

UNIVERSIDADE ESTADUAL PAULISTA
“JÚLIO DE MESQUITA FILHO”
FACULDADE DE ARQUITETURA, ARTES E COMUNICAÇÃO-FAAC
MESTRADO EM MÍDIA E TECNOLOGIA-PPGMiT

André Rodrigues de Medeiros

MOVIMENTO MAKER E INTELIGÊNCIA ARTIFICIAL:
desenvolvimento de um protótipo reprodutível e aplicável

BAURU/SP
2021

André Rodrigues de Medeiros

MOVIMENTO MAKER E INTELIGÊNCIA ARTIFICIAL:
desenvolvimento de um protótipo reprodutível e aplicável

Trabalho de Conclusão de Mestrado apresentado ao Programa de Pós-Graduação em Mídia e Tecnologia (PPGMiT), da Faculdade de Arquitetura, Artes e Comunicação, Universidade Estadual Paulista “Júlio de Mesquita Filho”, para obtenção do título de Mestre em Mídia e Tecnologia sob a orientação do Prof. Dr. Dorival Campos Rossi

BAURU/SP

2021

Medeiros, André Rodrigues.

Movimento maker e inteligência artificial: desenvolvimento de um protótipo reprodutível e aplicável/ André Rodrigues de Medeiros, 2021
70 f.

Orientador: Dorival Campos Rossi

Dissertação (Mestrado)-Universidade Estadual Paulista. Faculdade de Arquitetura, Artes e Comunicação, Bauru, 2017

1. Movimento Maker. 2. Inteligência Artificial. 3. Protótipo de software. I. Universidade Estadual Paulista. Faculdade de Arquitetura, Artes e Comunicação. II. Título.



UNIVERSIDADE ESTADUAL PAULISTA

Câmpus de Bauru



ATA DA DEFESA PÚBLICA DA DISSERTAÇÃO DE MESTRADO DE ANDRÉ RODRIGUES DE MEDEIROS, DISCENTE DO PROGRAMA DE PÓS-GRADUAÇÃO EM MÍDIA E TECNOLOGIA, DA FACULDADE DE ARQUITETURA, ARTES E COMUNICAÇÃO - CÂMPUS DE BAURU.

Aos 09 dias do mês de março do ano de 2021, às 14:00 horas, por meio de Videoconferência, realizou-se a defesa de DISSERTAÇÃO DE MESTRADO de ANDRÉ RODRIGUES DE MEDEIROS, intitulada **Movimento Maker e Inteligência Artificial: desenvolvimento de um protótipo reprodutível e aplicável**. A Comissão Examinadora foi constituída pelos seguintes membros: Professor Assistente Doutor DORIVAL CAMPOS ROSSI (Orientador(a) - Participação Virtual) do(a) Departamento de Design da Faculdade de Arquitetura, Artes e Comunicação / Universidade Estadual Paulista, Professor Doutor CLEMILTON LUÍS BASSETTO (Participação Virtual) do(a) Faculdades Integradas de Bauru, Professora Associada VÂNIA CRISTINA PIRES NOGUEIRA VALENTE (Participação Virtual) do(a) Departamento de Artes e Representação Gráfica da Faculdade de Arquitetura, Artes e Comunicação / Universidade Estadual Paulista. Após a exposição pelo mestrando e arguição pelos membros da Comissão Examinadora que participaram do ato, de forma presencial e/ou virtual, o discente recebeu o conceito final: **APROVADO**. Nada mais havendo, foi lavrada a presente ata, que após lida e aprovada, foi assinada pelo(a) Presidente(a) da Comissão Examinadora.

Professor Assistente Doutor DORIVAL CAMPOS ROSSI

"I propose to consider the question, 'Can machine think?' This should begin with definitions of the meaning of the terms 'machine' and 'think'."

- Alan Turing

"O homem erudito é um descobridor de fatos que já existem - mas o homem sábio é um criador de valores que não existem e que ele faz existir."

- Albert Einstein

"No que diz respeito ao empenho, ao compromisso, ao esforço e à dedicação, não existe meio termo. Ou você faz uma coisa bem feita ou não faz."

- Ayrton Senna

AGRADECIMENTOS

A Deus, à minha mãe Marlene, meu pai Francisco e todos os meus avós, por todo amor, carinho e educação.

Ao saudoso e brilhante Prof. Dr. João Fernando Marar (*in memoriam*), por ter inicialmente acolhido minhas ideias e projeto, mas que, porém, muito brevemente nos deixou. Mas deixou também a imensa vontade e o ímpeto de estudar e realizar.

Um grande professor e grande homem!

Ao perspicaz e inovador Prof. Dr. Dorival Campos Rossi, que recebeu meu projeto de braços abertos e com muito carinho, sendo meu orientador e conselheiro durante a pós-graduação, sempre dando-me a liberdade necessária de pesquisar, trabalhar e executar, além de colaborar em todos os cruciais momentos em que precisei (e ainda precisarei!) de todo seu conhecimento e vivência.

À gentil e intuitiva Prof.^a Dr.^a Maria Cristina Gobbi, que desde a especialização na UNESP Bauru, me mostrou o engrandecedor universo da pós-graduação, entusiasticamente me convidando a colocar os pés na pesquisa científica.

À minha querida companheira de todas as horas, e há muitos anos, Daniela Kage e nossas três belas e incríveis filhas: Clara, Lis e Aurora. Em todos os momentos elas foram compreensivas e me impulsionaram na direção dos meus sonhos, que agora são compartilhados entre todos nós, possibilitando que esse trabalho fosse possível graças também ao amor e a união.

Aos amigos da TOTVS Bauru, saudosamente Microsiga e carinhosamente Data Manager (DM). Além de muito conhecimento e oportunidades, proporcionaram-me o incentivo e compreensão necessários para que eu pudesse realizar esse trabalho, mesmo em meio a tantas agendas e projetos.

A todos os professores e ao corpo técnico de pós-graduação da FAAC/UNESP Bauru pela oportunidade, aprendizado e respaldo, sem os quais todo esse trabalho não seria possível e viável.

A todos os empenhados e inteligentes colegas de turma e disciplinas com quem tive o prazer de compartilhar inúmeras horas de reflexões, discussões, desenhos, apresentações, pesquisas e questionamentos dos mais variados, causando-nos o verdadeiro impacto do compartilhar e viver o conhecimento.

Aos meus amigos Dudlei Landim, Juliano Caserta e Pedro Berti, por me ouvirem quando precisei dialogar e externar as mais variadas reflexões, além disso mostrando-me que a ciência pode unir-se com amizade, artes, filosofia, convivência e projetos.

É uma honra poder compartilhar com todos este humilde trabalho.

MEDEIROS, A. R. **MOVIMENTO MAKER E INTELIGÊNCIA ARTIFICIAL: DESENVOLVIMENTO DE UM PROTÓTIPO REPRODUTÍVEL E APLICÁVEL**, 2021, 70f. Trabalho de Conclusão (Mestrado em Mídia e Tecnologia) - FAAC - UNESP, sob a orientação do Prof. Dr. Dorival Campos Rossi, Bauru, 2021.

RESUMO

Neste trabalho explora-se a intersecção entre o movimento *maker* e tecnologias contemporâneas para inteligência artificial. Como notadamente essas novas tecnologias e metodologias tornam-se mais acessíveis e protagonistas no dia a dia de pessoas e empresas, são objetivos desse trabalho buscar analisar algumas dessas tecnologias, encontrar possíveis contributos para a área, além de demonstrar a viabilidade técnica de se construir um protótipo de software computacional de assistência pessoal, reproduzível e aplicável, utilizando-se dos conceitos *maker*, juntamente com o uso destas tecnologias.

Palavras-chave: movimento maker; inteligência artificial; protótipo de software

MEDEIROS, A. R. **MAKER MOVEMENT AND ARTIFICIAL INTELLIGENCE: DEVELOPMENT OF A REPRODUCIBLE AND APPLICABLE PROTOTYPE**, 2021, 70f. Conclusion Work (Master in Media and Technology) - FAAC - UNESP, under the guidance of Prof. Dr. Dorival Campos Rossi, Bauru, 2021.

ABSTRACT

This work pretend to explore the intersection between the maker movement and contemporary technologies to artificial intelligence. These new technologies and methodologies specially become more accessible and the protagonists in the daily lives of people and companies, the aims of this work are to analyze some of these technologies, to find possible contributions to the area, in addition to demonstrate a technical availability of building a prototype of personal assistance computer software, reproducible and applicable, using maker concepts, together with using these technologies.

Keywords: maker movement; artificial intelligence; software prototype

LISTA DE FIGURAS

Figura 1 - Macro Região Centro Oeste Paulista.....	14
Figura 2 - Alexandre, o Grande, e Aristóteles.....	22
Figura 3 - Suposto criador da bitcoin.....	25
Figura 4 - Modelos de rede.....	27
Figura 5 - Comparativo crescimento no uso de Python.....	32
Figura 6 - Site Python Oficial.....	33
Figura 7 - Documentação Python online.....	34
Figura 8 - Pacotes disponíveis em Python.....	35
Figura 9 - Ambiente Anaconda Navigator (visual).....	40
Figura 10 - Ambiente Anaconda Linha de Comando (prompt).....	40
Figura 11 - Editor de texto e códigos Notepad++.....	41
Figura 12 - Macroprocesso Scrum.....	44
Figura 13 - Logotipo Creative Commons no Brasil.....	45
Figura 14 - Selo 1 Creative Commons.....	45
Figura 15 - Selo 2 Creative Commons.....	46
Figura 16 - Selo 3 Creative Commons.....	46
Figura 17 - Selo 4 Creative Commons.....	47
Figura 18 - Selo 5 Creative Commons.....	47
Figura 19 - Selo 6 Creative Commons.....	47
Figura 20 - Site Plataforma Github.....	48
Figura 21 - Captura de tela do protótipo.....	51
Figura 22 - Diagrama básico de execução do protótipo.....	52

LISTA DE TABELAS

Tabela 1. Princípios Ágeis.....	42
---------------------------------	----

SUMÁRIO

1 INTRODUÇÃO.....	13
1.2 Negócios em tecnologia na região centro oeste paulista.....	14
1.2.1 Resumo de perfil empresarial.....	17
2 RELATÓRIO TÉCNICO-CIENTÍFICO.....	19
2.1 Dados Gerais do Projeto.....	19
3 REFERENCIAL TEÓRICO-TÉCNICO.....	21
3.1 Inteligência Artificial.....	21
3.2 Movimento <i>Maker</i>	24
3.2.1 O exemplo do Bitcoin e <i>BlockChain</i>	25
3.3 Conceitos sobre <i>Open Source</i>	28
3.4 Linguagem de Programação Python.....	31
3.4.1 Bibliotecas Python.....	34
3.4.2 Bibliotecas Seleccionadas Neste Trabalho.....	36
3.5 Ambiente de Desenvolvimento Anaconda.....	39
3.6 Aplicativo Notepad++.....	41
3.7 Métodos Ágeis e Scrum.....	42
3.8 Licença Creative Common.....	44
3.9 Plataforma GitHub.....	48
4 MATERIAS E MÉTODOS.....	49
4.1 Tecnologias e recursos utilizados.....	49
4.2 Descrição do Protótipo.....	49
4.3 Aplicação de Software.....	49
4.4 Metodologia utilizada.....	50
5 RESULTADOS OBTIDOS.....	51
6 CONCLUSÕES.....	54
REFERÊNCIAS.....	55
ANEXO 1.....	58
ANEXO 2.....	61

1 INTRODUÇÃO

Nos últimos anos, o crescente interesse e aumento da procura por tecnologias para ciência de dados, sistemas inteligentes e cognitivos (inteligência artificial), e sobretudo a chamada cultura *maker*, tem marcado a formação de profissionais e o posicionamento de diversas empresas por todo o mundo.

Este trabalho pretende analisar essas tecnologias contemporâneas e o movimento *maker*, abordando conceitos e ferramentas que podem ser aplicadas em soluções reais.

Uma vez que experencia-se uma nítida mudança nas formas contemporâneas de produzir conhecimento, gerar negócios e compartilhar informação, a construção de novas ferramentas através de diversos recursos *open source* se torna quase que um imperativo nesse novo contexto de pessoas e empresas, exigindo não só novas abordagens e tecnologias, mas novas competências e habilidades que precisarão ser desenvolvidas de forma adequada e direcionada.

Cabe também, apenas a título de localização, apresentar brevemente o contexto regional onde este trabalho se insere. O Centro-Oeste Paulista é uma grande região do estado de São Paulo, no Brasil, que compreende o entorno das cidades de Araçatuba, Assis, Bauru, Marília, Tupã e Presidente Prudente.

A região destacou-se ao longo do século XIX e até a primeira metade do século XX, ao lado do Nordeste e Noroeste Paulista, como a principal região cafeeira do estado, grão que impulsionou a economia e a colonização local durante o período. Atualmente, a região ainda conta com forte presença do setor agropecuário em sua economia, que está associado a um significativo crescimento do terceiro setor, desde o final do século XX.

Figura 1 – Macro Região Centro Oeste Paulista



Fonte: <http://g1.globo.com/sp/presidente-prudente-regiao/especial-publicitario/unimed-centro-oeste-paulista/noticia/2016/07/interacao-com-comunidade-e-suporte-unimeds-o-que-faz-unimed-centro-oeste-paulista.html>

Ao longo das últimas duas décadas a região também vem se mostrando de grande potencial para empresas de tecnologia, principalmente softwares para negócios e serviços especializados em segmentos distintos de mercado, possuindo hoje grandes *players* nacionais da área, além de diversas pesquisas acadêmicas da UNESP Bauru direcionadas ao tema.

Com o advento da inteligência artificial e a onda de movimento *maker* chegando com força à região, surgiram diversos projetos, novos empreendimentos, bem como iniciativas que visam se apropriar dessas tecnologias e metodologias *open source* para gerar inovação, campos de estudo e novos negócios.

1.2 Negócios em tecnologia na região

As pesquisas sobre o tema tendem a ser complexas uma vez que os processos de desenvolvimento das ferramentas tecnológicas ainda não estão ‘acabados’, fruto da constante evolução técnica e das interações sociais subjacentes. No entanto, queremos entender nesse atual momento de desenvolvimento quais aspectos e impactos estão sendo trazidos para a cena tecnológica, bem como compreender o

como se produzem novas soluções, protótipos ou até mesmo novas empresas e modelos de negócios.

