IA identificaria o idioma ao qual o jogo está por meio de sua localização geográfica e adaptaria seu padrão de reconhecimento para aquele idioma. Posteriormente ela começaria a mapear cada

tela do jogo para testar sua funcionalidade, reconhecendo assim cada texto, ícone e imagem depois armazenaria a tela presente e clicaria no próximo botão para passar para outra tela e repetir o processo até mapear todos os níveis existente, no momento, do jogo.

Em tarefas mecânicas vemos que tem como se criar uma técnica de IA a ser usada em meio ao desenvolvimento, mas a questão é até onde ela iria sem prejudicar o próprio processo. De certa forma algum especialista ou encarregado ainda terá que verificar, tela por tela, nos primeiros momentos em que a inteligência começou a operar, para garantir que todo o processo foi mapeado corretamente e algum defeito na renderização das telas não interferiu no reconhecimento feito, existem uma infinidade de causas usuais, até a cor dos botões e caixas de textos, identificados, muito próxima as das bordas e fundo de telas podem gerar erros eventuais. Esses erros podem ser corrigidos pelo profissional responsável por meio de uma atualização da IA ou pela mudança no *design* do jogo de modo geral ou específico para aquela tela que ocasionou o erro.

Figura 02 – Reconhecimento e mapeamento de alguns níveis do Candy Crush Saga

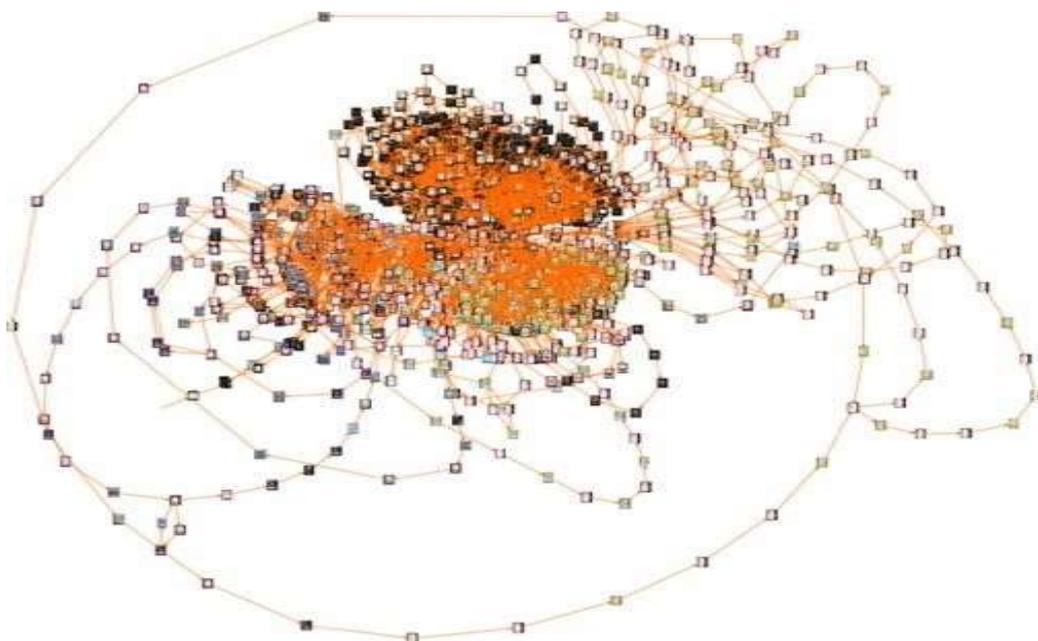


Fonte: Infoq (2020)

Na figura acima vemos o que a técnica de Inteligência artificial vasculhou e testou, reportando todo o funcionamento à medida que percorria os níveis do jogo *Candy Crush Saga* de forma a identificar bugs, que como explica (SÁ, 2010, p.24) “[...]um erro representa a manifestação oriunda da ação humana que causa problemas internos no *software* e são chamados de (*bugs*). Um (*bug*) pode levar a um defeito no produto” Todo o mecanismo criado até o momento, busca essa identificação de erros que pode ocasionar problemas até estruturais no *hardware* – como por exemplo – um alto poder de processamento gera uma maior quantidade, escalável, de calor, isto é, quando mais poder de processamento maior a temperatura, diminuindo a vida útil do *hardware* e equipamentos sensíveis próximos.

Assim sendo qualquer, fase do jogo que – por alguma funcionalidade errada – gere um aumento da demanda de processamento ou erros que podem ter o mesmo resultado, danificará o equipamento. Todo o caminho percorrido pelo BAIT servirá para fazer esse mapeamento completamente automático. Na figura a seguir será mostrada como ficou o mapeamento ao final e o número de níveis percorridos pela técnica de IA desenvolvida pela empresa.

Figura 03 – Gráfico de 4.500 níveis percorridos pelo BAIT



Fonte: Candy Crush Saga - Utilizando a Inteligencia Artificial nos testes (infoq.com)

Nesta imagem, o BAIT verificou todos os quatro mil e quinhentos níveis do jogo. Este mesmo teste poderia ser feito manualmente, mas a diferença de tempo na execução seria significativamente maior. Outro exemplo significativo seria o mapeamento do jogo *World Of Warcraft*, onde na criação de uma nova extensão para locomoção dos personagens ela fosse testada através de uma técnica de IA que percorria o mapa, por várias rotas diferentes, verificando se há algum *bug*, *lag*, que é o momento onde se trava ou ocorre uma lentidão, que em diversas causas para isso, ou qualquer outro tipo de problema como perda de coloração ou definição. Entretanto em WoW o teria um BAIT mais especializado, em que sua função deveria levantar dados de todos os mapas do jogo procurando possíveis erros ocasionados pela adição de uma nova extensão do jogo – o que popularmente acontece em *jogos player versus player* onde uma nova habilidade desbalanceando o jogo gerando interrupções na transmissão de imagens – esse mapeamento ocorreria nos objetos terrestres e suspensos tal como NPC que são os objetos interativos não jogáveis pelos usuários do jogo.

2.1.1 TESTE DE SOFTWARE

Tendo como objetivo a validação da qualidade, que é saber se o *software* está atendendo aos requisitos levantados, os testes se classificam de acordo com a área a ser testada. Como cita (SÁ, 2010)

Tabela 1 – Tipos de Testes de Software

Nível de Teste	Descrição
Teste de Componentes	Caracteriza-se pela busca por defeitos e verifica as funcionalidades dos componentes do <i>software</i> (módulos, programas, objetos, classes) que são testados separadamente;
Teste de Integração	Visa testar as interfaces entre componentes, interações para diferentes partes de um sistema como um sistema operacional, arquivo de sistema e hardware ou as interfaces entre eles;

Teste de Sistema	Relaciona-se ao comportamento do todo o produto/sistema conforme definido pelo escopo de desenvolvimento de um projeto ou produto. O principal foco do teste de sistema é a verificação frente aos requisitos especificados;
Teste de Aceitação	Teste de aceitação diz respeito às necessidades do usuário, requisitos, e processos de negócios realizados para determinar se aceita ou não o sistema;
Teste de Unidade	Cada módulo de <i>software</i> é testado separadamente com o objetivo de identificar erros de lógica e de implementação. Como há inter-relação entre os módulos, torna-se necessária a criação de pseudocontroladores (drivers) e/ou pseudocontrolados (stubs)
Teste de Integração	Representa uma progressão ordenada de testes em que os elementos individuais de <i>software</i> , os elementos de hardware, ou ambos sejam combinados e testados até que todo o sistema tenha sido integrado.
Teste de Sistemas	Teste de sistema completo: Nessa fase, o objetivo é garantir a funcionalidade correta da combinação do <i>software</i> com os demais elementos do sistema (hardware e banco de dados, por exemplo).

