Mestrado em Informática e Sistemas

Integração de Reconhecimento, Síntese e Gravação de Voz no Sistema de Tratamento de Atas

Relatório de Estágio apresentado para a obtenção do grau de Mestre em Informática e Sistemas
Especialização em Desenvolvimento de Software

Autor
Bruno Miguel Fernandes de Oliveira

Orientadores
Doutora Teresa Rocha
DEIS-ISEC

Mestre Ivo Santos
Software Developer
AIRC

Coimbra, Dezembro de 2018

Resumo

O estágio aludido no presente relatório realizou-se na AIRC (Associação de Informática da região Centro), e teve como principais objetivos, a investigação e integração de soluções para reconhecimento, síntese e gravação de voz numa das aplicações da empresa, o STA (Sistema de Tratamento de Atas).

O objetivo inicial pretendido neste estágio passava por facilitar a transcrição de reuniões de órgãos deliberativos realizadas verbalmente para texto. Porém, devido a determinados fatores que inviabilizavam este contexto, mais concretamente a realidade dos sistemas sonoros das assembleias que não satisfazem os requisitos para a implementação deste tipo de sistemas e a natureza habitual de debate nestas reuniões, dificultando a deteção e transcrição das vozes de todos os intervenientes, a equipa conjuntamente com o *Product Owner*, decidiu alterar o âmbito inicial do estágio para objetivos mais viáveis, que igualmente acrescentariam valor ao sistema. Sendo estes, a integração de um sistema de interação com o STA através de comandos de voz, a síntese de textos apresentados pelo mesmo e a possibilidade de efetuar gravações sonoras das reuniões mencionadas inicialmente.

Posto isto, durante o percurso do estagiário na empresa, o mesmo foi encarregue de conduzir uma investigação sobre os temas propostos e de escolher as ferramentas e tecnologias que melhor se adaptavam à realidade do sistema a integrar. Deste modo, foram realizadas várias provas de conceito, que permitiram estudar e testar diversas abordagens de integração destes tipos de tecnologia no sistema pretendido. Foi testado o reconhecimento de voz recorrendo à ferramenta Web Speech API, a síntese de voz recorrendo à *framework* Voice RSS e a gravação sonora utilizando a biblioteca JAVE numa aplicação Java que comunica com o STA via *websockets*.

Como resultado, integrou-se no STA um sistema de interação com a aplicação através do reconhecimento de comandos por voz, um sistema de gravação sonora destinado à gravação das reuniões e deixou-se uma porta aberta à integração de um sistema de síntese de voz.

Palavras Chave: Reconhecimento de Voz, Síntese de Voz, Gravação Sonora, *Web Speech API*

Abstract

The internship referred in this report was held at AIRC (Associação de Informática da Região Centro). Its main objectives were to research speech recognition, synthesis and recording solutions in order to integrate those technologies in one of the company's solutions.

The initial objective intended at this internship was to transcribe the audio obtained in the meetings to text. However, there were some issues that made this difficult to apply on real circumstances: most of the assemblies meeting rooms does not meet the requirements in order to apply these systems and, usually these meetings involve some discussion which implies some overlapping of the attendees' voices, whichever the speech recognition system will not be able to successfully transcribe. Regarding this, the team along with the Product Owner decided to change the initial scope of this internship to more viable goals: The integration of a voice interacting system with the STA system, a speech synthesis system that reads the STA texts and an integrated recording system to record the meetings' audio.

During the internship, the trainee was in charge of research on the proposing subjects and choosing the technologies and tools that best fit the reality of the system to be integrated. In this way, several proofs of concept were made, which allowed studying different integration approaches of this technology in the intended system. Thus, for the speech recognition system was used Web Speech API tool, Voice RSS framework to the speech synthesis system and, for the voice recording system, the JAVE library on a Java application which communicates with STA system through WebSockets.

As a result, the trainee integrates a speech recognition system intended to interact with the application through voice commands, as well as a sound recording system that allows recording the meetings. The research and developments made, also left an open door to a future integration of a speech synthesis system.

Keywords: Speech Recognition, Speech Synthesis, Sound Recording, Web Speech API

Agradecimentos

À Professora Doutora Teresa Rocha pelos conselhos, acompanhamento e orientação prestada durante e após a duração do estágio.