Além disso, como observa Belluzzo (2019), pode-se considerar que

“A partir de uma abordagem do mundo contemporâneo, dos novos paradigmas, da consciência planetária, da globalização, chegar-se-á à era digital, destacando-se os seus impactos sobre a valorização da informação e do conhecimento como bens de valor. Ressalta-se que a produção do conhecimento, na atualidade, é uma questão fundamental, tratando-se de uma teoria voltada à sua própria prática: ao mesmo tempo em que a fundamenta, subordina-se a ela, implicando necessariamente em mudanças no modo de pensar/atuar das pessoas. A educação também é parte desse cenário de mudanças e existem razões para que seja um referencial diferenciado na chamada “sociedade em rede”, sendo uma situação emergente a mudança de postura no que diz respeito à migração da sua identidade como transmissora de informação e de cultura para uma condição de ensinar a aprender e a pensar, preparando pessoas para que prolonguem os benefícios da escola além da escola mesma, tornando funcionais os conhecimentos adquiridos e, sobretudo, para que saibam empregar o poder da inteligência na vida profissional e no seu cotidiano”.

Com isso em mente e olhando a região atual de trabalho, notamos o surgimento não somente de “empresas de software”, mas *start-ups* (iniciativas que se tornam negócios de alta rentabilidade, baseados no empenho individual ou coletivo), projetos com diversas tecnologias contemporâneas, inclusive inteligência artificial, que servem de incubação para novos negócios através de eventos chamados *hackathons* (da junção das palavras inglesas *hacker* e *marathon*), maratonas de programação intensiva, de forma que sejam obtidos produtos, resultados e/ou soluções concretas ao final de cada encontro (MANOVICH, 2005).

É interessante entender como se comportam os profissionais e/ou pesquisadores da área com relação a esse novo cenário, com tecnologias e conhecimentos disponíveis abundantemente em diversos canais e meios? Seria necessária uma constante formação do indivíduo? Uma curadoria de conteúdos relevantes para cada demanda? Multidisciplinariedade no sentido de construir novas pontes?

Mas antes de aprofundar nossa reflexão, cabe definir alguns conceitos, tipos de atividades e negócios, tanto pelo prisma de resultados já obtidos como de planejamentos e possibilidades para novas metas e objetivos.

O vasto mercado de tecnologia impõe que profissionais, pesquisadores e empresas determinem constantemente novas plataformas, novas metodologias e, em

cima disso, tragam resultados tangíveis e concretos para os ambientes em que estão inseridos. No entanto, não há nenhuma ‘receita de bolo’ que funcione ‘tão bem como antes’, pois, no contexto contemporâneo, se ontem o problema foi resolvido por um caminho, hoje existem inúmeros meios de se fazer isso. Qual seria o mais vantajoso para todos os interessados?

Para responder a tais questionamentos selecionamos algumas empresas para exemplificar o cenário atual de negócios da região. Trata-se de empresas com atuação estratégica no estado de São Paulo e em todo Brasil, que utilizaram tecnologias contemporâneas para desenvolver seus projetos, além de possuírem uma atitude notadamente *maker* entre seus pesquisadores e empreendedores, alguns estudantes ou profissionais em início de carreira. Cita-se, a seguir, perfis de negócios multifacetados, que nos dará uma percepção mais completa do panorama do mercado regional. São elas:

Finch: Disrupção tecnológica na área jurídica com inteligência artificial e data science;

Lecom: BPM (Business Process Management, gestão de processos de negócios) e serviços especializados em tecnologia, para redução de retrabalho com ambientes digitais;

NewWay: Softwares e serviços para marketing digital que incluem tecnologias para aprimorar a comunicação entre empresas e indivíduos, usando também novas tecnologias;

TOTVS Bauru: Uma unidade da maior produtora de softwares e serviços segmentados por indústria no Brasil. A TOTVS foi uma das primeiras *start-ups* do Brasil em meados de 1983 e hoje esta presente em todo o mundo, com tecnologias de ponta;

Yank!: Robotização e automatização especializada de processos, reduzindo retrabalhos e erros humanos através de inteligência em processamento de dados e processos.

A seguir traça-se um breve histórico e contextualização de cada um desses ramos de negócios sediados em Bauru, no estado de São Paulo.

1.2.1 Resumo de perfil empresarial

Abaixo um breve resumo dessas empresas, conforme suas próprias definições disponíveis nos seus respectivos sites:

Finch: “A Finch Soluções nasceu em 2013 de nosso potencial disruptivo de revolucionar atividades e processos relacionados ao controle do contencioso de massa (volume de processos) do maior escritório de advocacia do Brasil, o JBM advogados, aumentando sua produtividade e ganhos de eficiência. No início fomos destaque pela implementação de robôs de captura de informação, automação e gestão de processos para negócios no mundo jurídico, criando cases que abriram caminho para novos paradigmas de evolução.” (Disponível em: <http://www.finchsolucoes.com.br/quem-somos>)

Lecom: “Há mais de 20 anos, temos nos especializado em transformar processos analógicos em constantes experiências digitais.” (Disponível em: lecom.com.br)

New Way: “Somos um grupo de pessoas conectadas pelo propósito de transformar o relacionamento digital das empresas com seus clientes e colaboradores, através de tecnologia e humanização. Inovamos para aproximar!” (Disponível em: newway.com.br)

TOTVS: “Fundada em 1983 como Microsiga, uma das primeiras *start-ups* do Brasil, a TOTVS expandiu, evoluiu e multiplicou seus negócios ao longo de sua trajetória. Como uma empresa originalmente brasileira, a TOTVS acredita no ‘Brasil que faz’ e apoia o crescimento de milhares de negócios e empreendedores, de norte a sul do país, por meio da tecnologia. Atua através de unidades próprias e franquias em mais de 40 países. A TOTVS Bauru é a única credenciada para esse portfólio na região.” (Disponível em: totvs.com.br)

Yank!: “A solução definitiva para a robotização de seus processos. Há 8 anos robotizando processos corporativos, a Yank conta com um time de especialistas em RPA (robotic process automation), para automatizar os processos de forma ágil, em diferentes verticais de mercado.” (Disponível em: yanksolutions.com.br)

Com essa breve introdução, pretendemos apresentar alguns conceitos que facilitam a compreensão e o acompanhamento do cenário atual do mercado de

solução em softwares, que usam recursos tecnológicos contemporâneos de inteligência artificial e que têm no movimento *maker* um inoculador da prática real de ideias e projetos.

2 RELATÓRIO TÉCNICO-CIENTÍFICO

2.1 Dados Gerais do Projeto

Nome do Produto:

ÍSYS, Protótipo de Software Maker com Inteligência Artificial Encapsulada para assistência pessoal.

Período de Execução:

4 meses.

Valor total do projeto:

Este projeto não recebeu nenhum tipo de investimento financeiro, somente o apoio através de recursos e infraestrutura oferecidos pela Unesp campus de Bauru e recursos de hardware (notebook) anteriormente adquiridos pelo pesquisador.

Bolsas - Financiamentos – Convênios e Parcerias:

Não houve.

Instituições participantes:

O projeto foi desenvolvido nas dependências da UNESP – Universidade Estadual Paulista “Júlio de Mesquita Filho” campus de Bauru, durante o curso de mestrado do PPGMiT – Programa de Pós-Graduação em Mídia e Tecnologia vinculado a FAAC – Faculdade de Arquitetura, Artes e Comunicação, também foi utilizada a residência do pesquisador em diversos momentos de atividade remota. Para que fosse possível a implementação do protótipo, foi necessária a utilização de recursos de hardware, software e conexão com a internet.

Caracterização da Pesquisa:

Pesquisa Técnico Científica de Base Tecnológica, reprodutível e aplicável.

Caracterização da pesquisa com uma breve justificativa para o enquadramento:

A sociedade e a tecnologia passam por processos naturais de transições. O constante avanço das tecnologias (inteligência artificial, ciência de dados, entre

outras) e sua interconexão com as pessoas tem criado boas oportunidades para desenvolvimento de novas iniciativas através do pensamento *maker*.

Visando explorar na prática as possibilidades de tais avanços, identificou-se a necessidade do desenvolvimento de um protótipo que pudesse ao mesmo tempo representar o momento em que vivemos e contribuir para a atividade acadêmica a partir de um conteúdo *open source* e *open design*.

Pesquisador: André Rodrigues de Medeiros

Orientador: Prof. Dr. Dorival Campos Rossi

Equipamentos adquiridos:

Para o desenvolvimento do projeto foi necessária a utilização de um notebook que já era de propriedade do pesquisador antes do ingresso na pós-graduação. Além disso, se faz necessário acesso à internet.

3 REFERENCIAL TÉCNICO-TEÓRICO

3.1 Inteligência Artificial

O termo “cognição e sistemas inteligentes” talvez descrevesse melhor o campo de estudos que estamos analisando, dentro do qual a chamada Inteligência Artificial (IA) acabou por se tornar mais proeminente e mais popular. Esse campo consiste em uma série de reflexões que buscam reproduzir características da mente humana usando elementos computacionais (LAUREANO, 1999).

Segundo Teixeira (2019), a Inteligência Artificial pode ser definida como a ciência que estuda a emulação do comportamento da inteligência humana por meio de máquinas, tentando compreender de que forma se processa o pensamento humano para entender de modo a poder representá-la por meio de um computador.

Nesse sentido, a mente humana funcionaria “como um computador, e por isso o estudo de programas computacionais é a chave para se compreender alguma coisa acerca de nossas atividades mentais” (TEXEIRA, 2019, p.10).

Com o conhecimento da simulação da mente humana em máquinas é possível, por exemplo, desenvolver *chatbots* (ROTHERMEL E DOMINGUES, 2007) para conversação e interação humano-máquina que podem desempenhar funções específicas e ‘ordens’, ou ainda aplicar algoritmos que simulariam a mente humana tomando decisões nos mais variados níveis, dependendo única e exclusivamente de volumes de dados e cálculo de padrões matemáticos (NEVES E BARROS, 2005).

Se nos remetermos às origens da computação podemos reconhecer tais ideias e conceitos:

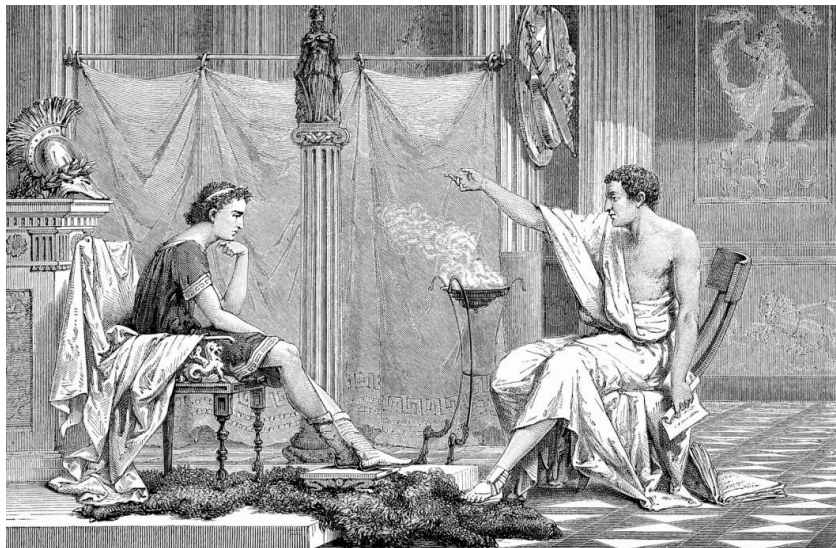
(...) Como Ada Lovelace, Alan Turing era um programador, voltava seu olhar para dentro e observava os passos seguidos pela lógica de sua mente. Ele imaginava a si mesmo como um computador. Destilou seus processos mentais nas mínimas partes que os constituíam, os átomos do processamento da informação. Alan Turing e Claude Shannon tinham códigos em comum. Turing codificava instruções sob a forma de números. Ele codificou os números decimais em zeros e uns. Shannon criou códigos para os genes e cromossomos e relés e interruptores. Ambos aplicaram sua engenhosidade ao mapeamento de um conjunto de objetos em outro conjunto: operações lógicas e circuitos elétricos; funções algébricas e instruções para máquinas. O uso de símbolos e a ideia de *mapear*, no sentido de encontrar uma correspondência rigorosa entre dois conjuntos, possuíam um lugar de destaque no arsenal mental de ambos. Esse tipo de codificação não tinha como objetivo obscurecer, e sim iluminar, descobrir se maçãs e laranjas eram afinal todas equivalentes ou, se não equivalentes, ao menos fungíveis. A guerra trouxe os dois ao terreno da criptografia em suas formas mais indecifráveis.” (GLEICK, 2013, p.95)

Segundo Marar (2016), a ciência da computação é uma área de estudo recente, mas não a inteligência artificial:

"Ela vem lá dos filósofos gregos, de Platão, de Aristóteles. É muito interessante observar esses estudiosos do passado. Eles vão remontando e construindo o que nós entendemos hoje como um modelo científico (...) Não que já existissem sistemas que faziam coisas por si próprios, mas a ideia de uma inteligência não humana que pensasse por si própria já era idealizada".

Ele diz ainda que Aristóteles, filósofo e professor de Alexandre, o Grande, rei da Macedônia (na Grécia Antiga), pensava em como livrar os escravos dos seus afazeres, automatizando diversas tarefas.

Figura 2 – Alexandre, o Grande, e Aristóteles



Fonte: <https://tecnoblog.net/195106/inteligencia-artificial-historia-dilemas/>

De acordo com Medeiros (2015), em sua dissertação de mestrado sobre inferência de emoções em textos, podemos constatar que existem quatro definições de Inteligência Artificial:

- (a) sistemas que pensam como humanos;
- (b) sistemas que agem como humanos;
- (c) sistemas que pensam logicamente;
- (d) sistemas que agem logicamente.

Segundo Pereira (1988), as duas primeiras categorias, a e b, são essencialmente empíricas e envolvem formulação de hipóteses e confirmação experimental. Já as outras duas, c e d, são teóricas e envolvem matemática e engenharia. O autor defende que o pensar, de acordo com a definição clássica de Turing, seria uma “imitação” do comportamento humano. Essa seria uma visão essencialmente antropocentrista, pois exige da máquina pensante um padrão de pensamento igual ao dos seres humanos ao exigir que para a máquina “pensar como humano” ela deveria imitá-lo.

O teste de Turing, segundo Whitby (1996), foi projetado para fornecer uma definição operacional satisfatória de inteligência. O computador passa no teste se um interrogador humano, depois de propor algumas perguntas por escrito, não conseguir descobrir se as respostas escritas vêm de uma pessoa ou de um computador (RUSSEL; NORWING, 1995).

Apesar disso, a definição pura da inteligência de Turing ainda se encontra com a descrita pela literatura de psicologia, ao defender que pensar seria a capacidade de processar informações ao ponto de resolver problemas. Russel e Norwig (1995) defendiam a ideia de que haveria três formas de se determinar como os humanos pensam, penetrando nos componentes reais da mente humana.