Fonte: Microsoft Word - HindenburgoElvasGonçalvesdeSá_Dissertação_VF03.doc (usp.br)

Um ponto a se considerar são essas classificações generalistas em algumas situações engessam a mente, de modo a limitar a imaginação para outras possibilidades. Por exemplo, já que toda documentação tem como objetivo a padronização de uma atividade, de forma a automatizar eliminando erros que poderiam ser feitos por iniciantes. Uma vez que temos padrões de teste a serem feitos não nos preocuparemos - em primeira mão - com em pensar preventivamente sobre o *software*, em específico, que está sendo desenvolvido. Essa atitude gera desvios, uma vez que em cada especificidade se deve testar algo diferente.

Em um sistema bancário se deve testar coisas específicas como validação e consultas que às vezes não estão descritas nos testes funcionais e de integração, como o *backup*.

Além da classificação geral, temos subdivisões como as que focam somente em testes do código fonte, testes somente na funcionalidade ou na mistura dos dois. Se tem subdivisões de modo a testar partes mais específicas e detectar erros pontuais que, por meio de uma procura em geral, não poderiam ser obtidos.

- caixa branca, em que existe conhecimento da composição interna do programa e o teste é realizado seguindo diversos caminhos lógicos e observando o que se passa internamente;
 - caixa negra, em que não existe conhecimento a composição interna do programa, o teste é realizado com base nas especificações e observando o comportamento externo. As condições de teste tendem a crescer sem limites e em termos práticos é apenas testado um subconjunto dos caminhos possíveis;
 - caixa cinzenta, que é uma mistura dos dois modelos descritos.
- (FERREIRA, 2016, p. 21)

O teste realizado pelo BAIT é chamado de teste funcional ou teste de caixa preta, em algumas bibliografias. Os casos de teste de *software* têm como definição ser, segundo (SÁ, 2010, p. 25) “[...]um conjunto de dados de entrada de teste, condição de execução e resultados esperados.” Cada caso de teste busca simular um ambiente, em que as características que serão obtidas por meio dos dados de entrada, possam ser testadas.

Dentre as técnicas de IA já criadas temos, segundo (BEZERRA, 2020) as seguintes:

- A. Applitools – Útil em validações visuais de testes funcionais usando uma linha base para comparar capturas de tela do aplicativo;
- B. Testim – Responsável pela criação de testes de código;
- C. Sealights – Identifica alterações no código e qual cenário será impactado;
- D. Test.AI – Reduz a complexidade do código por meio da execução de testes com Selenium ou Appium;
- E. MABL – Utiliza validação visual para modificar o teste de acordo com a variação na interface;
- F. ReTest – Aplicado em teste Manuais de regressão;

- G. Spidering AI – Cria cenários através de Aprendizado de Máquina;
- H. ReportPortal – Analisa e cria relatórios de testes automatizados ;
- I. Functionize – Usado para a criação e manutenção de testes sem a necessidade de código;
- J. Sofy.AI – Utiliza validação visual para testar qualidade, confiabilidade e acessibilidade;
- K. Appsurify – Identifica teste de risco e retorna relatórios de prioridades deles;
- L. Endtest.io – Extensão do navegador que analisa e auxilia a criação de testes;
- M. Screenster.io – Automatiza teste de interface de usuário;
- N. TestComplete – Voltado para teste de usuário em múltiplas plataformas.

Metade das técnicas descritas pelo autor são baseadas em aprendizado de máquina, que devido a repetição exaustiva requerida pelas modalidades de teste são as que mais se adequa ao ambiente tanto em facilidade de construção com em implementação e aprimoramento. Ela possui boa parte de seus códigos já construídos por causa do desenvolvimento par áreas médicas no diagnóstico de doenças.

Dentre a IA's já desenvolvidas as mais promissoras na testagem são as que envolvem a criação de cenários, podendo simular testes de usabilidade a interação de usuários com deficiências físicas ou com peculiaridades que variam desde a aptidão com a tecnologia até velocidade de identificação. Tais cenários que validados por um profissional de qualidade de *software* pode até servir para o próprio aperfeiçoamento da técnica de inteligência artificial.

2.2 TÉCNICAS DE INTELIGÊNCIA ARTIFICIAL

A Lógica *fuzzy*, ao contrário das tradicionais, tem como foco a separação em conjuntos atribuindo valores entre .0,1 (SÁ, 2020). Assim sendo a lógica difusa – outra nomenclatura dada – possui um mecanismo classificatório que separa um determinado elemento em conjuntos de pertença. Como se uma determinada coisa não fosse mais medida em verdadeiro e falso, mas em proximidade com um “conjunto verdade”. Ela serve, diretamente, para julgar informações imprecisas.

Já em outros casos para treinar uma inteligência computacional temos outra das técnicas mais usadas que é a matriz de confusão, que como diz (KOHAVI; PROVOST, 1998, p.271): “[...] Matriz de confusão – A matriz de confusão (*confusion matrix*) é usada para termos uma cenário mais completo quando estamos avaliando o desempenho de um modelo” isto é, ela determina por meio de comparações se a identificação foi feita corretamente, classificando-as em verdadeiros negativos, verdadeiros positivos, falso negativo e falsos positivos como mostra a tabela abaixo.

2.2.1 Matriz de Confusão

Para ter uma maior precisão nas identificações realizadas por algoritmos usa-se a Matriz de confusão ou teste de confusão.

Quadro 01 – Matriz de Confusão

Matriz de confusão	Noticia Falsa	Noticia Real
Noticia Falsa	Verdadeiro Negativo	Falso Positivo
Noticia Real	Falso Negativo	Verdadeiro Positivo

Fonte: Autor (2020)

O treinamento ocorre comparando a se o aplicativo acertou a predição. Pegamos uma notícia verdadeira, se o reconhecimento der como verdadeiro positivo quer dizer que a predição de que a notícia é verdadeira corresponde à realidade. Se, no entanto, for verdadeiro negativo, significa que embora a notícia seja verdadeira o algoritmo a identificou como falsa. Os dois restantes seguirão o mesmo padrão. Assumindo que a notícia é falsa, se resultar um falso negativo significa que a predição

acertou e correspondeu a realidade, se resultar em falso positivo equivale ao resultado que a predição errou.

- True positives (TP): os casos para os quais o classificador previu "real news" e as notícias eram realmente "real news".
- True negatives (TN) : os casos para os quais o classificador previu "fake news" e as notícias eram realmente "fake news".
- False positives (FP) : os casos para os quais o classificador previu "real news", mas, as notícias eram "fake news".
- False negatives (FN) : os casos para os quais o classificador previu "fake news", mas, as notícias eram "real news" (COSTA, 2019, p. 57)

Além disso, se tem uma classificação geral quando trabalhamos com testagem de um *software* para determinar sua qualidade. É a distinção entre defeito, engano, erro, falha. Elas são essenciais na classificação e geração da documentação especializada de como agir - e em que ponto do código - que poderia estar ocasionando determinado comportamento do sistema, em códigos muito grandes e complexos isso se torna imprescindível ao eliminar o desgaste do vasculhamento extensivo, e melhorar o clima organizacional da empresa ao qual trabalha ou presta serviço.

[...]O padrão IEEE número 610.12-1990 [79] diferencia os termos: defeito (fault) – passo, processo ou definição de dados incorreto, como por exemplo, uma instrução ou comando incorreto; engano (mistake) – ação humana que produz um resultado incorreto, com por exemplo, uma ação incorreta tomada pelo programador; erro (error) – diferença entre o valor obtido e o valor esperado, ou seja, qualquer estado intermediário incorreto ou resultado inesperado na execução do programa constitui um erro; e falha (failure) – produção de uma saída incorreta com relação à especificação. Neste texto, os termos engano, defeito e erro serão referenciados como erro (causa) e o termo falha (conseqüência) a um comportamento incorreto do programa[...]. (BARBOSA, 2000, p. 5)

O *Python* por ser uma linguagem interpretada simplificada elimina boa parte desses problemas por ter um número de definições e especificações reduzidas se comparada ao Java ou a linguagem C sendo essa uma das razões de sua liderança nas utilizações por programadores. Atualmente temos sistemas de apoio, extensões que podem ser criadas para linguagem antigas dando uma determinada vantagem, como as que completam a palavra ou trecho reservado, - no *javascript* se inicia com *back...* ele já sugere sua complementação com as variações – facilitando a codificação ágil e eliminando por parte dos enganos e que produzem vários ao sistema no momento de codificar.