Ao orientador da AIRC, Mestre Ivo Santos pela sua ajuda e disponibilidade constantes e pelos conselhos transmitidos durante todas as fases do estágio.

A todos os meus colegas da equipa de desenvolvimento e fora da mesma, que me apoiaram e ajudaram na integração no ambiente profissional.

Ao Diretor de Desenvolvimento da AIRC, Jorge Coimbra.

Aos meus pais, irmã e avós pelo incentivo e pela força que me transmitiram não só nesta etapa, mas em todas as que a precederam.

À minha namorada Micaela, e já companheira há quase metade da minha vida, pela motivação, pela força e pelo amor.

Ao meu amigo de infância e colega de curso, Bruno Monteiro.

A todos os meus amigos.

Índice

Lista de Figuras

Lista de Tabelas

Definições e Acrónimos

AFI – Alfabeto Fonético Internacional

AIRC – Associação de Informática da Região Centro

API – Application Programming Interface

CSS – Cascading Style Sheets

DARPA – Defense Advanced Research Projects Agency

EUA – Estados Unidos da América

GPS – Global Positioning System

HTML – HyperText Markup Language

IBM – International Business Machines

IDE – Integrated Development Environment

IPC – Instituto Politécnico de Coimbra

ISEC – Instituto Superior de Engenharia de Coimbra

IVR - Interactive Voice Response

iOS – iProducts Operating System

JAVE – Java Audio Video Encoder

JSON – JavaScript Object Notation

JNLP – Java Network Launch Protocol

MIS – Mestrado em Informática e Sistemas

MIT – Massachusetts Institute of Technology

MP3 – MPEG-1/2 Audio Layer 3

NVP – Network Voice Protocol

OCR - Optical Character Recognition

PHP – PHP: Hypertext Preprocessor (originalmente Personal Home Page)

SMS – Short Message Service

SSML – Speech Synthesis Markup Language

STA – Sistema de Tratamentos de Atas

SUR – Speech Understanding Research

UI – User Interface

VODER – Voice Operating Demonstrator

W3C – World Wide Web Consortium

WAV – Forma abreviada de WAVE ou WAVEform (formato de áudio padrão eleito pela *Microsoft* e IBM para armazenamento de áudio em computadores)

XML – eXtensive Markup Language

1 Introdução

A voz é o principal meio de comunicação entre as pessoas. Nas últimas décadas tem-se descoberto novas tecnologias e novas utilidades para usufruto ou expansão das capacidades deste meio. Desde o simples armazenamento de arquivos de voz, música e entretenimento, até à sua utilização aliada à inteligência artificial para criar sistemas que "nos entendam", esta capacidade, até agora exclusiva do homem, está cada vez mais perto de se tornar um dos principais meios de interação com a tecnologia que nos rodeia.

Com isto em mente, e sempre à procura de inovação, a entidade acolhedora do presente estágio, AIRC, pretendia trazer às suas aplicações este tipo de interação. Para tal, e dado que não existia ainda nenhum desenvolvimento a nível interno relativamente a este tipo de tecnologias, foi elaborada uma proposta de estágio que tinha como objetivo a investigação e desenvolvimento de uma solução que permitisse dotar uma das aplicações da empresa com este tipo de sistemas.

A aplicação-destino do sistema desenvolvido durante o estágio, foi o Sistema de Tratamento de Atas (STA), um sistema que integra a solução ERP AIRC, e que permite elaborar Atas, Minutas, Ordens de trabalho e Certidões das reuniões da Câmara ou Assembleia municipal.

Esta aplicação foi alvo de uma reformulação visual durante o estágio, tendo sido alterada a orientação gráfica baseada no sistema *Windows* para sistemas *web*, sendo que as decisões do estagiário quanto às ferramentas e tecnologias a adotar durante o desenvolvimento dos sistemas mencionados, foi condicionada pela compatibilidade com o novo sistema *web* utilizado pelo STA.

1.1 Enquadramento do Estágio

O presente estágio foi realizado no âmbito da unidade curricular de Estágio, do Mestrado em Informática e Sistemas (MIS), ramo de Desenvolvimento de Software, do Instituto Superior de Engenharia de Coimbra (ISEC), tendo sido realizado nas instalações da entidade acolhedora, a Associação de Informática da Região Centro (AIRC), sedeadas em Coimbra, tendo decorrido no período entre dezembro de 2015 e agosto de 2016.