Isso se daria por meio da introspecção, procurando captar os nossos pensamentos de forma particular, através de experimentos psicológicos e de sua observação através de imagens cerebrais, observando o cérebro em funcionamento. Turing é mais incisivo ainda, ao propor que não seria possível realizar uma imitação do sistema nervoso, mas tão somente do comportamento do ser pensante.

Nessa época, já existiam várias teorias de complexidade, simulação de linguagem, redes neurais e máquinas de aprendizagem. Turing imaginou utilizar o termo “inteligência artificial” para esses “sistemas de imaginação humana que usam a ciência da computação”.

Talvez, no presente momento, com o avanço da computação e a redução constante no custo de processamento e armazenamento de dados, seja possível vislumbrar essas “antigas” ideias se materializando em verdadeiras soluções para o mundo real, como algumas empresas e universidades têm apresentado ao longo das últimas décadas, conforme mencionado na introdução deste trabalho. Soluções estas que impactam, de fato, o mundo concreto, gerando novos negócios, novos profissionais e novas soluções sociais.

3.2 Movimento Maker

“Fazer com as próprias mãos” se tornou um novo mantra contemporâneo. Dentro desse conceito surge uma noção de reunir amigos, clientes, parceiros, colaboradores, de modo presencial ou remotamente, para cruzar as habilidades e multidisciplinariedade de todos visando obter algo novo no final.

Os *makers* surgiram nos Estados Unidos há alguns anos e ganharam força recentemente. Hoje, já existem várias publicações dedicadas aos interessados nesse tipo de empreendedorismo e seus seguidores se encontram anualmente em eventos para discutir as novidades do mercado. A *Maker Faire*, que acontece na Califórnia, é a principal feira que propaga a ideia do *Do It Yourself* (faça você mesmo).

Um novo produto, uma nova teoria, alguma plataforma, enfim, o objetivo é criar e co-criar novas ideias usando todos e quaisquer recursos disponíveis, principalmente tecnológicos e de relacionamentos humanos.

A versão atual do DIY teve seu início na primeira década do século XX. Norte americanos defensores do movimento de *Art and Crafts* promoveram o interesse pelo estilo simples da arquitetura doméstica, despertando o interesse do mercado produzindo magazines a respeito de mecânica popular e ciência popular, além de publicar artigos sobre como fazer coisas e encorajar os donos de casa a empreender algumas de suas remodelações (ROSSI, et al, 2016).

Dentro desse modo de pensar o mundo surgiram milhares de iniciativas visando construir novos conceitos e trazer à tona novas ideias, como que se realmente nossa geração buscasse “minerar” a mente humana, utilizando apoios diversos para obter um verdadeiro diamante do espírito deste tempo: a inovação.

Anderson (2012, p. 29) resumiu a história tecnológica das duas últimas décadas em duas sentenças: “nos últimos dez anos descobriu-se novos meios para as pessoas criarem, inventarem e trabalharem juntas na internet. Nos próximos dez anos esses conhecimentos serão aplicados no mundo real”.

Além disso, o autor lembra que mudanças profundas em determinados setores da sociedade terminam por desencadear novas mudanças em todo o tecido social:

“O que propiciou uma mudança cultural - um fascínio com novos protótipos digitais e um desejo de estender o fenômeno online no mundo real - está agora iniciando uma mudança econômica também. O movimento maker está começando a mudar a cara da indústria, na medida em que novos instintos empreendedores vão aparecendo e atividades que eram de lazer se tornam pequenas companhias.” (ANDERSON, 2012, p. 31)

Para entendermos essas considerações é necessário avaliarmos o espaço virtual como um vetor de novas possibilidades que encontra em seu próprio meio alternativas de reaproveitamento e atualização de ideias que geram produtos de inovação nunca antes pensados. Segundo Lévy (1996), virtual não é o contrário de real, mas sim tudo aquilo que tem *potencialidade* para se concretizar. As criptomoedas como a bitcoin são produtos advindos desta hibridização que tem ação real e concreta em promover cada vez mais o pensar global e o agir local¹.

3.2.1 O exemplo do Bitcoin e Blockchain

Em um pequeno, mas contundente artigo de apenas nove páginas, Nakamoto (2020), criou um conceito e o comprovou matematicamente e computacionalmente.

No texto original ele define em poucas páginas como um encadeamento de códigos com criptografia poderiam garantir a mesma confiabilidade que uma moeda real possui ao ser intermediada por governos e bancos, que para todos os efeitos são os garantidores de que uma moeda valha algo.

Figura 3 – Suposto criador da bitcoin



Fonte: <https://foxbit.com.br/blog/qual-o-motivo-de-satoshi-nakamoto-nao-movimentar-seus-bitcoins%EF%BB%BF-noticias-da-semana/>

Existe uma certa ‘lenda’ em torno do autor(a) do artigo original, pois Nakamoto até o momento ‘não existe’. Isso porque ninguém se apresentou como autor desta tecnologia, permanecendo em anonimato. Essa situação abre margem para diversas especulações, como a de que a bitcoin pode ter sido concebida não por uma, mas por

¹ Deve-se esta formulação ao sociólogo alemão Ulrich (1999), que se dispôs a refletir sobre o fenômeno da globalização.

diversas pessoas que escolheram o pseudônimo “Satoshi Nakamoto”. A algum tempo surgiram imagens do homem acima que afirmavam ser a do criador do bitcoin, mas ele negou veementemente.

Além disso, como criador ou criadores da bitcoin, hoje ele(s) possui(em) uma “cota”, ou melhor, uma verdadeira fortuna em bitcoins, um número que gira em torno de alguns bilhões de dólares. Dinheiro este que permanece intocado, rendendo lucros e muitas especulações, embora o mercado atual já utilize o bitcoin em grande escala.

O fato é que a bitcoin não depende de intermediários, nem de um banco, por isso resulta em tarifas quase zero. Por ser uma rede *peer-to-peer* (de ponto a ponto) compartilhável e altamente criptografada, recebe o trabalho de programadores conhecidos como “mineradores”. Usuários interessados se fazem responsáveis em quebrar a informação da bitcoin para entregar a moeda como ela é possível de ser utilizada. Nota-se que são diversos mineradores unidos que conseguem realizar esse trabalho em troca de receberem uma parte do objeto que mais se valoriza nos últimos anos. Essa analogia se assemelha aos ‘mineradores de ouro’, por isso o nome, e eles são recompensados pelas suas atividades de “minerar” a moeda na rede².

O funcionamento da bitcoin é feito por meio da chamada *blockchain*. A definição mais básica de *blockchain* é um *ledger* (registro) digitalizado e compartilhado que não pode ser alterado uma vez que uma transação tenha sido registrada e verificada. Todas as partes da transação, bem como um número significativo de terceiros mantêm uma cópia do *ledger* do *blockchain*, o que significa que seria praticamente impossível alterar todas as cópias do livro de registro globalmente para falsificar alguma transação (LYRA E MEIRIÑO, 2019).

O sucesso da bitcoin desencadeou o estabelecimento de quase mil novas criptomoedas, levando à ilusão de que a única aplicação da tecnologia *blockchain* é para a criação de criptomoedas. No entanto, a tecnologia *blockchain* é capaz de muito mais do que apenas criar criptomoedas e pode suportar transações como as que exigem identificação pessoal, revisão por pares, eleições e outros tipos de decisões democráticas e trilhas de auditoria.³

À parte das lendas urbanas, hoje existem dezenas, quiçá centenas de criptomoedas, versões alternativas e ampliadas, todas nascidas de iniciativas

² Confira em: <http://comunicacaoadoro.wixsite.com/adoro-robotica/single-post/2018/08/03/Bitcoin-Moeda-livre-de-fronteiras>

³ Confira em: <https://br.cointelegraph.com/tags/blockchain>

notadamente *makers*, sempre com muita pesquisa e muito “*hands-on*”, termo este que também vem sendo amplamente utilizado e que corresponderia a colocar a “mão na massa”, unindo pessoas com habilidades multidisciplinares e recursos tecnológicos para a construção de algo novo.

Figura 4 – Modelos de rede



Fonte: Sênior Softwares

Como podemos visualizar na imagem acima, dentre os vários modelos, o bitcoin envolve uma rede descentralizada. Cada integrante de rede é conhecido como um “nó”. Ou seja, se você tem um computador preparado com um software para realizar transações com bitcoins, essa máquina passa a ser um nó nessa rede, que só tende a crescer.

Esse software para transações de bitcoins mantém e ajuda a distribuir cópias atualizadas em tempo real da “cadeia de blocos”. Cada vez que transações são realizadas (blocos são adicionados), todos os “nós” são comunicados para que possam atualizar os registros com as novas informações em toda a rede. A informação se propaga para que todos se “auto-validem”.

O mecanismo de comunicação sempre é criptografado, com métodos de chaves privadas e públicas, e isso impede que a rede seja invadida ou que “nós” sejam adicionados de forma imprópria ou até ilegal.

Portanto, podemos considerar que os “nós” participam da validação das transações. Além disso, uma série de algoritmos são aplicados para verificar se há consenso sobre a realização de determinada transação.

Assim, se você está recebendo bitcoins de alguém como forma de pagamento, um processo de verificação junto aos “nós” confirmará a validade e consenso dessa operação.

Essa transação será então registrada na “grande rede” da *blockchain*. Como esse registro não pode ser apagado, em tese, ninguém consegue ‘subtrair’ esses bitcoins de você fazendo alterações nos blocos, pois isso geraria uma divergência entre as validações de todos os “nós”. Como consequência dessa lógica das criptomoedas, não haveria mais necessidade de “entidades garantidoras”, como bancos e governos.

Diante disso, torna-se interessante destacar neste trabalho essa iniciativa *puramente maker* e tecnológica de uma pessoa ou um grupo delas para, com ‘as próprias mãos’, criar a sua própria criptomoeda digital, impactando o mundo real com ideias trazidas à realização prática, reunindo o que há de disponível no presente momento. Sendo assim, diante de tais conquistas, cabe refletir: quantos outros feitos mais podem ser empreendidos nessa e em outras direções? Para isso será necessário que todos possam participar e colaborar no desenvolvimento dos softwares. Será necessário um código aberto.

3.3 Conceitos sobre *Open Source*

Um bom resumo do conceito de *open source* foi dado pelo site CanalTech (2020), que o define da seguinte forma:

“*Open source* é um termo em inglês que significa código aberto. Isso diz respeito ao código-fonte de um software, que pode ser adaptado para diferentes fins. O termo foi criado pela OSI (*Open Source Initiative*) que o utiliza sob um ponto de vista essencialmente técnico.”

O termo *open source*, bem como o seu ideal foram desenvolvidos por Eric Raymond e outros fundadores da OSI com a finalidade de apresentar um “software livre a empresas de uma maneira mais comercial”, evitando discursos éticos e de direitos.

A nomenclatura “*Open Source*” apareceu durante uma reunião ocorrida em fevereiro de 1998, em um debate que envolveu personalidades que posteriormente se tornariam referências no assunto: Todd Anderson, Chris Peterson, Larry Augustin, Jon “Maddog”, Sam Ockman e Eric Raymond.

A sigla FLOSS, que em inglês significa “*Free/Libre and Open Source Software*”, é uma forma agregadora de utilizar os conceitos de Software Livre e Código Aberto a

favor dos mesmos softwares, visto que ambos se diferenciam apenas na argumentação, como já mencionado.

Os desenvolvedores e defensores do conceito *Open Source* afirmam não se tratar de um movimento “anticapitalista”, mas sim de uma alternativa para o mercado de indústria de software e tecnologias. Esse modelo colaborativo presente no *open source* levou o direito do autor a ser olhado por um outro prisma, concatenando-se com o movimento *maker* que, em essência, é baseado em recursos já disponíveis e em colaboração.

A criação da *Open Source Development Lab* (OSDL) é um exemplo dos grandes esforços realizados por várias empresas já consolidadas, como IBM, Dell, Intel e HP, para trabalhar com a criação de tecnologias de código aberto, mesmo tendo vínculos societários e obrigações determinadas de obtenção de lucros e dividendos, o que denota a importância do conceito.

A OSI (*Open Source Initiative*) propõe dez pontos importantes para que um software possa ser considerado *Open Source*:

1 - Distribuição livre

A licença do programa não deve de nenhuma forma restringir o acesso gratuito por meio de venda ou mesmo de permutas. Nesta dissertação utiliza-se a licença *Creative Commons* que será abordada em respectivo tópico.

2 - Código fonte

De fundamental importância, o software deve conter um código fonte que deve permitir a distribuição também na forma compilada. Caso o programa não seja distribuído com seu código fonte, o desenvolvedor deve fornecer um meio para se obter o mesmo. O código fonte deve ser legível e inteligível para qualquer programador. Nesta dissertação possibilita-se todo e qualquer acesso.

3 - Trabalhos derivados

A licença do software deve fornecer permissão para que modificações sejam realizadas, bem como trabalhos derivados. Também deve permitir que sejam distribuídos, mesmo após modificação, sobre os mesmos termos da licença original.

4- Integridade do autor do código fonte

A licença deve, de maneira clara e explícita, permitir a distribuição do programa construído por meio do código fonte modificado. No entanto, a licença pode requerer que programas derivados tenham um nome ou número de versão distintos do programa original. Isso dependerá da preferência do desenvolvedor do código.

5 - Não discriminação contra pessoas ou grupos

A licença deve estar disponível para qualquer grupo de pessoas e qualquer indivíduo.

6 - Não discriminação contra áreas de atuação

A licença deve permitir que qualquer pessoa de qualquer ramo específico possa fazer a utilização do programa. Ela não deve impedir, por exemplo, que uma empresa faça uso de seu código.

7 - Distribuição da Licença

Os direitos associados ao software devem ser aplicáveis para todos aqueles cujo programa é redistribuído, sem que exista a necessidade da execução de uma nova licença, ou licença adicional para estas partes.

8 - Licença não específica a um produto

O programa pode fazer parte de um outro software, sendo que para utilizá-lo é obrigatório que seja distribuído todo o programa. Se o programa é extraído dessa distribuição, é necessário assegurar que todas as partes sejam disponibilizadas e redistribuídas para todos, visto que todos possuem os mesmos direitos que aqueles que são garantidos em conjunção com a distribuição do programa original.

9 - Licença não restrinja outros programas

A licença não pode ser considerada *open source* se colocar restrições em outros programas que são distribuídos juntos com o programa licenciado.

10 - Licença neutra em relação à tecnologia

A licença deve permitir que sejam adotados estilos e tecnologias sem restrições. Isso quer dizer que nenhuma cláusula da licença pode estabelecer regras para que estes quesitos mencionados sejam aplicados ao programa.