Cada teste atenta contra um desses pontos classificados, os teste denominado de caixa branca, negra ou cinza buscam a eliminação dos defeitos e enganos que o código, em seu estado nu, possa ter, o código-fonte. Já o teste de sistema visa encontrar falhas e erros na execução ou nas saídas, que são relacionados direto aos requisitos levantados, por exemplo, se é esperado que uma determinada aplicação emita uma nota fiscal, durante o teste ela deve emitir uma nota fiscal e replica-la para o comprador e os demais envolvidos.

Os demais testes visam de modo misto a verificação das mesmas coisas em diferentes momentos e com mais componente integrados, já que o aconselhável no ambiente é testar por etapas e não somente quando a aplicação estiver finalizada. Em uma subclassificação os erros se subdividem em erros computacionais e erros de domínio, com explica os autores:

“[...]erros computacionais – o erro provoca uma computação incorreta mas o caminho executado (seqüências de comandos) é igual ao caminho esperado; e erros de domínio – o caminho efetivamente executado é diferente do caminho esperado, ou seja, um caminho errado é selecionado[...]. (BARBOSA, 2000, p. 5)

Nesses se encontra constantemente em teste que visam verificar a integração ou que trabalham colhendo dados do banco de dados. Os computacionais geralmente ocorrem com banco de dados – pode ser em diversos outros componentes da aplicação – mas restritamente quando toda a leitura, chamadas e carregamentos são feitos corretamente, mas ou o banco está vazio ou alguma informação não foi tratada e não está podendo ser coletada por referência incorreta, tipo palavras com acentuação gráfica. Os de domínio, já são em integrações onde um componente não está se conectando a outro, em alguns casos até – tomando a nota fiscal como exemplo- a informação a ser replicada é salva no local errado.

Essa técnica de Inteligência computacional (IC) pode ser aplicada no teste que envolve mecanismos de coleta de informações. Imaginemos um sistema de classificação de *Fake News*. Todo sistema com essa finalidade tem como objetivo a busca, de uma determinada informação, e posteriormente a validação dela por meio de comparações objetivas com outras informações ou fonte de informações.

Se o sistema obtém a informação de que o *Lockdown* é efetivo, por meio da busca e classificação das bibliografias existente - que ele tem acesso – ele atribui uma nota, um valor, que o joga para perto ou distante no conjunto atribuído como verdadeiro, e se aquela premissa que gerou a pesquisa estiver com um nota próxima

a esse, ela se classificará como provável verdade, ou “Não é uma *Fake News*”, podendo até – dependendo da objetivo do sistema retornar uma porcentagem de verificação, como por exemplo: Notícia com 50% de chance de ser verdadeira com base em 10.000 notícias analisadas. A partir daí se pode aplicar uma especialização por meio do teste, pode ser até o refinamento do resultado através de umas análises das matrizes de confusão.

Os alunos do 8º período de Sistemas de informação, em uma atividade para obtenção de nota na disciplina de Programação Visual para Web realizaram a construção de sistema para validação de notícias, em que ele teria que classificar as notícias em verdadeira ou falsa e, de acordo com os resultados trazidos da internet por meio de verificações constantes na bibliografia sendo a principal as contidas no Wikipedia, e retornar a porcentagem de *Fake News*.

Para isso adotaremos o conceito de Fake News descrito (COSTA, 2019, p. 17) “[...] fake news são definidas como informações falsas, muitas vezes sensacionalistas, divulgadas sob o disfarce de notícias” que consiste no fato de alterar uma determinada informação ou contexto de forma a prejudicar alguém, alguma pessoa pública, instituição. Ela se usa do artifício que o compartilhamento de informações na rede mundial de computadores é sem nenhum controle rigoroso para se propagar, da mesma forma que um vírus em um sistema imunológico deficiente. Isso somado ao fato que a maioria das pessoas não abre a chamada da notícia para ler e, compartilha e tira seu entendimento somente da legenda da mesma, causa grande dano e acarreta até no que atualmente estão chamando de assassinatos de reputação.

Para (WARDLE, 2017) existem algumas subclassificações das *Fake News* descritas como:

- A. Falsa conexão – A chamada da notícia e o conteúdo é divergente;
- B. Conteúdo enganoso – Afirmações falsas para difamar a pessoa;
- C. Falso contexto – Conteúdo verdadeiro em um contexto falso;
- D. Conteúdo Impostor – Usa o nome de uma pessoa ou marca mais o conteúdo é falso.

Nas figuras 4 e 5 são mostradas a parte visual e a de código da aplicação desenvolvida pelos alunos.

Figura 04 – Print da aplicação soliverdades?



Fonte: Autor (2020)

Foi implementado o mecanismo de busca onde era realizado a consulta e retornava as fontes contidas na *Wikipedia* para averiguação do requerente da busca, todo o código pode ser acompanhado na figura a seguir.

Figura 05 – Código da aplicação

```

File Edit Selection View Go Run Terminal Help
index.html - soliverdades - Visual Studio Code

SOURCE CONTROL
index.html X
index.html > html > body > div.main.container-fluid > div.container.wiki.info
Message (Ctrl+Enter to com...)
Changes

events_runner-d5e4bf42585b8da8c18f7d963dbfc17cd66a79aa586c9448c4de8d6952ee9d97.js'</
script><script src="//production-assets.codepen.io/assets/editor/live/
css_live_reload_init-25d1423d5d6fb975e7d61832d2c061422a94963ca446583b965dfc5569147311.js"></
script><meta charset="UTF-8"><meta name="robots" content="noindex"><link rel="shortcut icon"
type="image/x-icon" href="//production-assets.codepen.io/assets/favicon/
favicon-8ea04875e70c4b0bb41da869e81236e54394d63638a1ef17fa558a4a835f1164.ico" /><link
rel="mask-icon" type="" href="//production-assets.codepen.io/assets/favicon/
logo-pin-f2d2b6d2c61838f7e76325261b7195c27224080bc099486dd6dccb469b8e8e6.svg" color="#111" /
><link rel="canonical" href="http://codepen.io/waynebunch/pen/QEbaNJ" />

<link rel="stylesheet prefetch" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/
bootstrap.min.css">
</link rel="stylesheet" href="/css/main.css">
</head><body>
<div class="container-fluid" id="main">

<a href="https://en.wikipedia.org/wiki/Special:Random"><h1 class="text-center">WikiSearch</h1></
a>
<a <div class="container">
<div class="row">
<div class="col-lg-6 col-lg-offset-3">
<div class="input-group">
<input class="form-control text-center" type="search" name="search" results="10"
id="userInput" onsearch="onSearch(userInput)" />
<span class="input-group-btn">
<button class="btn btn-secondary" id="submitButton" type="submit" onclick="onSearch()"
>Buscar</button>
</span>

```

Fonte: Autor (2020)

O código faria uma busca no *Wikipedia* e através de marcações consultaria as fontes descritas nas notas no final da página. De acordo com o retorno das referências terciárias que seria as que estão listados na própria enciclopédia calcularia a porcentagem. Caso as referências descritas não batessem, o algoritmo retornaria verdadeiro para *Fake News* somente a porcentagem. À medida que o banco de consulta fosse expandido o processo ficaria mais refinado, com a adição de consultas no Google Acadêmico, que é considerado confiável na obtenção de informações muito confiáveis em teses, livros, dissertações etc.