1.1.1 ISEC

O ISEC é uma instituição que faz parte do Instituto Politécnico de Coimbra (IPC) e tem como missão a criação, transmissão e difusão da cultura, ciência e tecnologia, ministrando uma formação de nível superior para capacitar os seus alunos para o exercício das atividades profissionais no sector da engenharia e promover o desenvolvimento da região onde se insere.

À data da elaboração do presente relatório, esta instituição contava com cerca de 2600 estudantes, inseridos em 37 cursos que se subdividem por licenciaturas, mestrados e cursos técnicos superiores profissionais.

1.1.2 MIS

O Mestrado em Informática e Sistemas é um curso de especialização superior, lecionado no ISEC e tem por objetivo a formação de mestres em Informática e Sistemas capazes de exercerem a sua atividade profissional com um elevado nível de competência técnica, científica e profissional.

Este curso tem disponíveis duas especializações que consistem em:

Desenvolvimento de Software

Tecnologias da Informação e do Conhecimento

A primeira destas especializações (na qual se insere o estagiário) propõe-se formar profissionais competentes no domínio de desenvolvimento de software nas várias etapas que o compõem, capazes de lidar com aspetos de gestão de projetos, equipas e garantia de qualidade.

1.1.3 AIRC

A AIRC é uma empresa fundada por 30 municípios da região centro, que faz parte do setor das sociedades não financeiras públicas, e tem como principal atividade a produção de *software* e fornecimento de produtos e serviços.

Tendo iniciado a sua atividade no ano de 1982, a AIRC tem registado um crescimento contínuo sendo que, à data da elaboração deste relatório, esta associação contava com mais de 250 clientes, dos quais 60% são municípios, o que lhe confere a liderança deste setor de mercado.

A estrutura desta empresa conta com uma equipa dinâmica, constituída por cerca de 100 profissionais com diferentes competências em investigação, desenvolvimento de *software*, consultoria, formação e assistência técnica.

Esta empresa explora um diverso número de temas, tais como gestão de juntas de freguesia, gestão comercial de águas, gestão e planeamento do território, gestão do capital humano, gestão financeira e *business intelligence*. Promove ainda a desmaterialização de documentos e processos através das suas soluções de gestão documental e a eficiência e produtividade dos seus clientes através das suas soluções de mobilidade.

Alguns dos seus produtos mais relevantes são o *MyNet*, uma solução de atendimento e relacionamento com o cidadão; o *MyDoc*, uma solução de gestão documental que permite desmaterializar documentos e processos; o *biAIRC*, uma solução que permite a recolha e análise da informação proveniente de atividades e resultados de uma organização; o ERP AIRC, onde se agrupam um conjunto de soluções integradas que permitem abranger quase a totalidade das áreas de atividade dos organismos públicos; entre outros.

1.1.4 O Sistema de Tratamento de Atas (STA)

O sistema desenvolvido durante o estágio teve como destino uma aplicação integrante do ERP AIRC, conhecida como Sistema de Tratamento de Atas (ou STA). Esta solução permite elaborar Atas, Minutas, Ordens de trabalho e Certidões das reuniões da Câmara ou Assembleia municipal. Permite ainda elaborar e consultar os assuntos tratados durante as reuniões, de modo a desmaterializar completamente todos os documentos produzidos no decorrer das mesmas. Para além disto, este sistema possibilita também o acompanhamento e realização da reunião através da votação *online*, disponibilizando, em tempo real, os pontos a serem votados por cada membro da reunião, assim como os resultados das mesmas.

Esta aplicação apresenta as seguintes funcionalidades:

- **Preparação da Reunião** – simplifica o processo de preparação da reunião ao criar o documento para a Ordem de Trabalhos.
- **Criação Automática de Documentos** – concebe documentos automaticamente, tais como certidões, ofícios, ordens de execução e editais de documentos de apoio, a partir de informações obtidas durante a reunião.
- **Envio de Ordens de Execução** – capacidade de envio por e-mail das ordens de execução aos serviços ou funcionários responsáveis pelos assuntos tratados durante a reunião.
- **Tratamento Documental** – permite controlar toda a documentação tratada durante a reunião, desde a sua criação até à sua aprovação e publicação.
- **Votações Online** – providencia o acesso à votação a cada membro com direito a voto, para que o mesmo possa transmitir o seu parecer sob cada ponto discutido no âmbito da reunião.

