

## Utilização de inteligência artificial no reconhecimento de voz para solicitação de ajuda

Graziela Cristina de Castro<sup>1</sup>, Leandro Marinho de Oliveira<sup>1</sup>, Rodrigo de Oliveira Plotze<sup>1</sup>

<sup>1</sup>Faculdade de Tecnologia de FATEC Ribeirão Preto (FATEC)

Ribeirão Preto, SP – Brasil

[gra.c.de.castro@gmail.com](mailto:gra.c.de.castro@gmail.com), [lmarinho1991@gmail.com](mailto:lmarinho1991@gmail.com),

[rodrigoplotze@gmail.com](mailto:rodrigoplotze@gmail.com)

**Resumo.** *Este artigo tem como objetivo demonstrar a fundamentação teórica, as ferramentas e a estrutura do projeto de um aplicativo que utiliza a inteligência artificial no reconhecimento de voz para a solicitação de ajuda.*

**Abstract.** *This article aims to demonstrate the theoretical foundation, the tools and the project structure of an application that uses artificial intelligence in voice recognition to request for help.*

### 1. Introdução

A história da inteligência artificial é confundida com a história da computação, pois, Alan Turing, que é considerado o pai da computação, foi um dos pioneiros na área da inteligência artificial (IA), ao introduzir o teste de Turing em seu artigo, de 1950, *Computing Machinery and Intelligence*, teste que questiona a capacidade da máquina digital em replicar as ações humanas, questionando se a máquina é capaz de pensar e interagir (ANDRADE, 2010). O teste consiste em uma pessoa interagindo com uma máquina, sem saber, e, caso a pessoa não perceba, pode-se considerar que a máquina pensa por si só e que passou no teste de Turing.

Logo após a Segunda Guerra Mundial, em uma conferência no campus do Dartmouth College, em 1956, se iniciava o campo de pesquisa em inteligência artificial. “Cada aspecto de aprendizado ou de outra forma de inteligência pode ser descrita de forma tão precisa que uma máquina pode ser criada para simular isso” (MCCARTHY, 1955, P.2).

Em meados dos anos 70 até o começo dos anos 80, houve um período sombrio conhecido como inverno da inteligência artificial, uma era de poucas novidades, cortes nos investimentos e baixa atenção ao setor, devido aos repetidos fracassos em levar para a prática boas ideias do papel (TECMUNDO,2018).

Em 1997 ocorreu um grande avanço na área quando a máquina derrotou o campeão soviético Garry Kasparov em um jogo de xadrez. O Deep Blue, supercomputador da IBM que conseguiu essa vitória, utilizava um método de cálculo de força bruta, analisando as possibilidades, prevendo as respostas e sugerindo os movimentos, sendo aprovado no teste de Turing (TECMUNDO,2018).

De acordo com IBM (2020), inteligência artificial (IA) é uma tecnologia que usa um mecanismo de processamento que simula a inteligência humana e é programada através de softwares com capacidades de desenvolver raciocínios e tomar decisões por conta própria.

A necessidade de segurança por parte das pessoas, principalmente idosos e com necessidades especiais, traz a possibilidade de criação de um aplicativo que auxilia em situações de riscos que utiliza a IA para reconhecer a voz e solicitar ajuda.

## 2. Fundamentação teórica

Com o avanço das tecnologias, os desenvolvedores de softwares necessitam buscar novos recursos que possam suprir as necessidades dos usuários, que estão cada vez mais complexas e exigentes. Pode-se citar comandos de voz, localização, entre outros, que se tornaram essenciais no auxílio a pessoas que desconhecem a tecnologia ou que possuem alguma necessidade especial. Também é importante que o software possua uma interface inteligente e intuitiva para que os usuários consigam utilizar sem muitas dificuldades.

Para que recursos de voz, recursos de localização e outros recursos essenciais sejam incrementados no sistema, é necessário a utilização de API's (Application Programming Interface, que em português é Interface de Programação de Aplicação) no software, que são rotinas e padrões de programação que acessam aplicativos de software e tem como objetivo a troca de informações entre dois ou mais sistemas para facilitar a realização do que foi solicitado pelo usuário (DEBATIN ET AL, 2017).

A capacidade de um computador interpretar a mesma linguagem utilizada pelos seres humanos é denominada de Processo de Linguagem Natural (PLN), que consiste no processamento de objetos de natureza linguística, situando-se como uma subárea da Inteligência Artificial (IA), que estuda a capacidade e limitações de entendimento da linguagem humana por parte do computador (FINATTO ET AL, 2015).