A questão da acurácia, na verificação em notícias, o problema que encontramos na busca foi da correlação de notícias - tendo em vista que muitos dos estudos da *Wikipedia*, consultados, através da referência estão defasados – foi em definir o a notícia. Pegando por exemplo o que se sabia sobre a eleição do presidente dos Estados Unidos da América em 2016, foi elencado que ele só foi eleito por meio de notícias falsas divulgadas, dados isso, se criaram várias notícias relacionadas a essa questão e em uma data posterior foi descoberto que não. Toda as notícias produzidas pelos meios de comunicação da época deveriam, a luz das novas descobertas serem classificadas com erradas ou não mais precisa em virtude dos novos fatos, mas esse processo classificatório não é feito, assim – usando o próprio conceito de conteúdo manipulado – a informação pode ser compartilhada novamente induzindo ao erro e levando muitos a seguir o mesmo processo ao qual na época foi acusado o adversário.

Sendo assim até o próprio treinamento no aplicativo que foi criado ficou prejudicado, mesmo adotando como referência portais da grande mídia, tidos como referência no âmbito da segurança da informação dada, recaia nesse mesmo problema, pela falta de um controle retroativo nas informações.

Nas notícias da grande mídia em geral também foi identificado outro problema durante a construção do algoritmo que foi o fato de a chamada não corresponder a nenhum conteúdo de fato, de certa forma pode-se descrever como falso contexto, é uma notícia descritiva, mas na hora ela não agrega nenhuma informação fora a que está escrita na chamada, a equipe de desenvolvimento da aplicação até deduzia, que a notícia em si não foi criada para ser lida como um todo mas para induzir a uma conclusão somente pela leitura e compartilhamento da chamada ou legenda da mesma.

2.2.2 REDE NEURAL

Segundo (HAYKIN, 2001) uma rede neural é uma forma de processar maciçamente - e em paralelo - distribuído que tem uma unidade de processamento simples e pode armazenar conhecimento experimental para o uso posterior. Semelhante aos neurônios cerebrais ele processa o conhecimento através de conexões isoladas que se comunicam e, através dela, constroem uma “*big data*” de informação. Esse banco de dados pode ser utilizado no seu aprimoramento e de outro ramo da inteligência artificial chamado de *machine learning*.

Outro aspecto das redes neurais são suas classificações segundo seu ramo de atividade e atuação. Ela se divide em 9 segundo Haykin mas para este artigo usaremos somente a 2 até o 5 que é vital para a elucidação de como ela pode ser aplicada a testagem na área delimitada pelos testes que envolve essa tecnologia, tendo em vista que cada modalidade da inteligência artificial – *machine learning*, redes neurais e computação cognitiva.

[...]3.Adaptabilidade. As redes neurais têm uma capacidade inata de adaptar seus pesos sinápticos a modificações do meio ambiente. Em particular, uma rede neural treinada para operar em um ambiente específico pode ser facilmente retreinada para lidar com pequenas modificações nas condições operativas do ambiente[...]. (HAYKIN, 2001, p. 29)

Essa característica pode ser aplicada ao teste de regressão que é o teste onde se testa uma funcionalidade da aplicação implementada está corrompendo ou danificando as outras. Pegando o exemplo anterior de WOW na implementação de um novo mapa, testa-se a integração e a regressão. Essa testagem feita corretamente por uma técnica inteligente gerando dados ou relatórios pode servir até para programações futuras onde o desenvolvedor já – com os dados dos erros na testagem anterior – elabora o código visando na prevenção a erros.

Esse mapeamento podia ser até implementado no próprio *Candy Crush Saga* se o objetivo dos desenvolvedores fosse coletar dados do equilíbrio do jogo, isto é, avaliar o desempenho de jogadores em cada nível e através dos dados coletados, como o número de vidas gastas, tempo na missão, uso de doces, associadas a um peso e salvas em um neurônio artificial, pudesse ter uma projeção de quais níveis poderiam ser alterados em uma futura remodelação do balanceamento do jogo.

A desvantagem é entender, absorver e explicar todos os dados obtidos, já que possuem uma exatidão. Mas não é o recomendado para teste de modo geral por sua difícil programação, exigem conhecimentos muito específicos e complexos e de difícil expressão por meio de linguagens simbólicas, isto é, a criação de regras.

[...]4.Resposta a Evidências. No contexto de classificação de padrões, uma rede neural pode ser projetada para fornecer informação não somente sobre qual padrão selecionar, mas também sobre a confiança ou crença na decisão tomada. Essa última informação pode ser utilizada para rejeitar padrões ambíguos, caso eles estejam presentes, e com isso melhorar o desempenho de classificação da rede[...]. (HAYKIN, 2001, p. 30)

Nesta pode-se implementar no teste de caixa-cinza que é o que verifica o código fonte da aplicação e o meio externo dela. Onde pode facilmente retirar dos códigos as incongruências e divergências tal como erros na codificação em que a própria linguagem não consegue identificar. Também pode ser aplicada no teste de integração, procurando os erros na transmissão e replicação da informação. Também para vasculhar um banco de dados criado, verificando se ele está devidamente construído, alguns campos de banco não podem ter informações replicadas ou que duas informações ambíguas associadas a uma mesma chave primária, como dois nomes associados a um mesmo cadastro de pessoa física (CPF).

[...]5.Informação Contextual. O conhecimento é representado pela própria estrutura e estado de ativação de uma rede neural. Cada neurônio da rede é potencialmente afetado pela atividade de todos os outros neurônios na rede. Consequentemente, a informação contextual é tratada naturalmente pela rede neural[...]. (HAYKIN, 2001, p. 30)

Com isso toda a informação obtida por meio do teste de carga pode ser analisada para saber se além de suportar o grande tráfego ele está obtendo o material certo e relevante. Mas como descrito no outro tópico a informação é de difícil compreensão, podendo ocorrer casos extremos onde ela ao ser interpretada pela própria rede, pode criar mecanismo de classificação e quantificação incompreensíveis a mente humana, como no caso do Bob e da Alice, *bots* desenvolvidos pelo *Facebook* para simular trocas entre si, que no decorrer do processo deduziram falhas e mudaram seu jeito de falar, começando a criar uma linguagem própria.

Isto pode exemplificar o quanto este estado de tratamento de informação pode ser potencialmente prejudicial ao processo de teste de *software*. Em contrapartida em uma classificação entre notícias verdadeiras ou falsas se poderia usar esse método para classificar o nível de veracidade de uma afirmação baseada

no número de vezes que ela é replicada em notícias e de sites confiáveis. Como nas citações bibliográficas, um livro muito citado é, geralmente, o que possui seu conteúdo já muito aceito na comunidade acadêmica e – de certa forma – consolidado no meio científico. Mesmo assim, uma mentira repetida mil vezes não pode se tornar uma verdade, logo, tem sempre que ter um fator de peso determinante.

Um a ser atribuído a esse mecanismo seria a de não contradição, criar um método onde a rede neural não analise somente o texto em si, mas o corpo envolta, não em sua completude, mas para aprimorar o reconhecimento não classificando artigos que a confirmam e artigos que refutam o trecho como o mesmo, assim separando-os de maneira a ser usados futuramente em outras classificações ou aprimoramentos da própria técnica por meio da matriz de confusão e demais técnicas.

No método descrito pelo autor a seguir também se dá outra forma de verificação e assimilação e comparação do conhecimento:

[...]Sistemas de computação baseados em redes neurais tem a capacidade de receber ao mesmo tempo várias entradas e distribuílas de maneira organizada. Geralmente, as informações armazenadas por uma rede neural é compartilhada por todas as suas unidades de processamento. Característica que contrasta com os atuais esquemas de memória, onde a informação fica confinada em um determinado endereço[...]. (MARCHADO, 2012, p. 61)

Dessa forma o problema identificado na questão anterior seria resolvido de maneira mais prática, uma vez que o conhecimento é compartilhado automaticamente entre as redes de neurônios artificiais, a sua validação quantificável seria mais fácil, tal como sua associação a porcentagens amostrais que era o objetivo inicial do projeto desenvolvido pelos alunos do 8 período.