1.1.5 O Problema

Pretendia-se, através da realização deste estágio, que se iniciasse a exploração do tema do reconhecimento de voz na empresa e se desenvolvesse conhecimento sobre o mesmo. Para tal, e como ponto de partida para a integração de sistemas de reconhecimento de voz nas aplicações do ERP AIRC, tencionava-se dotar o STA de um sistema deste género. O principal objetivo consistia em simplificar o processo de criação de documentos provenientes das reuniões, utilizando o reconhecimento de voz como forma de obter a informação debatida durante a reunião. No entanto, devido a determinados fatores que inviabilizavam esta solução (explorados no subcapítulo seguinte), foi necessária uma alteração do âmbito.

Com esta alteração de âmbito, pretendia-se igualmente acrescentar valor à solução mencionada, resolvendo outro tipo de problemas, tais como, permitir uma maior acessibilidade e praticabilidade na utilização do STA através da utilização de comandos por voz para aceder e executar tarefas nos diferentes contextos da aplicação; permitir a sintetização para voz do conteúdo dos ecrãs, fomentando também a acessibilidade providenciada pelo STA; e, finalmente, permitir o registo sonoro da reunião, possibilitando posteriormente a sua consulta de modo a facilitar a elaboração dos documentos a ser gerados como resultado da mesma.

1.2 Objetivos do Estágio

Descreve-se de seguida os objetivos iniciais do estágio, assim como as alterações de âmbito sofridas durante o período de estágio.

1.2.1 Objetivos Iniciais

- **Análise do estado da arte de *encoders* de reconhecimento de voz em Português:** efetuar um estudo sobre as ferramentas de reconhecimento de voz existentes que suportam a língua Portuguesa.
- **Seleção do *encoder* a adotar no projeto:** com base no estudo anterior, selecionar uma das ferramentas que melhor se adeque aos requisitos impostos pelo STA.
- **Implementação de uma biblioteca de reconhecimento de voz:** com base no *encoder* adotado, implementar uma biblioteca de reconhecimento de voz (a título de exemplo: uma base de dados onde são guardadas as diversas combinações de palavras, fonemas, pronúncias, etc. – este assunto é abordado no capítulo 2).
- **Possibilidade de transcrição de áudio para texto, inserindo essa informação em dados passíveis de usar na criação de documentos base da reunião:** transcrever o áudio obtido durante a reunião para texto de modo a que este possa ser utilizado na criação dos documentos originados pela mesma.
- **Armazenamento do ficheiro de áudio em base de dados:** armazenar o ficheiro de áudio obtido no decorrer da reunião em base de dados para que possa ser acedido mais tarde para consulta.
- **Integração na solução AIRC já existente - Sistema de Tratamento de Atas:** integrar o sistema desenvolvido na solução já existente (STA) como complemento do mesmo.

1.2.2 Alterações de Âmbito

Após análise de mercado e algumas decisões provenientes do *Product Owner*, o âmbito inicial da proposta acabou por se alterar durante o período de estágio, pelo que foram alterados alguns dos objetivos inicialmente propostos e acrescentados outros.

As razões que levaram a estas alterações deveram-se essencialmente aos seguintes fatores:

- **Falta de praticabilidade da solução pretendida:** tal como é descrito no capítulo 2.1.4 deste relatório, para que o reconhecimento de voz seja efetuado com sucesso, o discurso tem que ser claro, limpo de outras vozes ou ruídos e deverão ser evitados discursos simultâneos que

baralhem o sistema. Posto isto, a natureza do ambiente onde esta solução ia ser colocada em prática não iria reunir as condições necessárias para o seu correto funcionamento. O carácter de debate que habitualmente sucede neste tipo de reuniões é sujeito regularmente a discursos simultâneos, o que dificultaria ao sistema, a tarefa de deteção e transcrição das vozes de todos os intervenientes. De notar ainda que os intervenientes por vezes, podem recorrer a linguagem não verbal, característica esta que o sistema também não seria capaz de capturar.