A noção de *open source* trouxe um modo bastante interessante de se pensar o desenvolvimento da tecnologia digital e constitui, de certa maneira, a essência deste trabalho. Um dos exemplos de maior sucesso de programação *open source* é a linguagem Python.

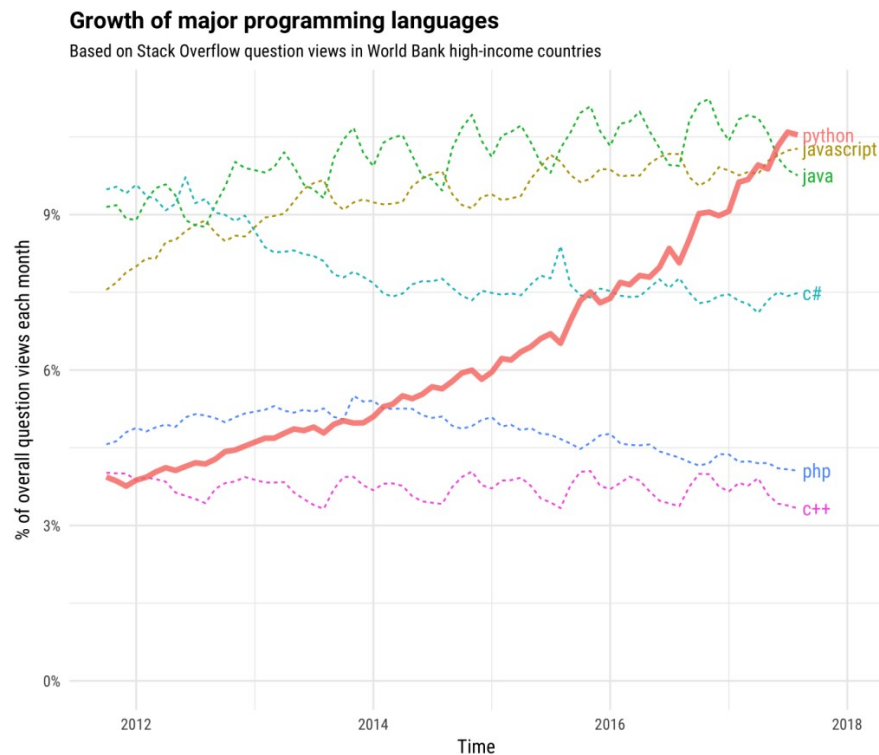
3.4 Linguagem de Programação Python

As linguagens de programação vêm evoluindo de forma bastante acelerada nas últimas décadas, inclusive acompanhando a evolução do hardware e das pessoas que se dedicam a esta atividade.

Cabe lembrar que, atualmente, profissionais de outras áreas, além de ciências exatas, vem se interessando pela programação e se envolvido com tal conteúdo, de forma a encontrar e viabilizar aplicações voltadas especificamente para seus ramos de estudo.

Nesse contexto a linguagem de programação Python tem se mostrado de “fácil acesso” para o público em geral, sendo vastamente já utilizada no meio acadêmico e em pesquisas de empresas. Agora, acompanhamos um aumento vertiginoso da sua utilização por profissionais das mais diversas áreas, especialmente nos últimos oito anos, conforme pode-se notar no gráfico abaixo que apresenta um breve comparativo de algumas linguagens de programação contemporâneas que se destacam no ambiente de desenvolvimento de softwares:

Figura 5 – Comparativo crescimento no uso de Python



Fonte: Data Sciency Academy

A linguagem de Programação Python foi originalmente desenvolvida por Guido Van Rossum, na década de 80. A proposição de Python considerava o interesse explícito dos programadores em diminuir os esforços demandados pela própria linguagem, tais como àqueles relativos ao sistema de tipos e a marcação de final de comando (com ponto e vírgula, por exemplo), tornando o tempo dedicado à programação mais eficiente (ROSSUM E DRAKE, 2012).

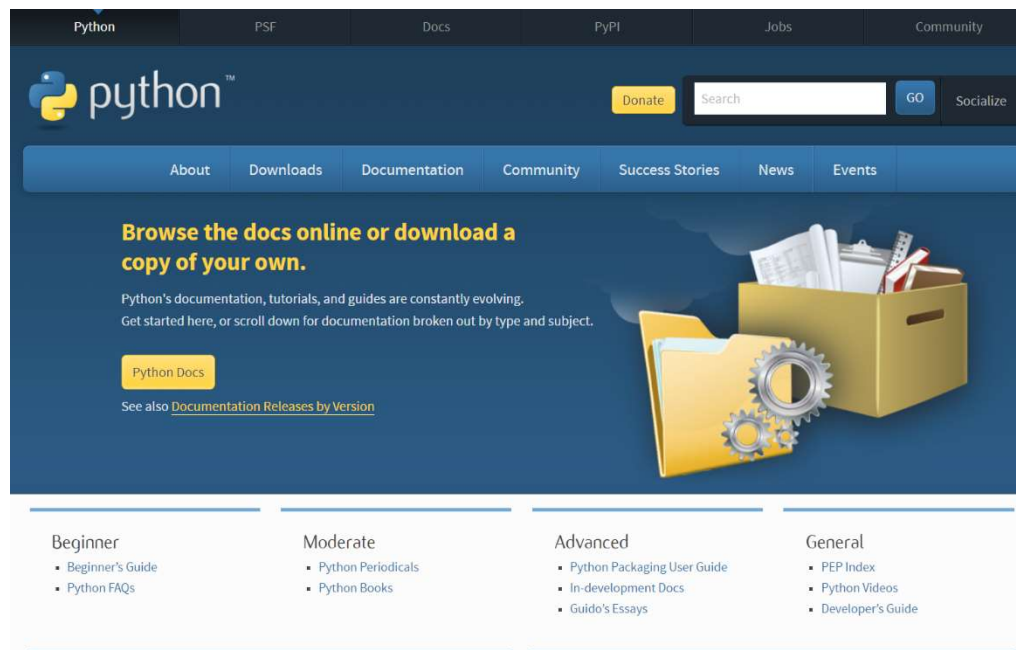
Em meados dos anos 2000, essa linguagem de programação torna-se *open source*, visando a participação ativa e voluntária de diversas pessoas com a finalidade de contribuir para o ecossistema da linguagem, tornando-a constantemente evoluída. Além disso, com a massiva quantidade de dados disponíveis nas mais variadas formas e plataformas, além da proliferação de tecnologias voltadas para Internet, as micro, pequenas e médias empresas (PMEs), apesar de suas restrições de tamanho, receita e pequeno número de funcionários, podem também se beneficiar desta vasta informação disponível e gerar valor para si próprias.

Importante também lembrar que Python conta com uma grande documentação disponível em vários idiomas, além de inúmeros fóruns de discussão onde toda

pessoa ao redor do mundo pode colaborar com exemplos, dúvidas, dicas, melhorias e outras demandas.

A título de exemplo sobre a documentação referida, segue abaixo uma captura de tela do site oficial em sua versão em inglês:

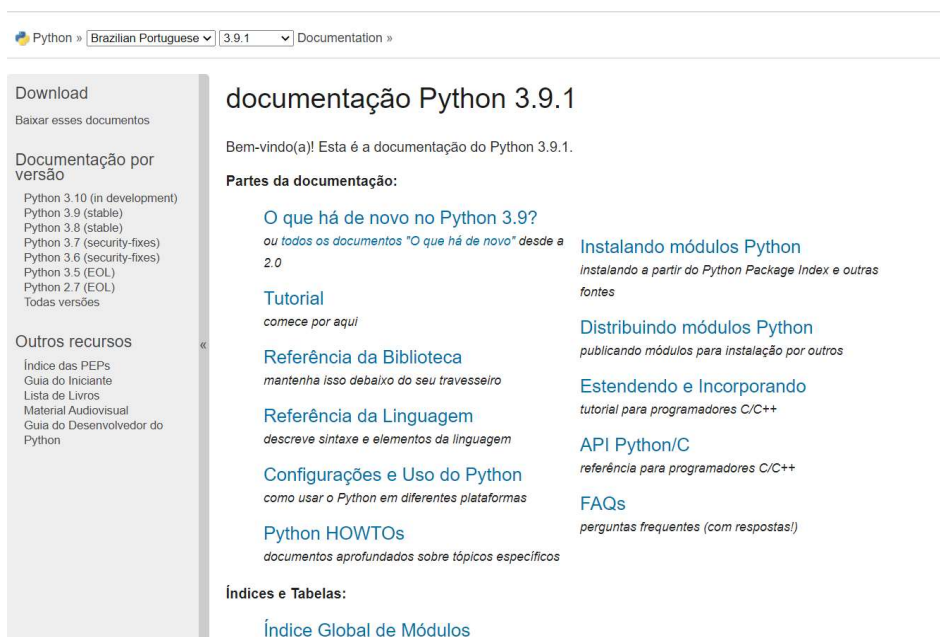
Figura 6 – Site Python Oficial



Fonte: <https://www.python.org/doc/>

Também na versão em português brasileiro encontraremos vasto material, sendo a comunidade brasileira uma das mais movimentadas do mundo:

Figura 7 – Documentação Python online



Fonte: <https://docs.python.org/pt-br/3/>

Com todos esses atributos, a linguagem em Python demonstrou-se ser a mais adequada para a realização deste trabalho de dissertação, possibilitando colocar em prática ideias sobre tecnologia, movimento *maker* e *open source* colaborativo, com o objetivo de obter um impacto concreto no ecossistema em que está inserido.

3.4.1 Bibliotecas Python

Além da facilidade de instalação, aprendizado e utilização, Python oferece também milhares de bibliotecas prontas (pacotes adicionais), com recursos já pesquisados por grandes universidades, empresas e pesquisadores independentes, que contribuem com a comunidade de desenvolvimento.

De maneira bastante simples é possível adicionar esses “pacotes” ao Python e usufruir de novos recursos e comandos que simplificam a vida do profissional e/ou pesquisador. Dessa forma, podemos, por exemplo, usar uma biblioteca de inteligência artificial, sem nem ao menos programar nenhum código ou algoritmo de *machine learning* (aprendizado de máquina).

Isso é possível porque a linguagem consegue encapsular tais recursos em camadas, permitindo chamadas mais simples de determinados comandos, facilitando imensamente sua utilização e produtividade, sendo o desenvolvedor responsável por consumir tais recursos gradativamente conforme suas demandas (SGANDERLA,

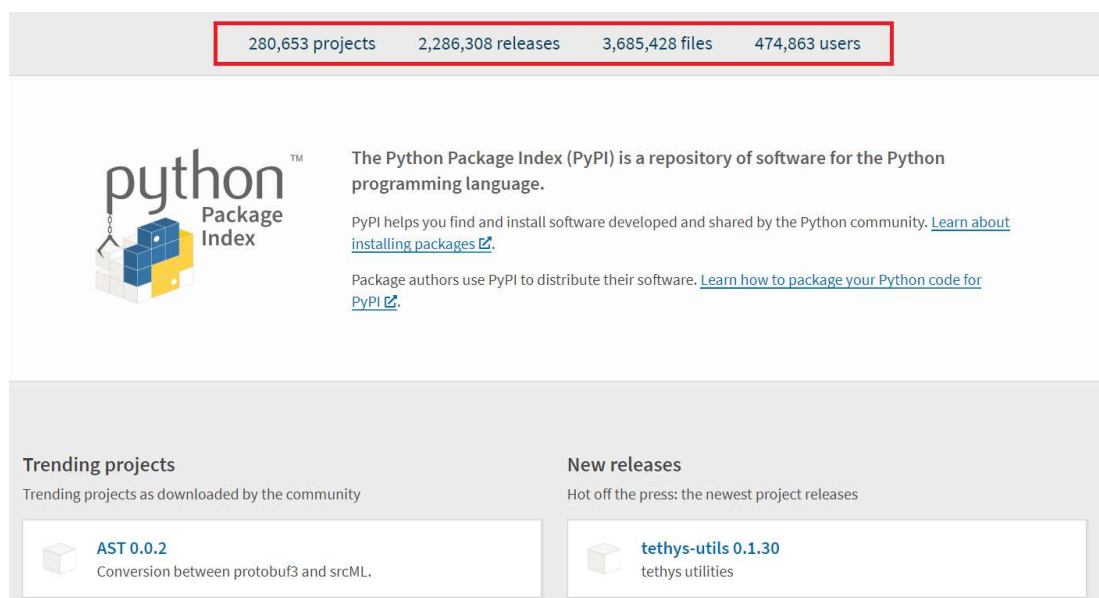
FERRARI, GEYER, 2003). É o caso, por exemplo, da biblioteca “*ChatterBot*”, que utilizamos neste trabalho de dissertação e que será abaixo mais detalhadamente descrita, assim como as demais bibliotecas selecionadas e utilizadas nesta pesquisa (TEIXEIRA ET. AL., 2005).

Para tanto, há alguns anos foi criada a ferramenta *PyPi* (*Python Package Index*, ou em tradução livre “Índice de pacotes Python”), onde todos os interessados podem incluir e fazer *downloads* de novos recursos. Pode-se usá-la para instalar pacotes, programas ou simplesmente códigos, feitos pela comunidade e armazenados como pacotes no PyPi:

O PyPI é, de fato, o maior repositório de código Python que temos hoje em dia. Nele, temos desde códigos amplamente usados, como o *numpy*, até códigos bem menores e menos conhecidos, como o *cidadaobr*, um gerador de dados de um brasileiro que pode ser ampliado e melhorado. (ALURA CURSOS, 2020)

Observe assinalado em vermelho na captura de tela abaixo, a quantidade de projetos, atualizações, arquivos e usuários disponíveis:

Figura 8 – Pacotes disponíveis em Python



Fonte: <https://pypi.org/>

É possível realizar diversas buscas por temas e assuntos relevantes que proporcionarão resultados com as bibliotecas mais adequadas para o projeto, conforme as suas demandas.

Neste trabalho utilizamos algumas bibliotecas que encapsulam recursos básicos e avançados, abaixo elencadas. Vale lembrar que os procedimentos necessários para a reprodução dos resultados obtidos, por qualquer profissional ou pesquisador interessado no assunto, estão disponíveis nos anexos 1 e 2.

3.4.2 Bibliotecas Selecionadas Neste Trabalho

Como o foco do trabalho é a intersecção entre movimento *maker* e recursos tecnológicos relacionados à inteligência artificial contemporânea, pesquisou-se quais pacotes (bibliotecas ou *packages*) estariam mais adequadamente prontos, a fim de reduzir o esforço necessário para desenvolvimento, execução e testes da aplicação, garantindo seu funcionamento e reprodutibilidade, além de demonstrar que o “mão na massa” (*maker*) funcionaria em conjunto com tais tecnologias em nosso ambiente regional.