2.2.3 ALGORITMOS GENÉTICOS

Os algoritmos genéticos são uma subclassificação dos algoritmos evolucionários que segundo (LIDEN, 2008) tem o objetivo de usar uma simulação dos processos naturais da teoria da evolução como uma alternativa para resolver problemas em meios computacionais. Assim sendo, terão a capacidade de evoluir através de uma seleção, mutação e reprodução.

Esse caráter evolutivo já pode ser integrado no aprimoramento de uma determinada funcionalidade do sistema onde ela terá que repetir constantemente uma determinada operação e precisa verificar métodos de realizar a tarefa de forma mais rápida e com menos erros de processo.

[...] Algoritmos genéticos são algoritmos de busca baseados nos mecanismos de seleção natural e genética. Eles combinam a sobrevivência entre os melhores com uma forma estruturada de troca de informação genética entre dois indivíduos para formar uma estrutura heurística[...]. (LIDEN, 2008 p. 44)

Como dito, são uma técnica de busca baseada em seleção natural, isso aplicados ao teste de unidade diminui a carga de repetição. Imaginemos um sistema, tal como no exemplo anterior, através de AG ele precisara localizar erros em várias unidades do sistema antes de passar para a testagem de integração, ao identificar e aplicar uma solução ao problema ele verificará se essa solução também é eficaz em outras unidades que apresentem o mesmo problema, se for correspondente o AG irá reforçar a solução e replica-la, baseado no modelo natural de mutação e reprodução.

Entretanto mesmo com essa solução sendo eficaz, não deixará de aplicar outras, procurando assim melhores e mais rápidas. O mesmo algoritmo evolutivo pode ser utilizado para uma classificação e catálogos de erros durante os ambientes de testagem, procurando os mais recorrentes e danosos ao *software*, sistema, aplicação.

2.2.4 APRENDIZADO DE MÁQUINA

Outra das técnicas disponíveis ao ambiente de testagem é o aprendizado de máquina (AM) que consiste em um determinado algoritmo auto aprimorar conforme vai coletando dados.

[...]A área de AM é responsável por pesquisar métodos computacionais adequados para a aquisição de novos conhecimentos, novas habilidades e novas formas de organização do conhecimento já existente. O aprendizado possibilita que o sistema faça a mesma tarefa ou tarefas sobre uma mesma população de uma maneira mais eficiente a cada execução[...]. (MACHADO, p. 42)

Essa técnica aplicada a mapeamentos de funcionalidades nos testes de usabilidade como no jogo *Candy Crush Saga* gera relatórios de como o melhor usuário poderia se comportar perante a aplicação – o que no *marketing* se denominam de

persona – o pior e o mediano, desta forma terão como base o tempo médio que a aplicação responderá a cada tipo de usuário. Na obtenção de soluções aos erros computacionais, quando deparados com problemas complexo durante a testagem. Boa parte desses problemas estão sendo discutidos em fóruns ou até referenciados em livros.

2.2.4.1 Aprendizado Supervisionado

Dentre o aprendizado de máquina se tem duas subclassificações que se subdividem em 2 terciárias, cada uma. O aprendizado supervisionado como explica (MACHADO, 2012) é aquele tipo onde o algoritmo já é alimentado com classes predefinidas para usar como base na criação de regras de identificação de outras classes não nomeadas. Esse método de predição se divide em:

- A. Classificação – Identificação de conjuntos menos usuais;
- B. Regressão – Previsão de uma constante numérica através de um mecanismo retroativo.

2.2.4.2 Aprendizado Não Supervisionado

Ao contrário, o não supervisionado não usa predefinição, os conjuntos são analisados para obtenção de similaridades entre eles, como explica (MACHADO, 2012, p. 45) “[...]O modo de aprendizagem não-supervisionado supõe que o conjunto de exemplos não está rotulado, com isto o sistema tenta classificar estes conjuntos agrupando os semelhantes em determinadas classes.”

- A. Agrupamentos – Junção de objetos com atributos parecidos;
- B. Associação – Similaridades entre atributos.

Em outras bibliografias são elencados mais 3 tipos de técnicas de aprendizado de máquina que são, segundo o (GRANATYR, 2016) as:

- A. Detecção de desvios – Identificar objetos fora de um padrão estabelecido;
- B. Padrões sequenciais – Busca identificar padrões em uma linha de tempo, não somente as no mesmo período temporal;

C. Sumarização – Extração de características para traçar um perfil de cada conjunto estabelecido;

Em casos em que há uma passagem de códigos ou uma criação de aplicação orientada a reuso que é a equipe pegar um código já feito e adapta-lo a sua necessidade – em se tratando de códigos muito extensos – se pode usar a detecção de desvios alinhadas ao reconhecimento fotossensível para identificar os trechos que o próprio mecanismo da linguagem apontar como erro.

Já a sumarização aplicada junto com os padrões sequencias pode elencar se os testes empregados no seu processo seguem os aplicados em *softwares* referenciais no mundo empresarial e tecnológico, tal como indicar quais os melhores métodos de testagem e em que áreas deve ocorrer essa testagem. Mas, em contrapartida é necessário ter em mente que embora cada técnica facilite o processo sempre é necessário adequar as necessidades que o programa em questão a ser testado exige.

2.2.5 DEEP LEARNING

O ambiente de IA por ser recente está em constante desenvolvimento por isso, além das modalidades criadas separadamente se tem a tentativa de união entre duas que resolvem problemas ou atuam em áreas que podem ser complementadas como o Aprendizado profundo que a união de uma rede neural com um aprendizado não supervisionado.

Deep learning é um conjunto relativamente novo de métodos que está mudando o aprendizado de máquina de formas fundamentais. O deep learning não é um algoritmo propriamente dito, mas uma família de algoritmos que implementam redes profundas com aprendizado sem supervisão. Essas redes são tão profundas que novos métodos de cálculo, como GPUs, são necessários para construí-las (além de clusters de nós de cálculo). (JONES, 2017, n/p)

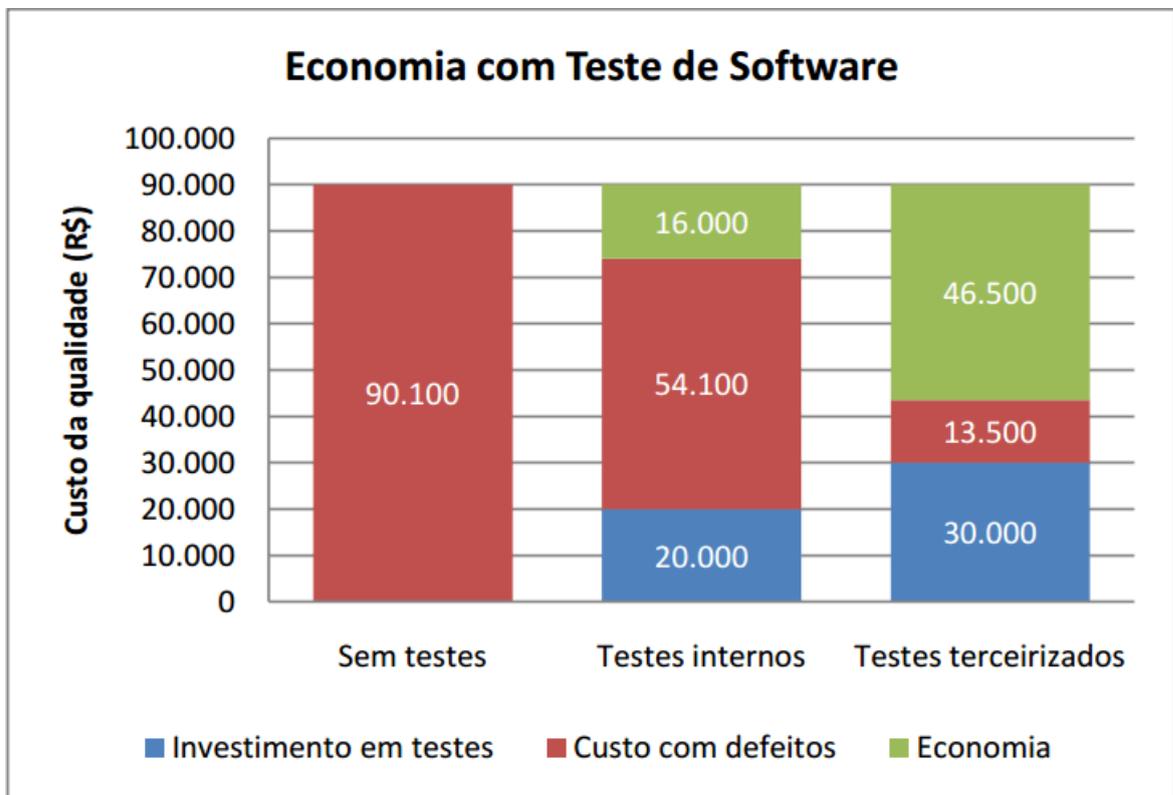
Desse modo no desenvolvimento de uma técnica IA podemos até usar como base duas outras que resolvem problemas específicos para resolver o problema de testagem que se apresentou em determinado momento. As técnicas de aprendizado profundo são indicadas na resolução de problemas mais complexos provindo de teste de qualidade de *software*.