- **Falta de equipamento específico por parte das assembleias:** relativamente ainda ao assunto abordado no ponto anterior, eventualmente seria possível implementar uma abordagem que amplificaria o sucesso do reconhecimento de voz. No entanto, esta abordagem iria exigir que cada um dos participantes da assembleia estivesse equipado com um microfone provido de um bom isolamento sonoro, o que requereria um investimento significativo para a aquisição de uma funcionalidade que não iria funcionar no seu pleno, já que esta abordagem não iria resolver o problema dos discursos simultâneos nem da linguagem não verbal.

- **Falta de recursos/*encoders* para síntese de voz que suportem a língua Portuguesa:** é exposto no capítulo 2.1.5.8 deste relatório que a quantidade de ferramentas de desenvolvimento/bibliotecas gratuitas específicas para o reconhecimento de voz é bastante reduzida sendo que, a oferta ainda é mais reduzida ou praticamente inexistente quando se trata da situação particular do reconhecimento de voz para a língua Portuguesa.

- **Complexidade/Exequibilidade:** desenvolver uma biblioteca de raiz seria também uma tarefa bastante complexa e morosa dado à quantidade de combinações, palavras, pronúncias, etc. que seria necessário compor (estes temas são abordados no capítulo 2.1.2 deste relatório). No tempo disponível para a realização do estágio e recorrendo apenas ao trabalho do estagiário, esta tarefa seria inexequível.

Posto isto, e de modo a igualmente acrescentar valor ao sistema tendo em conta a presente temática, foram alterados os objetivos de modo a seguir uma abordagem mais viável. Objetivos estes que iam sendo corroborados durante o período de estágio através das provas de conceito que constam no capítulo 5 deste relatório.

A tabela seguinte ilustra as alterações efetuadas aos objetivos.

Tabela 1 – Alterações de âmbito da proposta de estágio

Objetivo	Estado
Análise do estado da arte de *encoders* de reconhecimento de voz em Português	Mantido
Seleção do *encoder* a adotar no projeto.	Mantido
Implementação de uma biblioteca de reconhecimento de voz	Removido
Possibilidade de transcrição de áudio para texto, inserindo essa informação em dados passíveis de usar na criação de documentos base da reunião	Removido
Armazenamento do ficheiro de áudio em base de dados	Ajustado
Integração na solução AIRC já existente - Sistema de Tratamento de Atas	Mantido
Estudo das funcionalidades da ferramenta de reconhecimento de voz adotada	Acrescentado
Implementação de um sistema de reconhecimento de voz para interpretação de comandos	Acrescentado
Análise e implementação de um sistema para síntese de voz	Acrescentado
Implementação de um sistema para gravação sonora (gravar áudio das reuniões)	Acrescentado

1.3 Estrutura do Documento

O presente relatório está organizado em diversos módulos, de acordo com a seguinte estrutura:

1. **Introdução –** Este capítulo tem como finalidade, dar a conhecer o âmbito do estágio realizado, assim como os seus objetivos e as instituições intervenientes.
2. **Estado da Arte –** Neste capítulo apresenta-se o estado da arte dos três principais temas deste estágio, o reconhecimento, síntese e gravação de voz. É descrita aqui a sua história, assim como o modo de funcionamento e outras informações relevantes para a investigação realizada.
3. **Ferramentas e Tecnologias –** São descritas neste capítulo, todas as ferramentas e tecnologias utilizadas durante o período de estágio.

4. **Metodologia de Trabalho –** Neste capítulo descreve-se a metodologia de desenvolvimento utilizada na instituição e o modo como foi adaptada à realidade do estágio, assim como os papéis desempenhados pela equipa.
5. **Desenvolvimento –** Este capítulo tem como objetivo, explicar como decorreram as diferentes fases de desenvolvimento durante o estágio. São detalhados aqui, pormenores sobre a investigação, objetivos, medidas adotadas e resultados ocorridos nas diferentes fases de desenvolvimento.
6. **Conclusões –** Apresentam-se aqui as conclusões e reflexões finais relativamente ao estágio, assim como uma visão sobre o trabalho futuro.

2 Estado da Arte

Foi proposto ao estagiário, no decorrer do período inicial do estágio, a realização de uma investigação acerca dos assuntos a abordar, nomeadamente o reconhecimento, síntese e gravação de voz, dando especial ênfase aos dois primeiros, já que estes foram o tema central do estágio. Sendo este um projeto pioneiro na empresa, esta investigação foi projetada no sentido de desenvolver conhecimento sobre os assuntos mencionados e tecnologias análogas. Era também relevante perceber o tipo de aplicações existentes que recorrem ao reconhecimento, síntese e gravação de voz e as razões que as levam a valer-se deste tipo de tecnologias.