Segundo Debatin et al (2017) “O processamento da Linguagem Natural auxilia na manipulação que é necessária para o uso das tecnologias utilizadas para melhorar a interação humana com computadores”.

O PLN está direcionado a três aspectos da comunicação em linguagem natural, sendo esses aspectos baseados em cinco níveis de processamento: O som, que através da fonologia, são detectados os sons que compõem as palavras, a estrutura orientada pela morfologia e a sintaxe que examinam como as palavras de uma frase se desagregam em componentes e como é afetado o estado gramatical delas para que seja analisada a estrutura da sentença, e o significado que canaliza a semântica e a pragmática, no intuito de identificar e associar significado à estrutura sintática, visando eliminar resultados equívocos do sentido da estrutura (FINATTO ET AL, 2015).

O objetivo final do processamento de linguagem natural seria ter um sistema com conhecimento de mundo o suficiente para ser capaz de se envolver em uma discussão com humanos sobre qualquer assunto (COPPIN, 2010).

Acessar recursos de software por meio da voz é cada vez mais útil e colabora para a inclusão social de pessoas com necessidades especiais, pessoas com deficiência visual e pessoas com grau de escolaridade baixo, entre outros.

O PLN é responsável por auxiliar na preparação para utilizar de tecnologias de voz, tecnologias que são divididas em duas classes: reconhecimento de voz e sintetizador de voz.

O reconhecimento de voz tem como objetivo converter a língua falada em texto, ou seja, reconhece a linguagem natural dos usuários (DEBATIN ET AL, 2017). Para que o reconhecimento de voz seja feito, o software digitaliza o que foi dito pelo usuário, convertendo as vibrações captadas, criadas pela voz, em dados digitais por meio de um conversor analógico-digital, filtrando o som digitalizado para que ruídos e interferências sejam separados. O som é sincronizado, para comparar com modelos de som que estão gravados na memória do classificador, e separados em partes menores, para que seja aplicado um método de busca associando as saídas com padrões de palavras e de voz armazenados em

seu banco de dados. Posteriormente é feita a comparação com frases ou palavras semelhantes, o resultado é analisado e é realizada a funcionalidade almejada (FERREIRA ET AL, 2016).

O objetivo de um sintetizador de voz, é converter o texto escrito em linguagem falada, ou seja, transformar o conteúdo do texto em áudio. O sintetizador recebe o texto na forma digital e transforma-o em ondas sonoras (DEBATIN ET AL, 2017). Utilizado na passagem dos caracteres de texto para sons similares a linguagem natural, o sintetizador é responsável pela reprodução das cordas vocais humanas com a finalidade de emitir sons.

Na Figura 1 está exemplificado o texto em Linguagem Natural que, posteriormente, é processado por regras de mapeamento e correlações de palavras previamente elaboradas, a fim de identificar possíveis modelos ou fragmentos de processo, visando a obtenção do texto orientado a processo.



**Figura 1** Abordagem de geração de texto orientado a processo a partir de texto em Linguagem Natural.

Fonte: FERREIRA (2016)

### 3. Materiais e métodos

Como inspiração e fontes de pesquisas, serão utilizados materiais que abordem o tema de inteligência artificial, processamento de linguagem natural, tecnologia, assistentes virtuais e informações sobre aplicativos similares.

Tem como objeto de estudo idosos, pessoas com necessidades especiais e demais usuários que tenham interesse em segurança pessoal com auxílio de inteligência artificial.

O aplicativo para dispositivos móveis será desenvolvido utilizando a IDE Android Studio, com a linguagem de programação Java e integrado ao banco de dados MySQL.

A interface com o usuário tem como principal característica a usabilidade, para que o sistema seja eficiente, intuitivo e adaptado ao usuário com o uso do comando de voz.

#### 3.1. Uso de API's

A API é um conjunto de rotinas e padrões de programação usados no desenvolvimento de softwares, que se comunicam com outros serviços, interligando múltiplas funções e aplicativos baseados na Internet (DEBATIN ET AL, 2017).

Para o correto funcionamento do aplicativo é necessário o uso de uma API que possua recursos de voz e recursos de localização. Neste âmbito, seguem algumas API's que possuem recursos de voz e que podem ser úteis no desenvolvimento do presente projeto.

A Java Speech (JSAPI) é uma API que possui o reconhecimento de voz e o sintetizador de voz, sendo uma interface de programação de aplicativo. Inicialmente

direcionada a plataforma Java, possui suporte multiplataforma de reconhecedores de comando e controle, sistema de ditado e sintetizadores de voz.

O Google Cloud Speech to text não possui sintetizador de voz, ou seja, ele apenas faz o reconhecimento da voz e o transforma em texto e o seu acesso ainda é limitado pelo Google por meio de seu site de desenvolvedor.