As bibliotecas selecionadas estão abaixo relacionadas com uma breve descrição de seus nomes, objetivos e funções disponíveis, visando com isso um maior entendimento dos recursos que o protótipo irá apresentar em sua execução preliminar, além de constarem os *links* diretos para acesso às suas documentações atuais, lembrando que a comunidade Python é ‘viva’ e muito veloz, atualizando-se constantemente em suas estruturas, pacotes, dados e documentação:

NumPy: Muito disseminada no meio acadêmico devido aos seus diversos recursos relacionados a cálculos, expressões algébricas e matemática em geral. Toda documentação atual está disponível em <https://numpy.org/doc/> com acesso realizado em 30/12/2020.

PyWinAuto: Trata-se de um conjunto de módulos Python para automatizar a interação do usuário com o sistema operacional Microsoft Windows. Em sua forma mais simples, ele permite que você envie ações do mouse e do teclado para os diálogos e controles do Windows, diretamente de um programa Python em execução. Toda documentação atual está disponível em <https://pywinauto.readthedocs.io/en/latest/> com acesso realizado em 30/12/2020.

Pyttssx3: Biblioteca que inclui *drivers* (formas e mecanismos de acesso a hardware) para os sintetizadores de um “formato texto” para “formato voz”. Existem vários *drivers* já validados. Toda documentação atual está disponível em <https://pyttssx3.readthedocs.io/en/latest/> com acesso realizado em 30/12/2020.

SpeechRecognition: O “reconhecimento de fala” é o processo de conversão de palavras faladas em texto. Essa biblioteca Python oferece suporte a muitos mecanismos de reconhecimento de voz, incluindo Google Speech Engine, Google Cloud Speech API, Microsoft Bing Voice Recognition e IBM Speech to Text. Toda documentação atual está disponível em <https://pythonprogramminglanguage.com/speech-recognition/> com acesso realizado em 30/12/2020.

ChatterBot: ChatterBot é uma biblioteca Python que facilita a geração de respostas automatizadas para a entrada de um usuário. O ChatterBot usa uma seleção de algoritmos de *machine learning* (aprendizado de máquina) para produzir diferentes tipos de respostas, conforme as entradas recebidas. Isso torna mais fácil para os desenvolvedores criar *bots* (robôs de conversação) e automatizar conversas com os usuários. Toda documentação atual está disponível em <https://chatterbot.readthedocs.io/en/stable/> com acesso realizado em 30/12/2020.

SpaCy: Uma biblioteca Python para Processamento de Linguagem Natural (PLN), que facilita a utilização de inteligência artificial para compreensão de diálogos em linguagem humana. Permite a utilização de várias línguas e formatos de análise. Toda documentação atual está disponível em <https://spacy.io/api/doc> com acesso realizado em 30/12/2020.

PyAuto: O PyAutoGUI permite que seus scripts Python controlem o mouse e o teclado para automatizar as interações com outros aplicativos. A biblioteca foi projetada para

ser bastante simples. O PyAutoGUI pode funcionar em sistemas operacionais Windows, macOS e Linux. Toda documentação atual está disponível em <https://pyautogui.readthedocs.io/en/latest/> com acesso realizado em 30/12/2020.

PyAudio: PyAudio fornece vínculos Python para portas de áudio, portanto a biblioteca trata de entradas e saídas de áudios em dispositivos. Com PyAudio, é possível usar Python facilmente para reproduzir e gravar áudio em uma variedade de plataformas. Toda documentação atual está disponível em <https://people.csail.mit.edu/hubert/pyaudio/docs/> com acesso realizado em 30/12/2020.

PocketSphinx: Um mecanismo de reconhecimento de voz que permite trabalhar com modelos prontos e criar os próprios modelos conforme sua necessidade. Toda documentação atual está disponível em <https://cmusphinx.github.io/doc/pocketsphinx> com acesso realizado em 30/12/2020.

OpenCV-Python: OpenCV (*Open Source Computer Vision Library*: <http://opencv.org>) é uma biblioteca de código aberto que inclui várias centenas de algoritmos de visão por computador ou visão computacional. Proporcionando manipular dados obtidos por câmera e realizar algoritmos de detecção de informações de maneira visual. É opcional neste trabalho, mas foi mantida devido a sua relevância e utilidade. Toda documentação atual está disponível em <https://docs.opencv.org/master/d1/dfb/intro.html> com acesso realizado em 30/12/2020.

Spyder: Trata-se um robusto ambiente científico escrito em Python, para Python e projetado por e para cientistas, engenheiros e analistas de dados. Ele apresenta uma combinação única de edição avançada, análise, depuração e funcionalidade de criação de perfil de uma ferramenta de desenvolvimento abrangente com a exploração de dados, execução interativa, inspeção profunda e capacidades de visualização gráfica. Também é opcional neste trabalho, mas consta devido sua importância e

reconhecimento. Toda documentação atual está disponível em <https://docs.spyder-ide.org/current/index.html> com acesso realizado em 30/12/2020.

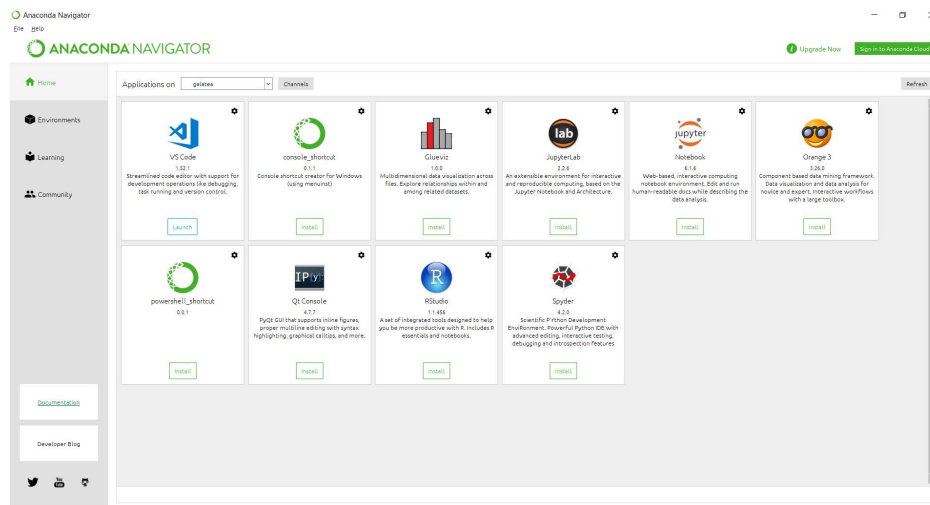
Até o momento de desenvolvimento deste trabalho foram utilizadas apenas as bibliotecas supracitadas capazes de potencializar enormemente as capacidades de desenvolvimento do profissional e/ou pesquisador. Também pode-se observar que todas as bibliotecas apresentadas possuem um bom volume de documentação, conforme *links* de suas descrições, além de fóruns online não oficiais, aqui não relatados, mas facilmente pesquisáveis, que permitem ainda mais colaboração entre todos os interessados ao redor do mundo.

3.5 Ambiente Desenvolvimento Anaconda

Ambientes de desenvolvimento se caracterizam por incorporarem ferramentas de edição e produtividade que podem aumentar ainda mais a efetividade de utilização, produção e análise de um software, proporcionando ao profissional e/ou pesquisador ter sua própria “área de trabalho” direcionada e/ou personalizada para cada projeto em que estiver envolvido. Não bastam apenas desenvolvimentos constantes, mas é de suma importância manter as manutenções tanto do código fonte em si, como do ambiente no qual o profissional está trabalhando.

Entre esses ambientes, o Anaconda é bastante conhecido no mercado e robusto em sua estrutura, proporcionando uma versão gratuita para download em formato *Open Source*. Abaixo, apresentamos o ambiente padrão do Anaconda, a versão visual *Navigator* e a versão de linha de comando respectivamente:

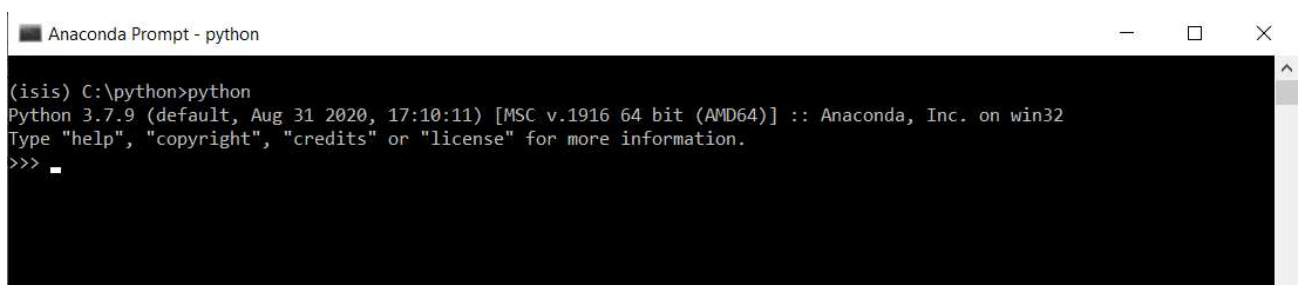
Figura 9 – Ambiente Anaconda Navigator (visual)



Fonte: Captura de tela do próprio pesquisador

Podemos notar na Figura 10 que trabalhamos nesta dissertação com um ambiente Anaconda para Python Versão 3.7.9 e MSC Versão 1916, sistema operacional Windows 64 bits

Figura 10 – Ambiente Anaconda Linha de Comando (prompt)



Fonte: Captura de tela do próprio pesquisador

É importante constar que o ambiente Anaconda também possui vasta documentação e conteúdos diversos para facilitar o percurso de aprendizado e utilização dos interessados⁴.

⁴ Disponível em: <https://docs.anaconda.com/anaconda/com>. Acesso realizado em 30/12/2020

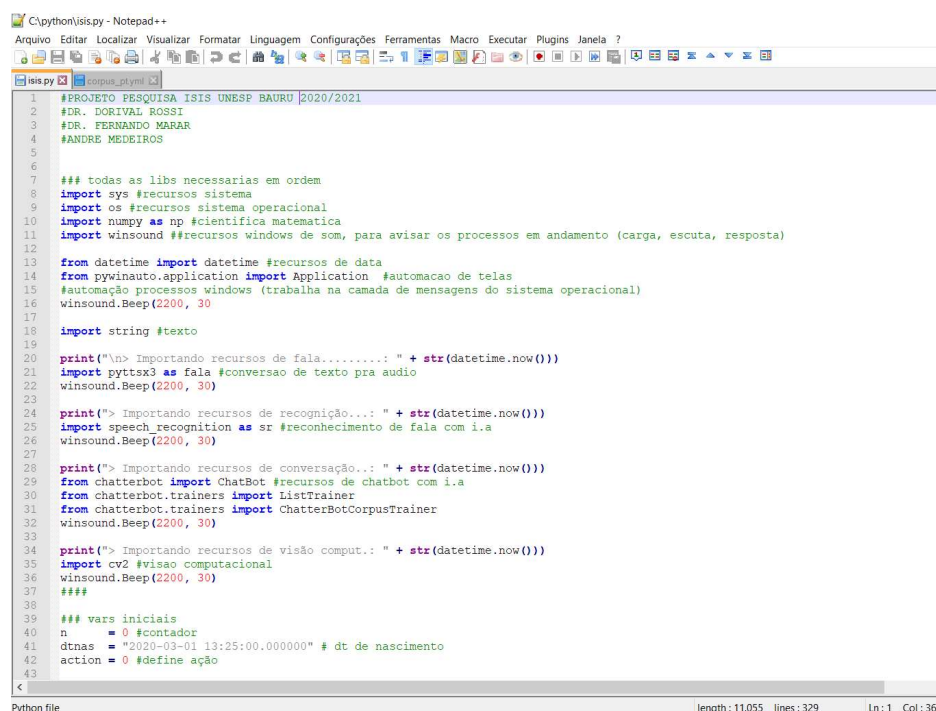
3.6 Aplicativo Notepad++

Da mesma forma como existem ambientes de desenvolvimento mais elaborados como o Anaconda existem também “editores de texto” mais “simples”, que podem ser utilizados por aqueles que preferem uma experiência de programação mais *clean* (limpa, rápida, elegante) e/ou tradicional, focando apenas em codificação de texto e facilitadores para visualmente organizar melhor o código escrito, conceito também chamado de *endentação* (neologismo para termo inglês *indentation*) de código fonte.

Neste cenário, destaca-se o caso do aplicativo Notepad++, já conhecido na comunidade de desenvolvimento de software, que mantém suas características de simplicidade, porém com muitos adeptos e recursos facilitadores para desenvolvedores contumazes.

Abaixo observa-se o exemplo do código fonte que estamos disponibilizando como resultado e protótipo desta pesquisa, em linguagem Python, reforçando que este código fonte está disponível como Anexo 2, ao final deste trabalho, e também na íntegra e online, sob licença *Creative Commons*, na plataforma *GitHub*, que serão adequadamente abordados nos próximos tópicos.

Figura 11 – Editor de texto e códigos Notepad++



```

1 #PROJETO PESQUISA ISIS UNESP BAURU 2020/2021
2 #DR. DORIVAL ROSSI
3 #DR. FERNANDO MARAR
4 #ANDRE MEDEIROS
5
6
7 ### todas as libs necessarias em ordem
8 import sys #recursos sistema
9 import os #recursos sistema operacional
10 import numpy as np #cientifica matematica
11 import winsound ##recursos windows de scm, para avisar os processos em andamento (carga, escuta, resposta)
12
13 from datetime import datetime #recursos de data
14 from pywinauto.application import Application #automacao de telas
15 #automação processos windows (trabalha na camada de mensagens do sistema operacional)
16 winsound.Beep(2200, 30)
17
18 import string #texto
19
20 print("\n> Importando recursos de fala.....: " + str(datetime.now()))
21 import pyttsx3 as fala #conversao de texto pra audio
22 winsound.Beep(2200, 30)
23
24 print("> Importando recursos de reconhecimento...: " + str(datetime.now()))
25 import speech_recognition as sr #reconhecimento de fala com i.a
26 winsound.Beep(2200, 30)
27
28 print("> Importando recursos de conversação...: " + str(datetime.now()))
29 from chatterbot import ChatBot #recursos de chatbot com i.a
30 from chatterbot.trainers import ListTrainer
31 from chatterbot.trainers import ChatterBotCorpusTrainer
32 winsound.Beep(2200, 30)
33
34 print("> Importando recursos de visão comput.: " + str(datetime.now()))
35 import cv2 #visao computacional
36 winsound.Beep(2200, 30)
37
38
39 ### vars iniciais
40 n = 0 #contador
41 dtnas = "2020-03-01 13:25:00.000000" # dt de nascimento
42 action = 0 #define ação
43

```

Fonte: Captura de tela do próprio pesquisador

3.7 Métodos Ágeis e Scrum

O Manifesto Ágil para Desenvolvimento de Software⁵ preconiza que “estamos descobrindo maneiras melhores de desenvolver softwares, fazendo-o nós mesmos e ajudando outros a fazerem o mesmo. Através deste trabalho, passamos a valorizar”:

Tabela 1 – Princípios Ágeis

Indivíduos e Interações	... mais que processos e ferramentas
Software em funcionamento	... mais que documentação abrangente
Colaboração com o cliente	... mais que negociações de contratos
Responder a mudanças	... mais que seguir um plano

Fonte: <https://agilemanifesto.org/iso/ptbr/manifesto.html>

E o manifesto finaliza da seguinte maneira: “ou seja, mesmo havendo valor nos itens à direita, valorizamos mais os itens à esquerda.”.