Grande parte dessa investigação foi então convertida para o presente capítulo de Estado da Arte, cuja natureza extensa reflete-se pela forte componente de investigação inicial exigida no estágio.

2.1 Reconhecimento de Voz

Pode definir-se como reconhecimento de voz, a capacidade de um dispositivo eletrónico conseguir "entender" a fala humana. O dispositivo recebe como entrada o sinal analógico da voz através de um microfone, processa essa entrada e interpreta-a através do *software* de reconhecimento de voz, que atribuirá significado a cada som recebido, formando palavras [1].

Esta tecnologia tem evoluído exponencialmente com o passar do tempo, começando por reconhecer apenas número e sílabas simples até ser capaz de reconhecer vocabulários e dialetos complexos e até controlar diversos dispositivos eletrónicos apenas por comandos de voz (ex. *Siri, Cortana, Google Now*) [2].

2.1.1 Revisão Histórica

Os primeiros sistemas de reconhecimento de voz limitavam-se apenas ao reconhecimento de dígitos, era ainda impossível reconhecer o discurso humano dado a sua complexidade. Ainda assim, no início da década de 60 a IBM (International Business Machines), desenvolveu a "*Shoebox Machine*", considerada como o projeto precursor do reconhecimento de voz atual [3], onde já era possível, entre os 10 dígitos de 0 a 9, reconhecer algumas palavras em inglês tais como "*plus*", "*minus*", "*total*" que permitiam resolver operações aritméticas simples. Entretanto, antes da década de 70, alguns laboratórios pelo mundo iam desenvolvendo algumas máquinas para reconhecimento de voz, chegando até a ser construído numa Universidade em Inglaterra (*University College London*), um sistema capaz de reconhecer 4 vogais e 9 consoantes [4].

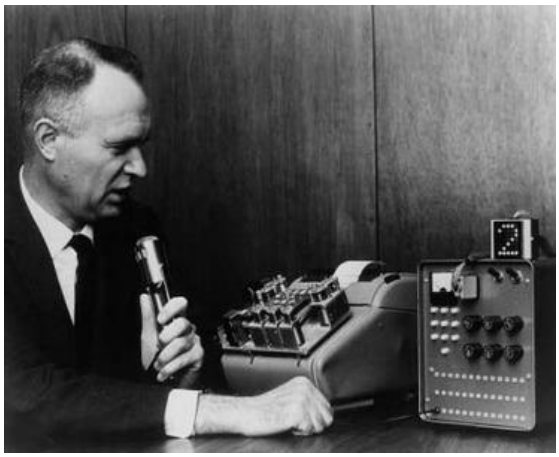


Figura 1- Shoebox Machine [120]

Na década de 70, o departamento de defesa dos Estados Unidos da América (EUA), começou a manifestar um grande interesse nas tecnologias de reconhecimento de voz. Foi então que se deu a evolução mais significativa desta tecnologia. Foi criado um programa de investigação denominado por *Speech Understanding Research* (ou SUR), sendo um dos mais largos estudos da história acerca deste tema e que deu origem ao *Harpy* – um sistema de reconhecimento de voz, capaz de "entender" 1011 palavras. Este sistema recorria a uma rede de possíveis frases, conseguindo reconhecê-las dando já alguma atenção, embora ainda um pouco rudimentar, às diferentes pronúncias ou sotaques do utilizador [4].

Na década de 80, o reconhecimento de voz passou das centenas para os milhares de palavras que eram possíveis de reconhecer. Isto devido a novas abordagens estatísticas que ficaram conhecidas como "O Modelo Oculto de Markov" (em Inglês, *Hidden Markov Model*). Este método consistia em, para além do uso habitual dos padrões de som já utilizados, considerar também a probabilidade de sons desconhecidos pelo sistema serem também palavras, ou até palavras conhecidas, mas pronunciadas de forma diferente. Graças à popularidade e ao fluxo constante de inovação e melhorias deste método, o mesmo tem tido grande impacto na tecnologia de reconhecimento de voz até aos dias de hoje. No entanto, nesta altura ainda permanecia um problema nestes sistemas de reconhecimento de voz que consistia em, a cada palavra dita, era necessário fazer uma pausa para que o sistema conseguisse separar as palavras de uma determinada frase [4].