Segundo Google (2020), no reconhecimento de fala avançado com o Cloud Speech-to-Text, os desenvolvedores converte áudio em texto ao aplicar modelos de rede neural avançados em uma API fácil de usar, usando machine learning e ferramentas de comando de voz compartilhadas nos produtos Google Search, Google Now, entre outros. A API reconhece 120 idiomas e variantes para oferecer suporte à sua base de usuários global. Além disso, essa API processa streaming em tempo real ou áudio pré-gravado, usando a tecnologia de machine learning do Google.

Segundo a Microsoft (2020), a Speech API do Bing fornece algoritmos de ponta para processar a linguagem falada e os desenvolvedores podem incluir facilmente a capacidade de adicionar ações orientadas por fala aos seus aplicativos. Em certos casos, as APIs também permitem a interação em tempo real com o usuário.

API IBM Watson Developer Cloud é um supercomputador da IBM. Uma plataforma de serviços cognitivos que adquire conhecimento conforme recebe informações, processo similar ao utilizado pela mente humana, a cognição. O Watson é a integração de várias API's que utilizam soluções com Inteligência Artificial e segurança. Uma dessas API's é a que realiza o reconhecimento e a sintetização de voz, possuindo suporte ao português do Brasil e liberada gratuitamente pela IBM. Existe ainda a capacidade de controlar a pronúncia de palavras específicas, o que é um diferencial (IBM, 2020).

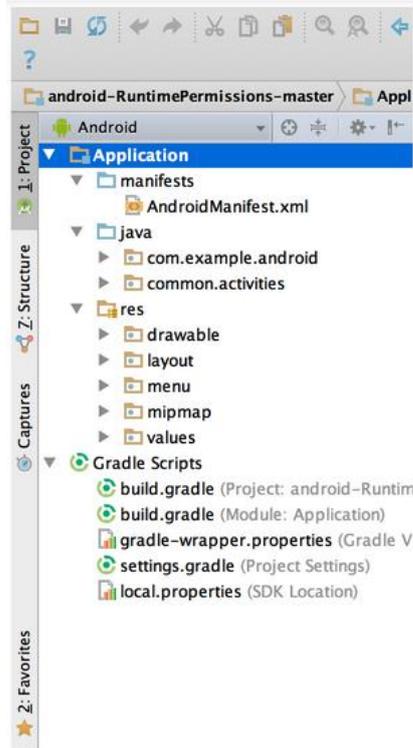
### **3.2. IDE Android Studio**

Uma IDE (Integrated Development Environment ou Ambiente de Desenvolvimento Integrado), é um software desenvolvido para auxiliar os desenvolvedores, facilitando o desenvolvimento de novos softwares. A IDE Android Studio possui características e ferramentas que dão apoio e agilizam o desenvolvimento destes novos softwares. É na IDE que são carregadas as API's (ANDROID, 2020).

O Android Studio é a IDE oficial para o desenvolvimento de apps para Android e é baseado no IntelliJ IDEA (link em inglês). Além do editor de código e das ferramentas de desenvolvedor avançadas do IntelliJ, o Android Studio oferece ainda mais recursos para aumentar sua produtividade na compilação de apps Android.

O Android Studio é de fácil assimilação, possuindo organização na estrutura do projeto. Contém arquivos de código-fonte contidos em módulos e no projeto pode haver um ou mais módulos de arquivos, podendo ser módulos de apps Android, módulos de biblioteca, módulos do Google app, entre outros módulos (ANDROID, 2020).

Para visualizar a estrutura dos arquivos do projeto no Android Studio basta clicar na aba "Project". Nessa estrutura, como mostra a Figura 2, os arquivos são organizados em pastas de acordo com seus respectivos tipos, com a finalidade de tornar simples a navegação do desenvolvedor.



**Figura 2** Estrutura dos arquivos do projeto visualizados no Android Studio.

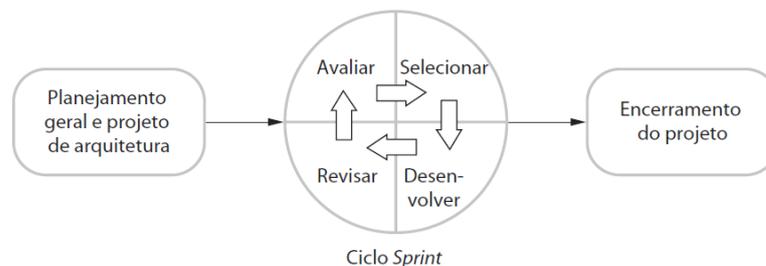
Fonte: ANDROID (2020)

O Android (2020) dispõe de informações mais detalhadas de uso do Android Studio e suas funcionalidades, possuindo muitos recursos e ferramentas com diversas funcionalidades.