O Manifesto Ágil foi criado em 2001 por 17 programadores que notaram semelhanças em suas formas de trabalho e os bons resultados obtidos seguindo essas premissas, conforme os cenários que se apresentavam. Ao assinarem o manifesto se comprometeram a praticar e a disseminar essas propostas a todo o mundo. Os signatários originais são: Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, Dave Thomas.

Os 12 princípios que sustentam o Manifesto Ágil são:

1. Nossa maior prioridade é satisfazer o cliente, através da entrega adiantada e contínua de software de valor.
2. Aceitar mudanças de requisitos, mesmo no fim do desenvolvimento. Processos ágeis se adequam a mudanças, para que o cliente possa tirar vantagens competitivas.

⁵ Disponível em: <https://agilemanifesto.org/iso/ptbr/manifesto.html>. Acesso em 30/12/2020.

3. Entregar software funcionando com frequência, na escala de semanas até meses, com preferência aos períodos mais curtos.
4. Pessoas relacionadas à negócios e desenvolvedores devem trabalhar em conjunto e diariamente, durante todo o curso do projeto.
5. Construir projetos ao redor de indivíduos motivados. Dando a eles o ambiente e suporte necessário, e confiar que farão seu trabalho.
6. O Método mais eficiente e eficaz de transmitir informações para, e por dentro de um time de desenvolvimento, é através de uma conversa cara a cara.
7. Software funcional é a medida primária de progresso.
8. Processos ágeis promovem um ambiente sustentável. Os patrocinadores, desenvolvedores e usuários, devem ser capazes de manter indefinidamente, passos constantes.
9. Contínua atenção à excelência técnica e bom design, aumenta a agilidade.
10. Simplicidade: a arte de maximizar a quantidade de trabalho que não precisou ser feito.
11. As melhores arquiteturas, requisitos e designs emergem de times auto-organizáveis.
12. Em intervalos regulares, o time reflete em como ficar mais efetivo, então, se ajustam e otimizam seu comportamento de acordo.

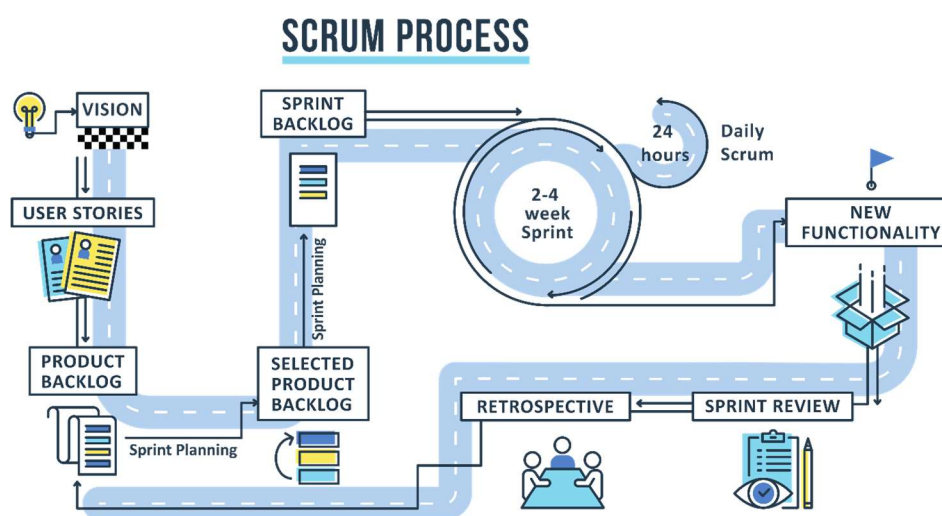
Diante desta perspectiva, neste trabalho optamos por uma metodologia ágil já bastante difundida e adotada tanto em universidades como em empresas ao redor do mundo: o Scrum.

Scrum é uma metodologia utilizada para facilitar e agilizar o serviço necessário para se atingirem os objetivos de cada projeto. Inicialmente foi criado para potencializar o processo de desenvolvimento de projetos e softwares, mas hoje é utilizado em diversas áreas do conhecimento. Trata-se de uma espécie de quadro que fica exposto nos setores para todos acompanharem, de forma a obter um tipo de planejamento completamente organizado. Nele estão contidas informações sobre o

que a pessoa/equipe precisa fazer, o que já se fez, quais as próximas tarefas a serem executadas e projetos finalizados.

Além disso, são realizadas reuniões periodicamente para ver se o método está funcionando. Para facilitar e otimizar ainda mais o trabalho, já existem aplicativos online. Abaixo destaca-se um quadro com um modelo tradicional de Scrum:

Figura 12 – Macroprocesso Scrum



Fonte: <https://www.appvizer.com.br/>

3.8 Licença Creative Commons

Quando se aborda o tema da inventividade humana, nota-se que há uma ramificação e complexidade de alto nível. Há muitos séculos o ser humano vem criando e desenvolvendo ideias que suscitam questões como a propriedade intelectual, registros de patentes, licenças de uso, entre outros tópicos correlatos.

É importante lembrar que todos devem ter direito a manter e usufruir aquilo que desenvolveu e criou, mas conforme acompanhamos no decorrer deste trabalho, muitas novas tecnologias e arranjos humanos tem acontecido, o que proporcionaram “novas formas de licenciamento”. É o caso da organização e modalidade de licenciamento conhecida como *Creative Commons*, que foi eleita para licenciar todo este trabalho.

Figura 13 – Logotipo Creative Commons no Brasil



Fonte: Creative Commons Organization

Segundo a própria definição oferecida pela unidade brasileira da organização⁶:

“Creative Commons é uma organização sem fins lucrativos que permite o compartilhamento e uso da criatividade e do conhecimento através de instrumentos jurídicos gratuitos. O Creative Commons ajuda você a compartilhar legalmente seu conhecimento e criatividade para construir um mundo mais justo, acessível e inovador. Nós desbloqueamos todo o potencial da internet para impulsionar uma nova era de desenvolvimento, crescimento e produtividade.”

A organização oferece várias modalidades de licenciamento, permitindo que o profissional e/ou pesquisador utilize o modelo mais adequado para sua intenção e necessidade. Além do descritivo da licença, existem “selos” que são utilizados para identificação, facilitando a compreensão daqueles que desejam utilizar os produtos e soluções disponibilizados sob tais licenciamentos. Os principais selos são os seguintes, conforme o site da organização:

Figura 14 – Selo 1 Creative Commons



Atribuição CC BY

Esta licença permite que outros distribuam, remixem, adaptem e criem a partir do seu trabalho, mesmo para fins comerciais, desde que lhe atribuam o devido crédito pela criação original. É a licença mais flexível de todas as licenças disponíveis. É recomendada para maximizar a disseminação e uso dos materiais licenciados.

Fonte: Creative Commons Organization

⁶ Disponível em: <https://br.creativecommons.org/sobre/>. Acesso em 30/12/2020

Cabe lembrar que este trabalho está totalmente licenciado sob esta modalidade (CC BY) do Creative Commons. Permitindo livre acesso e comprovando o caráter de compartilhamento preconizado durante o trabalho.

Figura 15 – Selo 2 Creative Commons



Atribuição-Compartilhaigual CC BY-SA

Esta licença permite que outros remixem, adaptem e criem a partir do seu trabalho, mesmo para fins comerciais, desde que lhe atribuam o devido crédito e que licenciem as novas criações sob termos idênticos. Esta licença costuma ser comparada com as licenças de software livre e de código aberto "copyleft". Todos os trabalhos novos baseados no seu terão a mesma licença, portanto quaisquer trabalhos derivados também permitirão o uso comercial. Esta é a licença usada pela Wikipédia e é recomendada para materiais que seriam beneficiados com a incorporação de conteúdos da Wikipédia e de outros projetos com licenciamento semelhante.

Fonte: Creative Commons Organization

Figura 16 – Selo 3 Creative Commons



Atribuição-SemDerivações CC BY-ND

Esta licença permite a redistribuição, comercial e não comercial, desde que o trabalho seja distribuído inalterado e no seu todo, com crédito atribuído a você.

Fonte: Creative Commons Organization

Figura 17 – Selo 4 Creative Commons



Atribuição-NãoComercial
CC BY-NC

Esta licença permite que outros remixem, adaptem e criem a partir do seu trabalho para fins não comerciais, e embora os novos trabalhos tenham de lhe atribuir o devido crédito e não possam ser usados para fins comerciais, os usuários não têm de licenciar esses trabalhos derivados sob os mesmos termos.

Fonte: Creative Commons Organization

Figura 18 – Selo 5 Creative Commons



Atribuição-NãoComercial-Compartilha Igual
CC BY-NC-SA

Esta licença permite que outros remixem, adaptem e criem a partir do seu trabalho para fins não comerciais, desde que atribuam a você o devido crédito e que licenciem as novas criações sob termos idênticos.

Fonte: Creative Commons Organization

Figura 19 – Selo 6 Creative Commons



Atribuição-SemDerivações-SemDerivados
CC BY-NC-ND

Esta é a mais restritiva das nossas seis licenças principais, só permitindo que outros façam download dos seus trabalhos e os compartilhem desde que atribuam crédito a você, mas sem que possam alterá-los de nenhuma forma ou utilizá-los para fins comerciais.

Fonte: Creative Commons Organization

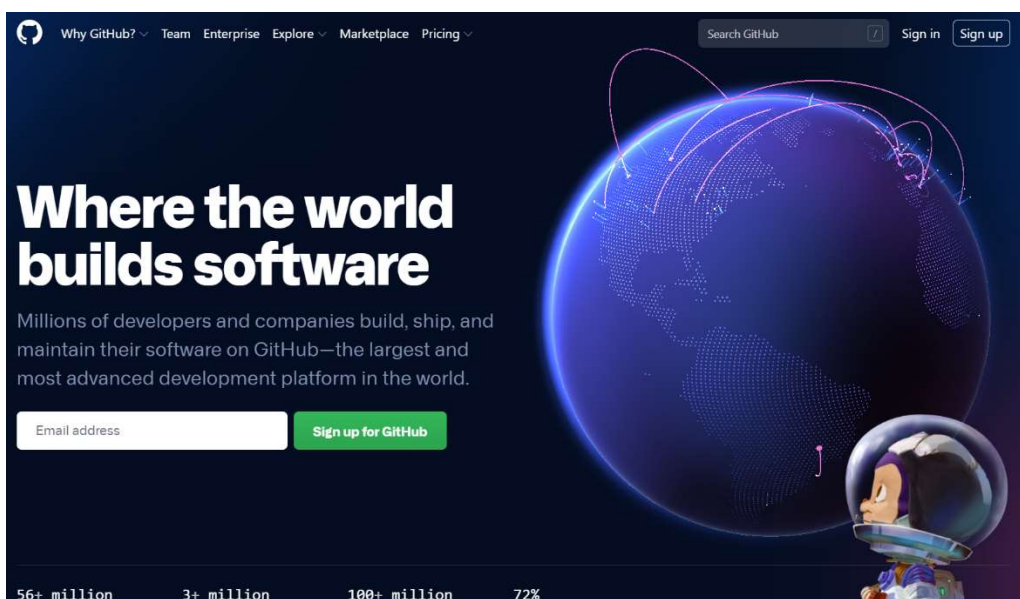
Todas as licenças supracitadas estão disponíveis no site oficial da organização, onde os interessados poderão aprofundar-se no entendimento e leitura detalhada dos termos legais, conforme suas demandas de utilização.

3.9 Plataforma GitHub

Trata-se de um ambiente para publicação e troca de experiências em programação com, praticamente, espaço para todas as linguagens atualmente utilizadas.

O código fonte do protótipo desenvolvido neste trabalho está disponibilizado nessa plataforma para que todos possam acessá-lo, além de constar, textualmente, como anexo 2 deste trabalho. Também é importante lembrar que os procedimentos para reprodução do ambiente necessário para obtenção de resultados, se não iguais, ao menos semelhantes aos obtidos por este trabalho estão disponíveis no anexo 1.

Figura 20 – Site Plataforma Github



Fonte: Github.com

O GitHub oferece uma plataforma de compartilhamento e colaboração inédito na história recente de tecnologias para softwares, sendo hoje usado por milhões de pessoas que atestam sua eficácia e continuam participando e colaborando para evolução do ecossistema de desenvolvimento de software, com forte inclinação para o *open source* e movimento *maker*.

4 MATERIAIS E MÉTODOS

4.1 Tecnologias e recursos utilizados

Para o desenvolvimento deste projeto foram utilizadas as tecnologias e os processos descritos no referencial teórico. Em resumo, utiliza-se a linguagem de programação Python 3, como base para escrita do protótipo, ambiente Anaconda *Open Source* para facilitar a integração da linguagem Python com pacotes e recursos de inteligência artificial, Aplicativo Notepad++ para edição do código fonte original do protótipo, Método Ágil Scrum para acelerar os resultados, Licença Creative Commons para disponibilizar cópias e direitos de mudanças ao público em geral e Plataforma GitHub para publicação online do código fonte na íntegra.

Todo o desenvolvimento foi realizado no notebook pessoal do pesquisador, com a configuração de processador Intel Core i7, HD SSD 240 Gb, 8 Gb de memória RAM e sistema operacional Windows 10, já adquirido em momento anterior ao ingresso na pós-graduação.

4.2 Descrição do Protótipo

O presente trabalho apresenta como produto final um protótipo de software com recursos de inteligência artificial, desenvolvido conforme as premissas do movimento *maker*.

Esta aplicação visa explorar a praticidade dos conceitos *maker* e rotinas de inteligência artificial para demonstrar a viabilidade técnica de se produzir um protótipo de software para assistência pessoal, com recursos *open source* e baixíssimo custo, sendo possível reproduzir tais resultados, conforme procedimentos descritos nos anexos 1 e 2 deste trabalho.