2.3 IMPACTOS DA APLICAÇÃO DE INTELIGÊNCIA ARTIFICIAL

Quando tratamos de qualidade de *software*, que é obtida por meio dos resultados coletados nos testes das variadas modalidades, um ponto a ser sanado é se os requisitos levantados foram atendidos e estão em pleno funcionamento como previsto pelo cliente e projetados pelos desenvolvedores do projeto.

Para (ANDELKOVIC, 2020, n/p) “[...]é importante envolver pessoas que tenham a determinação de fazê-lo funcionar. Nos projetos de IA, facilmente se pega um problema grande demais, o que atrasa o tempo necessário para ver os resultados reais para determinar se o problema foi resolvido.”. Ele cita que teve como problema a dificuldade de achar profissionais que trabalhem com inteligência artificial. Devido à alta dificuldade os profissionais atuantes são escassos, mesmo tudo estando migrando para esse mercado de trabalho.

Gráfico 02 – Economia com Teste de Software sem IA



Fonte: onedaytesting (2019)

Como descrito no título da figura, está exemplificado a economia sem a utilização de técnicas de testagem inteligente. O ganho com a aquisição de uma empresa terceirizada é maior em virtude de o corpo de integrantes já serem devidamente especializados nesse ramo de teste. Mas ainda se em vez de uma empresa terceirizada poder utilizar uma técnica que produza não somente a economia de uma empresa terceirizada, mas por ser um IA e coletar dados de um mar de ambientes e empresas de teste proporcionar um nível de retorno ainda maior que ambientes profissionais terceirizados.

[...]A experiência de campo do autor mostra ganhos em produtividade (t/h) e energéticos (Kcal/t) de até 3% a 5% em relação a operadores especialistas. O detentor da tecnologia aponta diversas aplicações com ganhos de até 6% a 10% em produtividade e de até 3% em eficiência energética. Outras experiências internacionais também são relatadas[...]. (SELLITTO, 2002, p.375)

O autor elenca os resultados da aplicação de inteligência artificial na indústria que além de automatizar o processo reduziu os custos e maximizou os lucros em produtividade. Essa aquisição gerou similares resultados como aponta o autor ao citar testes internacionais.

[...]As vantagens relatadas decorrem principalmente da melhor resposta obtida por um sistema automatizado às variabilidades do processo, em oposição às incertezas do comportamento de especialistas humanos, principalmente em situações de emergências, períodos longos e contínuos de operação e necessidade de alta produtividade. A capacidade de aprender e de assumir riscos também auxilia na construção deste resultado[...]. (SELLITTO, 2002, p.375)

Como descrito a principal vantagem é na constância de alta produtividade por longos períodos de tempo, uma vez que o impacto é menos por um objeto mecânico ou lógico não precisar de descanso. Outra questão é o conjunto de especialidades suportadas e aprendidas, uma vez que o especialista só poderá agir dentro do campo que já estudou até o momento, o sistema especialista pode estruturar uma árvore de aprendizado e decisões simultâneas respondendo melhor e, a longo prazo, a variabilidades do processo.

No aspecto psicológico o impacto é maior, uma vez que suas decisões não consideram fatores como a desempregabilidade de um contingente do setor, suas tomadas de decisão são mais diretas e com maior custo benefício para a empresa podendo até eliminar aquisição de empresas terceirizadas.

Obtém-se também com a implementação de uma testagem inteligente no fator custo e economia, um ganho mais acentuado pela diminuição de pessoal necessário e posteriormente, a alta especialização que decorrerá de um aperfeiçoamento constante das técnicas. A identificação de defeitos será mais rápida e precisa, podendo adiantar o lançamento e gerar mais valor agregado em virtude da diminuição potencial dos erros.

[...]o processo de desenvolvimento usado tenha uma influência significativa sobre a qualidade de software e que bons processos são mais suscetíveis de conduzir o software de boa qualidade. O gerenciamento e a melhoria de qualidade de processo podem gerar softwares com menos defeitos. Contudo, é difícil avaliar os atributos de qualidade de software, como manutenibilidade, sem usar o software por um longo período[...]. (SOMMERVILLE, 2011 p.458)

Já o uso dessas técnicas por um longo período só dará mediante ao aprimoramento das mesmas em ambiente de desenvolvimento dedicados. O problema é apontado por (SILVA, 2005, p. 8) “[..] As principais utilizações se restringem ao paradigma de regras para modelar sistemas baseados em conhecimento que auxiliam o desenvolvimento de *software*.”. Cada IA tem sua modalidade específica de uso, sua construção de modo completo, atendendo a todos os testes de *software* é inviável com a tecnologia atual.

Eventualmente se tem como impacto a curto prazo a busca das universidades e escolas técnicas a abrir cursos profissionalizantes na área de desenvolvimento de sistemas inteligentes. Podemos até ramifica-los, já nesses 3 temas mais recorrentes na inteligência artificial, que são as redes neurais, aprendizado de máquina e computação cognitiva.

Mas primeiro, para isso deve-se fomentar que os próprios professores da área de sistemas, engenharia de *software*, qualidade de *software* e TI em geral, busquem mais informação e formação sobre eles, tal como os lecionadores de cursos em áreas de tecnologia da informação incentivarem seu alunos a pesquisas relacionadas ao desenvolvimento de IA's, já que é um mercado emergente, facilitará tanto para suas pretensões salariais como para a estabilidade, tendo em vista que a grande preocupação da população é que essa etapa da revolução tecnológica venha a tirar empregos como foi a industrial.

[...]Estamos começando a analisar questões como: "Se executarmos uma instância do BAIT e ele encontrar uma falha após a execução por algumas horas, o que acontecerá se 1000 instâncias do BAIT estiverem em execução no mesmo período de tempo? Vamos encontrar mais falhas? Ter um bot de controle de qualidade com inteligência artificial nos permite ter esses tipos de discussões estratégicas de teste e fazer tentativas/ testes. Meu pressentimento é que isso nos beneficiará no futuro, e acho que isso pode mudar o controle de qualidade, levando os testes a um novo nível com essa abordagem de tecnologia de ponta[...]. (ANDELKOVIC, 2020, n/p)

Com o implemento desse nível de algoritmo inteligente podemos elevar o padrão de qualidade dos *softwares* a um novo patamar, onde discussões sobre o melhoramento de outras áreas não exploradas podem vir à tona melhorando ainda mais, a experiência do usuário como, além do foco que cada desenvolvedor tem em proporcionar uma boa experiência a usuário sem deficiência física, poder começar a investir em painéis de acessibilidade de acordo com a limitação imposta pelo corpo do usuário ao lidar com a aplicação.