### 3.3. Metodologia Ágil Scrum

Os processos de desenvolvimento rápido de software são concebidos para produzir, rapidamente, softwares úteis. O software não é desenvolvido como uma única unidade, mas como uma série de incrementos, onde cada incremento inclui uma nova funcionalidade do sistema (SOMMERVILLE, 2013).

No Scrum, existem três fases como mostra a Figura 3. A primeira é uma fase de planejamento geral, em que se estabelecem os objetivos gerais do projeto e da arquitetura do software. Em seguida, ocorre uma série de ciclos de sprint, sendo que cada ciclo desenvolve um incremento do sistema. Finalmente, a última fase do projeto encerra o projeto, completa a documentação exigida, como quadros de ajuda do sistema e manuais do usuário, e avalia as lições aprendidas com o projeto (SOMMERVILLE, 2013).



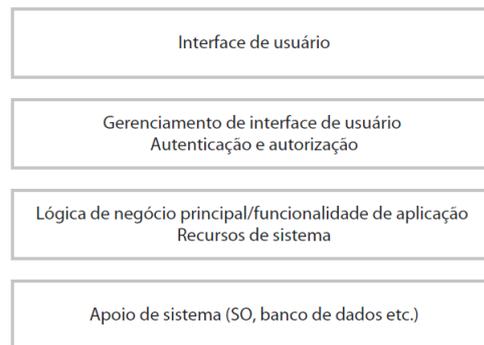
**Figura 3** O processo Scrum.

Fonte: SOMMERVILLE (2013)

### 3.4. Arquitetura de Sistema em Camadas

O projeto de arquitetura identifica os principais componentes estruturais de um sistema e os relacionamentos entre eles. O resultado do processo de projeto de arquitetura é um modelo de arquitetura que descreve como o sistema está organizado em um conjunto de componentes de comunicação (SOMMERVILLE, 2013).

A arquitetura em camadas, como mostra a Figura 4, organiza o sistema em camadas com a funcionalidade relacionada associada a cada camada. Uma camada fornece serviços à camada acima dela; assim, os níveis mais baixos de camadas representam os principais serviços suscetíveis de serem usados em todo o sistema (SOMMERVILLE, 2013).



**Figura 4 Arquitetura genérica em camadas.**

**Fonte: SOMMERVILLE (2013)**

### 3.5. Requisitos do Sistema

#### 3.5.1. Requisitos Funcionais (RF)

Requisitos com as principais funções do aplicativo.

Função para cadastrar usuários como nome, idade, data de nascimento, tipo sanguíneo, endereço e se possui alguma alergia;

Função para cadastrar contatos, inserir, editar ou excluir números de telefone de pessoas que serão avisadas em caso de emergência;

Função para envio e recebimento de mensagens para enviar mensagem aos contatos cadastrados;

Função para confirmação de recebimento de mensagens: Essa função servirá para que o sistema tenha uma confirmação de que o contato preferencial recebeu a mensagem e que possa socorrer o usuário, por meio de uma mensagem feedback.

#### 3.5.2. Requisitos Não-Funcionais (RN)

Requisitos para otimizar o uso do aplicativo.

Limite do tamanho da foto (Mb) para limitar o tamanho da foto do usuário no cadastro;

Interface gráfica simples e intuitiva para facilitar o uso do aplicativo, realizando cadastro de contatos e personalização do aplicativo;

Limite de 60s para receber *feedback* das mensagens enviadas.

### 3.5.3. Requisitos de Restrições (RR)

Requisitos para utilizar os recursos do aplicativo.

Utilizar banco de dados MySQL para salvar de maneira persistente os dados do usuário, seus contatos sejam salvos e mensagens;

O sistema deverá ser implementado utilizando serviço Mobile que permite a utilização de API's e a instalação em celulares e smartwatches.