4.3 Aplicação de Software

Hoje existem diversas aplicações de assistência pessoal desenvolvidas por grandes empresas e que necessitam de muito poder de processamento, tecnologias fechadas e alto custo de desenvolvimento e sustentação. Diante disso foi proposto desenvolver um protótipo de baixíssimo custo, executado conforme as premissas do movimento *maker* e utilizando-se de tecnologias contemporâneas de inteligência

artificial, que contemplasse minimamente recursos como responder a comandos por voz, gerar respostas de conversação, executar ações específicas, ativar câmera e reconhecer determinados objetos, sendo *open source* e tendo sua reprodutibilidade garantida através dos procedimentos descritos neste trabalho.

4.4 Metodologia utilizada

Além de referencial teórico e técnico, experiência de mercado e pesquisas na pós-graduação, optou-se pelas seguintes etapas necessárias para o desenvolvimento do protótipo:

- 1- Pesquisa exploratória e bibliográfica sobre o tema da pesquisa;
- 2- Construção do referencial teórico e exploração sobre aplicações similares já desenvolvidas;
- 3- Definição do objeto e problema de pesquisa;
- 4- Levantamento de requisitos a fim de determinar as funcionalidades inerentes à aplicação de software;
- 5- Exploração sobre a arquitetura de software utilizadas no experimento, bem como as ferramentas de apoio como frameworks e ambiente de trabalho;
- 6- Instalação e configuração do ambiente de desenvolvimento de software e de hardware;
- 7- Elaboração do projeto de software;
- 8- Implementação/codificação da aplicação de software;
- 9- Testes da aplicação de software e do hardware utilizado.

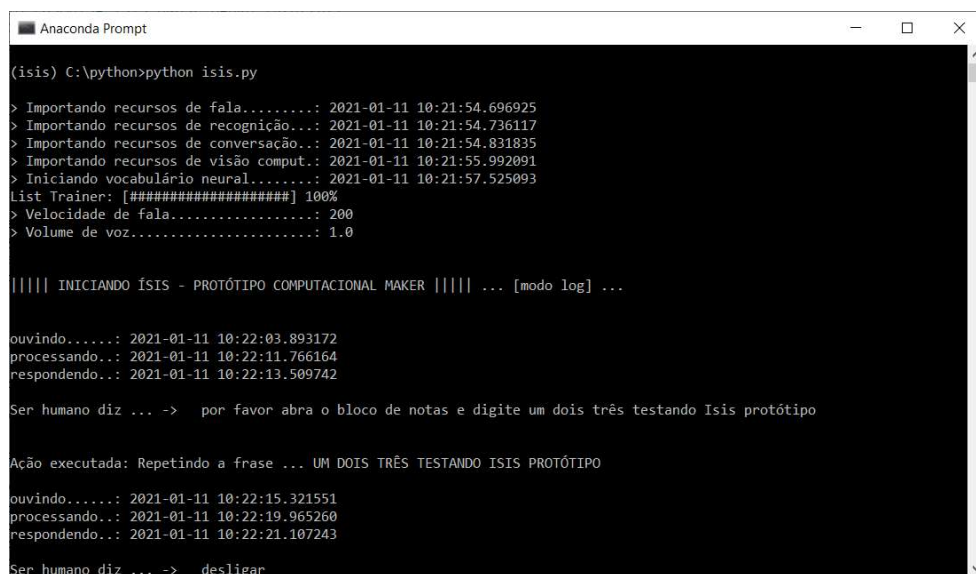
5 RESULTADOS OBTIDOS

O protótipo, apesar de ter uma estrutura basilar simples, traça os caminhos necessários para a sua evolução, além de unir consistentemente diversos recursos tecnológicos em uma única aplicação, utilizando-se de baixíssimo custo, demonstrando a tese de que o movimento *maker* e a inteligência artificial podem juntos trazer novas perspectivas de trabalho científico e corporativo.

Além disso o protótipo desenvolvido foi apresentado em dois seminários internos da UNESP Bauru, onde outros pesquisadores puderam realizar questionamentos e ver o produto em funcionamento.

O projeto proporcionou uma experiência muito interessante em que foi possível identificar as peculiaridades das tecnologias disponíveis e a necessidade de unir diversos conhecimentos para obtenção do resultado esperado. Alguns problemas foram identificados como, por exemplo, compatibilidades entre pacotes, que será abordado a seguir.

Figura 21 – Captura de tela do protótipo



```

Anaconda Prompt
(isis) C:\python>python isis.py

> Importando recursos de fala.....: 2021-01-11 10:21:54.696925
> Importando recursos de reconhecimento...: 2021-01-11 10:21:54.726117
> Importando recursos de conversação...: 2021-01-11 10:21:54.831835
> Importando recursos de visão comput.: 2021-01-11 10:21:55.992091
> Iniciando vocabulário neural.....: 2021-01-11 10:21:57.525093
List Trainer: [#####] 100%
> Velocidade de fala.....: 200
> Volume de voz.....: 1.0

||||| INICIANDO ÍSIS - PROTÓTIPO COMPUTACIONAL MAKER ||||| ... [modo log] ...

ouvindo.....: 2021-01-11 10:22:03.893172
processando...: 2021-01-11 10:22:11.766164
respondendo...: 2021-01-11 10:22:13.509742

Ser humano diz ... ->  por favor abra o bloco de notas e digite um dois três testando Isis protótipo

Ação executada: Repetindo a frase ... UM DOIS TRÊS TESTANDO ÍSIS PROTÓTIPO

ouvindo.....: 2021-01-11 10:22:15.321551
processando...: 2021-01-11 10:22:19.965260
respondendo...: 2021-01-11 10:22:21.107243

Ser humano diz ... ->  desligar
  
```

Fonte: Próprio pesquisador

Figura 22 – Diagrama básico de execução do protótipo



Fonte: Próprio pesquisador

É importante lembrar que como foram utilizadas tecnologias bastante recentes, além de várias bibliotecas *open source*, pode existir certa dificuldade em encontrar conteúdo atualizado em língua portuguesa. Enfrenta-se também pequenos problemas de versionamento das bibliotecas/pacotes, devido ao uso conjunto das mesmas e de vários recursos encapsulados que podem conflitar em determinados momentos de desenvolvimento. A comunidade em torno dos pacotes evolui e apresenta novas formas de resolver “o mesmo problema”. Por isso é importante manter atualizada a documentação e o acompanhamento de novidades em cada uma delas.

Devido à característica de reprodutibilidade desse trabalho, tenta-se minimizar ao máximo a ocorrência de qualquer erro na execução, aplicação e testes, conforme evidencia-se nos anexos deste trabalho, onde constam os procedimentos necessários para a reprodução do ambiente e execução do protótipo em funcionamento.

Soma-se a isso, ainda, a lembrança de que se trata de temas contemporâneos em constante evolução e transformação que exigem do profissional e/ou pesquisador uma certa flexibilidade para acompanhar esse cenário, obtendo os resultados almejados e, por que não, nunca antes imaginados.

O código fonte e demais procedimentos necessários para montagem do ambiente encontram-se, além dos anexos 1 e 2, com acesso livre e domínio público no endereço: <https://github.com/armedeiros/isis> (acesso em 20/03/2021).

Todo o respaldo técnico e o arcabouço de pesquisa desenvolveu-se dentro da UNESP campus Bauru. Também houve apoio em disponibilização de tempo e

recursos tecnológicos da TOTVS Bauru, onde o pesquisador também atua corporativamente, já citada nos agradecimentos deste trabalho.

6 CONCLUSÕES

Conclui-se que é viável empreender esforços em reunir os recursos tecnológicos contemporâneos de inteligência artificial e o movimento *maker*, colocando em prática desde o desenho, a pesquisa necessária, desenvolvimento de códigos, execução, testes, até a correção de erros, algumas vezes ainda não documentados.

Isso habilita o profissional e/ou pesquisador a ampliar seu instrumental de conhecimento, além de incentivar que toda a comunidade, seja científica ou corporativa, engaje-se nesse tipo de pensamento, nessas metodologias e tecnologias, trazendo inúmeros outros benefícios para um conjunto maior da sociedade.

Notadamente os avanços nessa linha de pesquisa continuarão a passos largos, cabendo ao profissional da área manter-se sempre atualizado e que seus produtos possam possibilitar que outras pessoas o modifiquem, adaptem e até mesmo gerem negócios a partir dessas tecnologias *open source* com o pensamento *maker* em paralelo, construindo soluções cada vez mais elaboradas.

Pessoas, profissionais, pesquisadores e quem mais desejar poderá munir-se das ferramentas, ambientes e colaboradores necessários e empreender muito mais que desejar.

Soma-se a isso a perspectiva de que novas e mais aprofundadas pesquisas poderão ser feitas a partir da ideia colocada em prática neste trabalho, evoluindo gradativamente o protótipo e que poderão ser desenvolvidas em continuidade na pós-graduação.

REFERÊNCIAS

ALBINO, J. P. **Uma Abordagem para Criação de Valor em Dados Abertos para Pequenas e Médias Empresas Utilizando o Ecossistema R**. III Encontro Internacional de Dados, Tecnologia e Informação. 2016.

ANDERSON, Chris. **MAKERS. The New Industrial Revolution**. USA: Crown Business. 2012.

BELLUZZO, Regina C. B. **Competências na era digital: desafios tangíveis para bibliotecários e educadores**. ETD – Educação Temática Digital. Disponível em: <<https://periodicos.sbu.unicamp.br/ojs/index.php/etd/article/view/772>>. Acesso em: 28 mai. 2019.

CANALTECH. **O que é Open Source**. Disponível em: <<https://canaltech.com.br/produtos/O-que-e-open-source/#:~:text=Open%20source%20%C3%A9%20um%20termo,ponto%20de%20vista%20essencialmente%20t%C3%A9cnico>>. Acesso em: 24 out. 2020.

GLEICK, James. **A informação: uma história, uma teoria, uma enxurrada**. Editora Companhia das Letras, 2013.

LAUREANO, E. A. G. C. **ConsultBot** - um chatterbot consultor para ambientes virtuais de estudo na Internet. 1999.

LEVY, Pierre. **O que é o Virtual?** São Paulo: Editora 34, 1996.

LEONHARDT, M. D. et al. **Um Chatterbot para uso em ambiente educacional**. II Ciclo de Palestras sobre Novas Tecnologias na Educação, 2003.

LYRA, J. G. M., MEIRIÑO, M. **BITCOIN E BLOCKCHAIN: Aplicações além da moeda virtual**. Fórum Inovação e propriedade intelectual. Disponível em:

<<https://azepdf.com/bitcoin-e-blockchain-blockchainbrasilorg.html>>. Acesso em: 1 mar. 2019.

MANOVICH, Lev. **Novas mídias como tecnologia e ideia**: dez definições. O chip e o caleidoscópio: reflexões sobre as novas mídias. São Paulo: Editora SENAC, p. 24-50, 2005.

MEDEIROS, A.R. **Bitcoin e Blockchain: minerando ideias e construindo conceitos**. In: ROSSI, Dorival; GONÇALVES, Juliana; MOON, Rodrigo (Org.). Movimento Maker e Fab Labs: design, inovação e tecnologia em tempo real. Bauru: UNESP. FAAC, 2019.

MEDEIROS, Richerland. **Inferência de emoções em textos curtos**. Disponível em: <<http://hdl.handle.net/11449/150974>>. Acesso em: 5 mar. 2020.

NAKAMOTO, Satoshi. **Bitcoin: A peer-to-peer electronic cash system**. Disponível em: <<https://bitcoin.org/bitcoin.pdf>>. Acesso em: 28 out. 2020.

NEVES, André; BARROS, Flávia. **iAIML**: um mecanismo para o tratamento de intenção em Chatterbots. 2005. Tese (Doutorado). Programa de Pós-Graduação em Ciência da Computação, Universidade Federal de Pernambuco, Recife, 2005.

PEREIRA, Luís Moniz. **Inteligência Artificial**: mito e ciência. Revista Colóquio-Ciências, v. 3, p. 1-13, 1988.

ROSSI, Dorival et.al. **Sagui Lab**: Um experimento educacional híbrido. In: Ensaios em design, ações inovadoras. Bauru: Editora Canal 6, 2016.

ROTHERMEL, Alessandra; DOMINGUES, Maria José Carvalho. **MARIA**: Um chatterbot desenvolvido para os estudantes da disciplina “Métodos e Técnicas de Pesquisa em Administração”. SEGET-SIMPÓSIO DE EXCELÊNCIA EM GESTÃO E TECNOLOGIA, Resende, 2007.

RUSSELL, Stuart; NORVIG, Peter. **Intelligent agents**. Artificial intelligence: A modern approach, v. 74, p. 46-47, 1995.

SGANDERLA, Rachele Bianchi; FERRARI, Débora Nice; GEYER, Cláudio FR. **BonoBOT**: Um Chatterbot para Interação com Usuários em um Sistema Tutor Inteligente. In: Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE). 2003. p. 435-444.

TEIXEIRA, João. **O que é inteligência artificial**. E-Galáxia, 2019.

TEIXEIRA, Sérgio et al. **Chatterbots em ambientes de aprendizagem** – uma proposta para a construção de bases de conhecimento. In: Anais do Workshop de Informática na Escola. 2005.

ULRICH, Beck. **O que é globalização**. São Paulo: Editora Paz Terra, 1999.

WHITBY, Blay. **The turing test**: Al's biggest blind alley?'. Machines and thought: The legacy of Alan Turing, v. 1, p. 53-62, 1996.

Anexo 1

Etapas e procedimentos de execução

Passo-a-passo de instalação e execução do ambiente para que possam ser reproduzidos os resultados descritos neste trabalho, seja a título de pesquisa ou como de empreendimento pessoal.

1. Instalação Notepad++
2. Certifique-se que seu Windows já possua ferramentas de Build do Visual Studio 2017, senão instale: Visual Studio Compiler Sugestão de link, acesso em 30/12/2020: https://aka.ms/vs/16/release/vc_redist.x64.exe
3. Instalação Anaconda Versao Python 3.7
4. Após instalação entrar no Anaconda Navigator para criar o seu ambiente de trabalho, aqui utilizamos o nome “isis”, ou seja, o ambiente terá o mesmo nome do protótipo. Aguarde a criação do ambiente.
5. Acesso o Anaconda Prompt:

Na janela que abrir, digite, onde <enter> se refere a pressionar a tecla enter do teclado:

```
cd\ <enter>
```

Crie uma pasta chamada “python”, digitando:

```
mkdir python <enter>
```

Observação: Nesta pasta será colocado o arquivo isis.py que é o código fonte completo do protótipo. Disponível como Anexo 2.

Ainda no Anaconda Prompt, ative o ambiente “isis” digitando:

```
activate isis <enter>
```

Neste ponto, o ambiente está apto para receber a instalação de todas as bibliotecas/pacotes necessários, conforme comando elencados abaixo.