3 RESULTADOS E DISCUSSÕES

Com base no que foi apresentado temos como elencar atualmente as técnicas que mais estão sendo usadas em ambientes de desenvolvimento. Elas são as que em curto prazo, sendo fácil ou de difícil compreensão são as que mais praticamente podem ser aplicadas, seja nas áreas médicas – onde seu maior capital é empregado – quando nas áreas financeiras, industriais e até em testes de softwares. Como afirma Costa, ao apresentar a relação do sistema desenvolvido para a classificação de notícias verdadeiras ou falsas.

[...]as áreas de classificação de texto, processamento de linguagem natural, Machine learning, Deep Learning e os métodos de word embedding foram estudadas e exploradas. O objetivo de combinar estas áreas é obter modelos preditivos capazes de classificar as notícias em fakes ou reais. Os modelos testados utilizaram uma base de dados de notícias fakes e reais[...]. (COSTA, 2019, p. 68)

Como toda averiguação de eficácia a combinação e recombinação são fatores chaves para determinar a eficiência de alguma técnica. Essas combinações geram em uma determinada função ao aumento de eficiência e em outras a redução, cabe ao desenvolvedor e gerente de teste, adequar a dosagem e criar relatórios sobre sua especificidade, junto com os resultados da experiência, como mostra a autor.

[...]Os resultados obtidos nos classificadores com as palavras de dimensionalidade 300 mostraram que os algoritmos de Deep Learning são muito eficientes quando trabalham com dados de alta dimensionalidade. Foram feitos testes com os modelos Word2vec e Glove em conjunto com os algoritmos de Machine Learning e Deep Learning. O método com o melhor resultado foi o Glove utilizado em conjunto com o algoritmo de redes neurais convolucionais obtendo uma acurácia de 97,50%. Já o método Word2vec o melhor resultado obtido foi utilizando o classificador Redes Neurais Feedforward obtendo uma acurácia de 91,37%[...]. (COSTA, 2019, p. 68)

Já para criação de casos de teste as mais populares são as Redes Neurais Artificiais e a Lógica *Fuzzy* o estudo feito por outro autor demonstrou comparativamente a eficiência das duas em casos de teste.

[...]as diferenças de valores entre os dois métodos, fuzzy e RNAs, no que diz respeito principalmente ao tempo que ambos os algoritmos gastam para executarem uma configuração básica de para os dois métodos. Estas diferenças por menores qu sejam pelo número de LCs, ou pela CC ou mesmo pelo TĒ. Ressalta-se, porém, que à medida que aumentam os valores de ambos os modelos que sejam regras de inferências ou neurônios, a diferença mantém-se, fuzzy menos que RNAs[...]. (SÁ, 2010, P. 74)

Vale ressaltar que sempre depende da necessidade da empresa ou da testagem qual levará a definitiva vantagem e será adotado como explica (ANDRADE, 2016, p. 54) [...]”A utilização da RNA para um fim específico tem relação direta com a eficiência com que esta fornece respostas próximas aos dados de saída reais.”. Não é eficaz escolher o mais rápido, de fácil programação, e que utilize menos memória senão conseguir atender a demanda exigida, por vezes algoritmos de aprendizagem profunda melhor se adequam as necessidades do que as próprias redes neurais e lógica *Fuzzy*.

4 CONSIDERAÇÕES FINAIS

Com base no exposto, foi possível identificar a existência de técnicas de inteligência artificial que já auxiliam os gestores de qualidade de *software* a desenvolver uma boa aplicação para seus clientes como também se tem a possibilidade de progressivamente construí-las mais rebuscadas e robustas, com combinações periódicas de aprimoramentos como as citadas no corpo do trabalho.

Percebe-se ainda, os impactos positivos na empresa e sociedade se dão desde o menor consumo de energia e melhora na tomada de decisões até o desenvolvimento de novas profissões estudos acadêmicos, como acontece em todas as etapas da revolução industrial e tecnológica onde empregos deixaram de existir, mas a novas profissões ou profissões do futuro surgiram.

As limitações para o desenvolvimento do trabalho foi o período de pandemia instaurado durante a elaboração deste trabalho de conclusão de curso onde prejudicou a coleta de dados de forma empírica, e através de entrevistas presenciais, sendo a questão do formulário pouco eficaz em virtude das adaptações dos funcionários a nova situação ainda não ter sido consolidadas e alguns até tendo mais tarefas do que em períodos normais.

Por fim, cabe as comunidades e ambientes de desenvolvimentos, como também aos profissionais de áreas afins, proporcionar projetos de pesquisa e extensão que visem a criação e progressão das técnicas de IA tendo como meta a chegada da singularidade, que é deixar de ser uma técnica e começar a ser um Inteligência Artificial pura.

REFERÊNCIAS

- ADES, César. 1993. **MÚLTIPLA MEMÓRIA**. Disponível em: <http://www.revistas.usp.br/psicousp/article/view/34470/37208>. Acesso em: 16 nov. 2020
- ANTUNES, Juliana et al. 2020. **Python a predição de dados usando redes neurais multicamadas**. Disponível em: <https://www.brazilianjournals.com/index.php/BRJD/article/view/14311/11924> Acesso em: 12 nov. 2020
- ANDELKOVIC, Alexander. Candy Crush Saga - Utilizando a Inteligencia Artificial nos testes. **Infoq**, 2020, 12 FEV. Disponível em: <https://www.infoq.com/br/articles/candy-crush-QA-AI-saga/>. Acesso em: 22 jun 2020.
- ANDRADE, Minéia. 2016. **REDES NEURAIAS ARTIFICIAIS: PRINCÍPIOS BÁSICOS**. Disponível em: <https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwie1uqmvbPtAhXpHbkGHcEOCpMQFjACegQIAhAC&url=https%3A%2F%2Fperiodicos.utfrpr.edu.br%2Frecit%2Farticle%2Fdownload%2F4330%2FLeandro&usg=AOvVaw35JXkDQWomZYhYGhVZhrIj>. Acesso em: 30 nov. 2020
- ARARIBOIA, G. **Inteligência Artificial**. Rio de Janeiro: Ed. LTC, 1988.
- BARBOSA, Ellen et al. 2000. **Introdução ao Teste de Software**. Disponível em: https://www.researchgate.net/publication/306255146_INTRODUCAO_AO_TESTE_DE_SOFTWARE. Acesso em: 12 nov. 2020
- BEZERRA, Aldrich. 2020. **IA EM AUTOMAÇÃO DE TESTES**. Disponível em: <https://www.base2.com.br/2020/05/12/ia-em-automacao-de-testes/>. Acesso em: 30 nov. 2020
- BORGES, Luiz, 2010. **Python para Desenvolvedores**. Disponível em: https://ark4n.files.wordpress.com/2010/01/python_para_desenvolvedores_2ed.pdf. Acesso em: 12 nov. 2020
- COUTINHO, Thiago, 2020. **Conheça 11 bibliotecas Python aplicadas na Ciência de Dados**. Disponível em: <https://www.voitto.com.br/blog/artigo/principais-bibliotecas-python>. Acesso em: 11 nov. 2020
- CONAI: Tutorial Inteligência Artificial. In: CONGRESSO NACIONAL DE AUTOMAÇÃO INDUSTRIAL. **Anais...** São Paulo, 1994. Acesso em: 29 set. 2020
- COSTA, Lucas, 2019. **Classificação de fake news utilizando algoritmos de aprendizado de máquina e aprendizado profundo**. Disponível em: <https://periodicos.unipampa.edu.br/index.php/SIEPE/article/view/106503>. Acesso em: 10 nov. 2020.
- CAMPOS, Jober. 2018. **COMPARAÇÃO ENTRE FERRAMENTAS PARA GERAÇÃO DE DADOS DE TESTE UTILIZANDO INTELIGÊNCIA ARTIFICIAL E ALEATORIDADE**. Disponível em: http://repositorio.roca.utfpr.edu.br/jspui/bitstream/1/10036/1/DV_COENS_2018_1_3.pdf. Acesso em: 30 set. 2020