## 3.6. Modelagem do Sistema

### 3.6.1. Modelo de Processo: Diagramas de Atividades

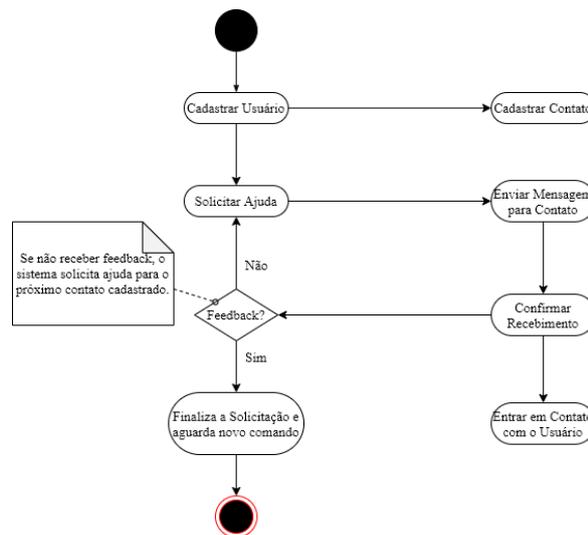


Figura 5 Diagrama de Atividades

### 3.6.2. Modelo de Interação: Diagramas de Casos de Uso

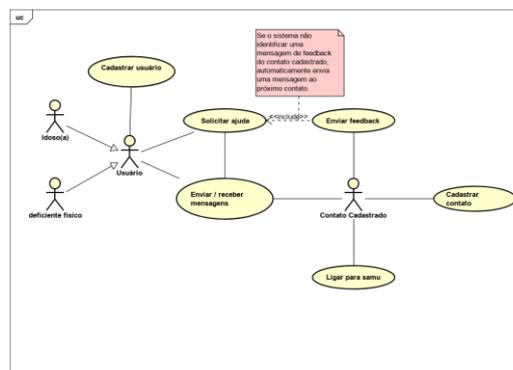


Figura 6 Diagrama de Casos de Uso



## 5. Conclusão

Não há uma conclusão definitiva, o presente projeto está em fase de prototipação.

## 6. Referências

- ANDRADE, Maria M. Introdução à Metodologia do Trabalho Científico. 10. Ed. São Paulo: Atlas, 2010.
- ANDROID. Conheça o Android Studio. Disponível em: <<https://developer.android.com/studio/intro?hl=pt-br>>. Acesso em 15 março 2020.
- COPPIN, Ben. Inteligência Artificial. 1.ed. Rio de Janeiro, 2010.
- DEBATIN, Lucas, FILHO, Aluizio H., NAU, Jonathan, ZANCHETT, Pedro S., CORREIA, Wagner. Implementação de API para Reconhecimento e Sintetização de Voz em um Aplicativo Móvel. In XIII BRAZILIAN SYMPOSIUM ON INFORMATION SYSTEMS, 5-8 junho 2017, Minas Gerais.
- DEBATIN, Lucas, FILHO, Aluizio H., DAZZI, Rudimar L. S. O Problema do Reconhecimento de Voz Offline em Dispositivos Móveis: em Busca de uma Abordagem Racional. In XXIII SIMPÓSIO BRASILEIRO DE SISTEMAS MULTIMÍDIAS E WEB: WORKSHOPS E PÔSTERES, 2017, Porto Alegre.
- FINATTO, Maria J., LOPES, Lucilene, & SILVA, Alena C. Processamento de Linguagem Natural, Linguística de Corpus e Estudos Linguísticos: uma parceria bem-sucedida. DOMÍNIOS DE LINGU@GEM. Uberlândia, MG, v. 9, n. 5, p. 41-59, dez 2015.
- FERREIRA, Renato C. B., THOM, Lucinéia H. Uma Abordagem Para Gerar Texto Orientado a Processo a partir de Texto em Linguagem Natural. In XII BRAZILIAN SYMPOSIUM ON INFORMATION SYSTEMS, 17-20 maio 2016, Florianópolis.
- GOOGLE. Google Speech-to-Text. Disponível em: <<https://cloud.google.com/speech-to-text>>. Acesso em: 4 de março de 2020.
- IBM. IBM Watson. Disponível em: <<https://www.ibm.com/watson/developer>>. Acesso em 26 de fevereiro de 2020.
- IBM. Libere o poder da IA com o IBM Watson. Disponível em: <<https://www.ibm.com/IBM/Cloud>>. Acesso em 10 de março de 2020.
- MCCARTHY, John, MINSKY, M. L., ROCHESTER, Nathaniel, SHANNON, C.E. A proposal for the Dartmouth summer research project on artificial intelligence. Dartmouth College, Hanover-NH 1955.
- MICROSOFT. Preço dos serviços Cognitivos – API de fala do Bing. Disponível em: <<https://azure.microsoft.com/pt-br/pricing/details/cognitive-services/speech-api/>>. Acesso em 8 de março de 2020.
- SOMMERVILLE, I. Engenharia de Software. 9.ed. São Paulo: Pearson Education, 2013.
- TECMUNDO. A história da inteligência artificial. Disponível em: <<https://www.tecmundo.com.br/mercado/135413-historia-inteligencia-artificial-video.htm>>. Acesso em 12 de maio de 2020.