6. Atualizar o instalador PIP

```
python -m pip install --upgrade pip <enter>
```

```
#apoio para casos de exceção
```

```
#pip install <pacote> --use-feature=2020-resolver
```

7. Bibliotecas necessárias, aguarde a instalação individual de cada uma e depois instale a próxima.

```
pip install numpy <enter>
```

```
##>pip install numpy==1.19.3
```

```
pip install pywinauto <enter>
```

```
pip install pyttsx3 <enter>
```

```
pip install speechrecognition <enter>
```

```
#para o chatterbot
```

```
pip install spacy <enter>
```

```
#garante os pacotes de linguagem
```

```
python -m spacy download em <enter>
```

```
#https://spacy.io/models/pt
```

```
pip install chatterbot <enter>
```

```
pip3 install chatterbot_corpus <enter>
```

```
pip install ez_setup <enter>
```

```
pip install pipwin <enter>
```

```
pip install pyauto <enter>
```

```
pip install pyaudio <enter>
```

```
##      https://www.lfd.uci.edu/~gohlke/pythonlibs/#pyaudio  
baixar a versão adequada ao seu pc
```

```
## PyAudio-0.2.11-cp37-cp37m-win_amd64.whl
```

```
pip install spyder <enter>
```

```
pip install msvc-runtime <enter>  
pipwin install pocketsphinx <enter>  
##cv2  
pip install opencv-python <enter>  
pip install nltk <enter>
```

Anexo 2

Software modelo assistência computacional

Aqui pretendemos cruzar todas as referências que tratamos no decorrer do trabalho para demonstrar a possibilidade técnica reprodutível e aplicável de criação de um software utilizando-se amplamente de conceitos do movimento maker, tecnologia contemporânea para inteligência artificial e open source.

O software nomeado ÍSIS nasceu da demanda de testar e validar conceitos estudados em algumas áreas do conhecimento, conforme anteriormente relatadas, de modo que evidenciasse as necessidades, obstáculos e recursos para sua criação.

Todo o código fonte está sob licença *Creative Commons* e publicado gratuitamente no GitHub, a fim de que qualquer pessoa possa alterar livremente e criar mais recursos em cima de sua base estrutural.

```
#Prof. Dr. DORIVAL ROSSI
#Prof. Dr. FERNANDO MARAR
#ANDRE MEDEIROS (mestrando)
#UNESP BAURU

### todas as libs necessarias em ordem,
### os prints dos imports podem ser removidos, servem apenas para verificar o
tempo de carga
### os beeps servem de alerta e também são opcionais
### o código preza pela simplicidade para ser acessível a todos os públicos

import sys #recursos sistema
import os #recursos sistema operacional
import numpy as np #cientifica matematica
import winsound ##recursos windows de som, para avisar os processos em andamento
(carga, escuta, resposta)

from datetime import datetime #recursos de data
from pywinauto.application import Application #automacao de telas/processos
windows (trabalha na camada de mensagens do sistema operacional)

import string #texto

winsound.Beep(2200, 30)
```

```

print("\n> Importando recursos de fala.....: " + str(datetime.now()))
import pyttsx3 as fala #conversao de texto pra audio
winsound.Beep(2200, 30)

print("> Importando recursos de reconhecimento...: " + str(datetime.now()))
import speech_recognition as sr #reconhecimento de fala com i.a
winsound.Beep(2200, 30)

print("> Importando recursos de conversação...: " + str(datetime.now()))
from chatterbot import ChatBot #recursos de chatbot com i.a
from chatterbot.trainers import ListTrainer
from chatterbot.trainers import ChatterBotCorpusTrainer
winsound.Beep(2200, 30)

print("> Importando recursos de visão comput.: " + str(datetime.now()))
import cv2 #visao computacional
winsound.Beep(2200, 30)
####

### vars iniciais
n      = 0 #contador
dtnas  = "2020-03-03 13:33:00.000000" # dt de 'nascimento'
action = 0 #define ação

rec    = sr.Recognizer() #tratamento do reconhecimento de fala
###

###tratamento da robo
##bot = ChatBot('ISIS') #inicia o Bot no ambiente

bot = ChatBot('ISIS',
              storage_adapter='chatterbot.storage.SQLStorageAdapter',

              logic_adapters=[
                  'chatterbot.logic.MathematicalEvaluation',
                  #'chatterbot.logic.TimeLogicAdapter',
                  'chatterbot.logic.BestMatch',

```

```

],
#filters=[filters.get_recent_repeated_responses]
database_uri='sqlite:///database.db'
)
###

winsound.Beep(2200, 30)

#carga inicial de vocabulario, aqui deve-se incluir aquele que for mais
apropriado para necessidade

#pode ser arquivos com frases ou direcionado pra ações específicas

chats = ['Como vai?', 'Tudo legal', 'Voce pode me ajudar?', 'Claro que posso!
Diga?', 'Voce Conhece ÍSIS?', 'Ouvi dizer que esse software eh muito bom!', 'Radio',
'Noticias sao importantes', 'Qual seu nome?', 'Me chamo ÍSIS muito prazer em conhecer
voce']

chats += ['Oi!', 'Olah Pessoa Humana!', 'O dia esta bom', 'Dias assim sao bons',
'Voce sabe sobre muita coisa?', 'Posso aprender se voce me ensinar', 'Como voce se
chama?', 'Me chamam de ÍSIS.', 'Quantos anos voce tem?', 'Eu nao tenho idade...']

chats += ['Voce gosta?', 'Gosto nao se discute', 'Ja ouviu falar?', 'Depende,
posso pesquisar...']

chats += ['Desligar', 'Ja vai? Ateh logo!', 'Voce sabe se vai chover amanha?',
'Vamos ver o clima...']

chats += ['Entender', 'Sempre refletindo', 'Bom dia', 'Olah ser humano! Tenha
um bom dia.']

chats += ['Orientador UNESP?', 'Professor Dorival']

#manda a robo treinar com o vocabulario inicial
print("> Iniciando vocabulário neural.....: " + str(datetime.now()))

# treinamento do bot
trainer = ListTrainer(bot)
trainer.train(chats)

###exemplos de treinos alternativos
#trainer = ChatterBotCorpusTrainer(ChatBot)
#trainer.train(chats)
#trainer.train("chatterbot.corpus.english") portuguese

#bot.set_trainer(ListTrainer)
#bot.train(chats) #executa o treino com o vocabulario fornecido

##

```

```

#tratamento de voz
voz = fala.init('sapi5')
ids = voz.getProperty('voices') #vozes disponiveis

rate = voz.getProperty('rate') #velocidade
print("> Velocidade de fala.....: " + str(rate))
voz.setProperty('rate', rate + 45) #ajusta velocidade da voz, buscando algo
mais 'natural' +20

volume = voz.getProperty('volume')
print("> Volume de voz.....: " + str(volume))
voz.setProperty('volume', 5.0) #ajusta o volume maximo
###

###funcao de visao computacional simples para detecção de objetos cor verde,
também pode-se alterar para a necessidade em específico
def cv():

    #Iniciamos camera
    captura = cv2.VideoCapture(0)

    while(1):

        #Captura img em RGB -> HSV
        _, imagen = captura.read()
        hsv = cv2.cvtColor(imagen, cv2.COLOR_BGR2HSV)

        #Faixa de cores desejadas
        verde_baixos = np.array([49, 50, 50], dtype=np.uint8) #49 50 50
        verde_altos = np.array([80, 255, 255], dtype=np.uint8) # 80 255

        #mascara com essa faixa de cores
        mask = cv2.inRange(hsv, verde_baixos, verde_altos)

        #encontrar a area onde o objeto esta
        moments = cv2.moments(mask)
        area = moments['m00']

```

```

if(area > 200000):

    #centro do objeto detectado
    x = int(moments['m10']/moments['m00'])
    y = int(moments['m01']/moments['m00'])

    #Mostramos coordenadas
    #print ("x = ", x)
    #print ("y = ", y)

    #retangulo de detecção
    cv2.rectangle(imagen, (x-5, y-5), (x+5, y+5), (0,0,255), 2)

    cv2.putText(imagen, "pos x,y:"+ str(x)+" "+str(y),
(x+10,y+10), cv2.FONT_HERSHEY_SIMPLEX, 1, (255,255,255), 1) #cv2.FONT_HERSHEY_SIMPLEX

    #mascara x original
    ##cv2.imshow('mask', mask)
    cv2.imshow('Câmera Atual do Dispositivo', imagen)

    #tecla = cv2.waitKey(5) & 0xFF = shutdown
    if cv2.waitKey(1) & 0xFF == ord('q'):
        captura.release()
        break

    #libera janela
    cv2.destroyAllWindows()

###----

#Loop 'robótico'
#inicia captura de audio
with sr.Microphone() as s:

    #limpa ruido
    rec.adjust_for_ambient_noise (s)

    #loop de dialogo
    while True:

```

```

rec.adjust_for_ambient_noise(s)
if (n == 0):
    print("\n\n||||| INICIANDO ÍSIS - PROTÓTIPO COMPUTACIONAL
MAKER ||||| ... [modo log] ... \n\n")

    voz.say("Oi Eu sou a PROTÓTIPO ÍSIS, como posso ajudar?")
    voz.runAndWait()

##estrutura de ssegurança do bloco de execução, tratando exceções
try:
    time = datetime.now()

    print('ouvindo.....: ' + str(time))
    winsound.Beep(2200, 30)

#captura o audio
    rec.adjust_for_ambient_noise(s)

    audio = rec.listen(s)

    time = time.now()

    print('processando...: ' + str(time))
    winsound.Beep(1000, 30)
    #processa via google open
    #call api para idioma pt-br/en
#existem outras formas e calls, pode-se alterar caso necessário
#exemplo:
#lib sphinx para off-line, em pt-br ainda em testes/desenvolvimento
#audio_capt = rec.recognize_sphinx (audio)
    audio_capt = rec.recognize_google(audio, language='pt')

    time = datetime.now()
    print('respondendo...: ' + str(time) + '\n')

except sr.UnknownValueError:
    voz.say("Desculpe. Eu nao consegui compreender...")
    voz.runAndWait()
    n += 1

```

```

        continue

    except sr.RequestError as e:
        voz.say("Desculpe. Mas estamos sem conexao no momento...")
        voz.runAndWait()
        n += 1
        continue

    except (KeyboardInterrupt, EOFError, SystemExit):
        voz.say("Desculpe. Identifiquei um desligamento
inesperado.")
        voz.runAndWait()
        print("\n\n||||| DESLIGANDO ÍSIS - PROTÓTIPO COMPUTACIONAL
MAKER ||||| \n||||| Andre Medeiros ")
        exit()

#imprime o que foi dito
print ("Ser humano diz ... -> " + audio_capt + "\n")

#upper pra facilitar os ifs
audio_capt = audio_capt.upper()

###
###tratamento de ações executadas pelo protótipo, pode-se incluir
quantas desejar
###

#verifica o que foi dito e executa ação em WINDOWS (você pode
definir palavras chaves pra suas próprias ações e também outros sistemas operacionais)
if (audio_capt.find('DESLIGAR') >= 0):
    print("\n\n||||| DESLIGANDO ÍSIS - PROTÓTIPO COMPUTACIONAL
MAKER ||||| \n||||| Andre Medeiros ") # + str(n)+ "\n")
    voz.say("Desligando...")
    voz.runAndWait()

    #fim dialogo
    exit()

if (audio_capt.find('YOUTUBE') >= 0):

```

```

        busca = audio_capt[audio_capt.find('YOUTUBE')+8:]
        print ("\nAção executada: Procurando videos com o tema...
" + busca + "\n")

        #abre o caminho/path

        os.startfile("https://www.youtube.com/results?search_query=" + busca)
        action = 1

        if (audio_capt.find('OUTLOOK') >= 0):
            print ("\nAção executada: Abrindo e-mail...\n")
            os.startfile("Outlook.exe")
            action = 1

        if (audio_capt.find('GMAIL') >= 0):
            print ("\nAção executada: Abrindo Gmail...\n")
            os.startfile("http://gmail.com")
            action = 1

        if (audio_capt.find('CBN') >= 0):
            print ("\nAção executada: Abrindo radio on-line...\n")

            os.startfile("http://cbn.globoaudio.globo.com/servicos/estudio-ao-
vivo/ESTUDIO-AO-VIVO.htm?praca=SP")
            action = 1

        if (audio_capt.find('PESQUIS') >= 0):

            busca = audio_capt[audio_capt.find('PESQUIS') + 9:]
            print ("\nAção executada: Pesquisando no Google sobre... "
+ busca + "\n")

            #abre o caminho invocando o browser padrao
            os.startfile("https://www.google.com.br/search?hl=pt-
BR&q=" + busca)
            action = 1

        if (audio_capt.find('TEMPO') >= 0) or (audio_capt.find('CLIMA')
>= 0):

            print ("\nAção executada: Verificando as condições
climáticas...\n" )

```

```

#abre o caminho invocando o browser padrao
os.startfile("https://www.ipmetradar.com.br/")

action = 1

if (audio_capt.find('ATIVAR CAMERA') >= 0) or
(audio_capt.find('ATIVAR CÂMERA') >= 0) or (audio_capt.find('ATIVAR A CÂMERA') >= 0):

    voz.say("Ativando câmera para identificar objetos")
    voz.runAndWait()

    ###inicia camera
    cv()
    action = 1

if (audio_capt.find('DIGITE') >= 0):
    repetir = audio_capt[audio_capt.find('DIGITE')+6:]

    print ("\nAção executada: Repetindo a frase ..." + repetir
+ "\n" )

    ### TESTE - automação com pywinauto - escrever o que foi
dito pelo humano num arquivo .txt
    app = Application().start("notepad.exe")

    # 'digita' o texto capturado pelo audio
    app.UntitledNotepad.Edit.type_keys(repetir, with_spaces =
True)

    ###-----

    action = 1

if action == 0: # se nao for uma ação ela processa uma resposta
    #a robo responde (processo array com palavras já treinadas
e ouvidas) - para reiniciar vocabulário, basta apagar arquivo dbsqlite
    resposta = bot.get_response(audio_capt)
    print ("ISIS responde ... -> " + str(resposta) + "\n")
    voz.say(resposta)
    voz.runAndWait()

```

```
n += 1 #contador  
action = 0 #verifica acoes executadas
```

Com este software de código relativamente pequeno e simples, e custo baixíssimo de desenvolvimento e processamento, é possível desenvolver soluções muito mais elaboradas e convertê-las inclusive em oportunidades de negócios concretas, dependendo do nível de dedicação do profissional e/ou pesquisador, além do desenvolvimento de novas competências e compreensões sobre o momento e contexto que vive-se, conforme relatado anteriormente no decorrer deste trabalho.