CAPUANO, Ethel. **O poder cognitivo das redes neurais artificiais modelo ART1 na recuperação da informação.** 2009. Disponível em: https://www.scielo.br/scielo.php?pid=S010019652009000100001&script=sci_arttext&tlng=pt. Acesso em: 11 abr 2020

DELAMARO, Márcio. MALDONADO, José. JINO, Mario. **Introdução ao Teste de Software.** 2013. Disponível em: https://books.google.com.br/books?hl=ptBR&lr=&id=zqw4DwAAQBAJ&oi=fnd&pg=PT3&dq=Introdu%C3%A7%C3%A3o+Teste+de+Software&ots=bTm_f6cCL5&sig=gdCjm7Et8SgMKkX8kJCsO49tRk#v=onepage&q=Introdu%C3%A7%C3%A3o%20Teste%20de%20Software&f=false. Acesso em: 12 abr. 2020

FERREIRA, Fredelico. **Inteligência Artificial na Verificação e Teste de Software para Desenvolvimento Ágil.** 2016. Disponível em: <http://hdl.handle.net/10400.21/7119>. Acesso em: 30 set. 2020

GRANATYR, Jones. 7 TÉCNICAS DE INTELIGÊNCIA ARTIFICIAL PARA PROFIS SIONAIS DE TI GANHAREM MAIS DINHEIRO. **EXPERT.** Disponível em: <https://iaexpert.academy/wp-content/uploads/2016/08/eBook-Aprendizagem-de-Ma%CC%81quina.pdf>. Acesso em: 20 nov. 2020

GUIMARÃES, Antonio. **Um sistema difuso inteligente para avaliar informações de usuários na Internet.** 2002. Disponível em: <https://www.scielo.br/pdf/ci/v31n3/a04v31n3>. Acesso em: 11 abr 2020.

JONES, M. Tim. UM GUIA PARA INICIANTES SOBRE INTELIGÊNCIA ARTIFICIAL, APRENDIZADO DE MÁQUINA E COMPUTAÇÃO COGNITIVA. **IBM.** 2017. 01/06/2017. Disponível em: <https://developer.ibm.com/br/technologies/artificial-intelligence/articles/cc-beginner-guide-machine-learning-ai-cognitive>. Acesso em: 20 nov. 2020

LAKATOS,

LINDEN, Ricardo. **ALGORITMOS GENÉTICOS:** Uma importante ferramenta da Inteligência Computacional. Nº. 2. Local: Rio de Janeiro. BRASPORT Livros e Multimídia Ltda. Nº p. 401

MACHADO, Vinícios. **INTELIGÊNCIA ARTIFICIAL.** 2012. Disponível em: <https://sigaa.ufpi.br/sigaa/public/docente/monitoria.jsf?siape=1446435>. Acesso em: 20 nov. 2020.

MCCARTHY, John. **O QUE É INTELIGÊNCIA ARTIFICIAL?.** 2007. Disponível em: <http://www-formal.stanford.edu/jmc/whatisai/whatisai.html>. Acesso em: 22 jun. 2020

MENDES, Raquel, 1997. **INTELIGÊNCIA ARTIFICIAL: SISTEMAS ESPECIALIS TAS NO GERENCIAMENTO DA INFORMAÇÃO.** Disponível em: https://www.scielo.br/scielo.php?script=sci_arttext&pid=S0100-19651997000100006. Acesso em: 10 nov. 2020.

PASTERNAK, Guitta. **Do Caos à Inteligência Artificial.** 1992 Disponível em: <https://books.google.com.br/books?hl=ptBR&lr=&id=GrWMiCteeHgC&oi=fnd&pg=PA11&dq>

=inteligencia+artificial&ots=4c9Dd2wAJd&sig=kiOfg1xSJcNRFjMFEyiqyrD7A14#v=onepage&q=inteligencia%20artificial&f=false. Acesso em: 22 jun. 2020.

PEREIRA, Adriano, 2018. **INDÚSTRIA 4.0: CONCEITOS E PERSPECTIVAS PARA O BRASIL**. Disponível em: http://periodicos.unincor.br/index.php/revistaunincor/article/view/4938/pdf_808. Acesso em: 30 nov. 2020

PRINA, Bruno; TRENTIN, Romario. GMC: Geração de Matriz de Confusão a partir de uma classificação digital de imagem do ArcGIS®. In: XVII SIMPÓSIO BRASILEIRO DE SENSORIAMENTO REMOTO, S/Nº, 2015. **Anais eletrônicos** [...] Local: João Pessoa-PB. Disponível em: <http://www.dsr.inpe.br/sbsr2015/files/p0031.pdf>. Acesso em: 20 nov. 2020.

RISSE, Matthias. **Direitos Humanos e Inteligência Artificial: Uma Agenda Urgentemente Necessária**. 2018. https://www.academia.edu/36988462/_Tradu%C3%A7%C3%A3o_Acesso_em:_12_nov._2020. Acesso em: 12 nov. 2020.

SÁ, Hindenburgo. **UM MÉTODO BASEADO EM INTELIGÊNCIA COMPUTACIONAL PARA A GERAÇÃO AUTOMÁTICA DE CASOS DE TESTE DE CAIXA PRETA**. 2010. Disponível em: https://www.teses.usp.br/teses/disponiveis/3/3141/tde-29112010-153615/publico/Dissertacao_Hindenburgo_Elvas_Goncalves_de_SA.pdf. Acesso em: 30 set. 2020

SELLITTO, Miguel. **INTELIGÊNCIA ARTIFICIAL: UMA APLICAÇÃO EM UMA INDÚSTRIA DE PROCESSO CONTÍNUO**. 2002. Disponível em: https://www.academia.edu/2861183/Intelig%C3%A7%C3%A3o_artificial_uma_aplic%C3%A7%C3%A3o_em_uma_ind%C3%A9stria_de_processo_cont%C3%ADnuo. Acesso em: 22 jun. 2020

SILVA, Mirna. **INTELIGENCIA ARTIFICIAL ADAPTATIVA PARA AJUSTE DINAMICO DE DIFICULDADE EM JOGOS DIGITAIS**. Orientador: Luiz Chaimowicz. Pós-Graduação em Ciência da Computação, Instituto de Ciências Exatas da Universidade Federal, Minas Gerais, 2015. Disponível em: <https://repositorio.ufmg.br/bitstream/1843/ESBF-AARQWM/1/mirnapaula.pdf>. Acesso em 23 jun 2020.

SILVA, Renato. **Inteligência Artificial Aplicada a Ambientes de Engenharia de Software: Uma Visão Geral**. 2005. Disponível em: <http://files.engenharia-de-software7.webnode.com/200000026b805cb9010/Intelig%C3%A7%C3%A3oArtificial%20Aplicada%20a%20Ambientes%20de%20Engenharia%20de%20Software.pdf>. Acesso em: 12 abr. 2020.

SOMMERVILLE, Ian, **ENGENHARIA DE SOFTWARE**. 9º. São Paulo. Pearson Education do Brasil, 2011. P. 548.

WARDLE, Claire, 2017. **Fake news. It's complicated**. Disponível em: <https://firstdraftnews.org/latest/fake-news-complicated/>. Acesso em: 20 nov 2020.

TestComplete Automated UI Testing Tool | SmartBear

Visual Regression Testing (screenster.io)

Endtest: Intelligent Test Automation

Appsurify TestBrain – Make your Testing Intelligent

Mobile App and Website Testing on Real Devices - (sofy.ai)