Na década de 90 surgiram os primeiros sistemas de reconhecimento de voz comerciais onde já era possível falar de forma natural, existindo sistemas capazes de interpretar 100 palavras por minuto. No entanto estes programas requeriam algum treino por parte do utilizador e os seus preços eram absurdamente altos. Surgiram também nesta altura os primeiros menus telefónicos

ativados por voz que ainda hoje se vão utilizando em algumas centrais de atendimento telefónico [2].

Foi em 2000 que se chegou à maior taxa de acertos por parte dos sistemas de reconhecimento de voz. Nesta altura a taxa de sucesso em termos de acerto rondava os 80%, ou seja, uma média de 2 palavras erradas por cada 10, sendo o reconhecimento de voz uma realidade cada vez mais bem-sucedida [2].

No final da última década, a taxa de acerto aumentou para um valor mais próximo dos 100% [5]. No entanto, uma taxa de sucesso dessa ordem é atualmente impossível devido a diversos fatores tais como, as diferentes pronúncias de determinadas palavras, a entoação, a dificuldade do sistema em entender a pontuação e a outros fatores que se encontram mais detalhados no capítulo **2.1.3**. Desta forma, a evolução do reconhecimento de voz acabou por estagnar. Ainda assim, existem hoje em dia diversos programas e ferramentas bem-sucedidas como é o caso dos assistentes digitais embutidos nos *smartphones*, automóveis e algumas ferramentas de acessibilidade dos computadores, que conseguem entender com elevada taxa de sucesso alguns comandos ditos pelo utilizador [6].

2.1.2 Modo de Funcionamento

O reconhecimento de voz é uma das áreas mais complexas da informática dado que envolve um conjunto de tópicos matemáticos que complementam os métodos informáticos envolvidos. No geral existem 4 abordagens diferentes que se podem seguir no reconhecimento de voz, que são: **Análise de Padrões Simples** (onde cada palavra é reconhecida inteiramente sem uma análise mais profunda); **Análise de Características e Padrões** (onde cada palavra é particionada e se detetam as suas características-chave como por exemplo as vogais nela contidas); **Modelo de Linguagem e Análise Estatística** (onde o conhecimento da gramática e a probabilidade da existência de certas palavras ou sons consecutivos são usados como forma de acelerar o reconhecimento e melhorar a precisão); e as **Redes Neuronais** (Modelos computacionais que simulam as ligações sinápticas do cérebro humano que poderão reconhecer padrões de forma mais confiável, como por exemplo os sons das palavras, após algum treino exaustivo) [7].

2.1.2.1 Análise de Padrões Simples

Este método consiste na utilização de um algoritmo de correspondência de padrões (*Pattern Matching*) que analisa palavras separadamente, tal como são ditas sem efetuar nenhuma análise mais extensa. Geralmente os sistemas que recorrem a este método têm uma breve lista de entradas de palavras que irão comparar com a palavra recebida ou, se for o caso, com uma frase,

mas particionando essa frase nas várias palavras que a compõem, analisando as mesmas separadamente. Estes métodos são então limitados a um número reduzido de palavras e servem essencialmente para menus interativos em *call-centers* ou sistemas que necessitem de comandos simples. Dado o seu teor, não existe neste tipo de metodologias a necessidade do reconhecimento da estrutura da linguagem ou o significado de certas palavras, pois é irrelevante que o discurso faça sentido [7].

2.1.2.2 Análise de Características e Padrões

Este método foca-se no facto de reconhecer as palavras como um conjunto separado de tons e não como uma palavra por si só. Ou seja, esta metodologia considera um conjunto de tons que recebe como por exemplo, uma palavra ou uma frase e constrói a sua analogia a partir desta base. Na linguagem humana, existem diversos tons que constituem palavras, muitos deles fazendo parte de diferentes palavras. Portanto, ao invés do sistema criar a sua base de dados através de entradas "palavra a palavra", este cria a sua base de dados partindo do princípio que os vários tons que recebe constroem palavras. Isto evita um treino tão exaustivo do sistema, não sendo necessário treiná-lo com todas as palavras possíveis para que funcione minimamente bem.

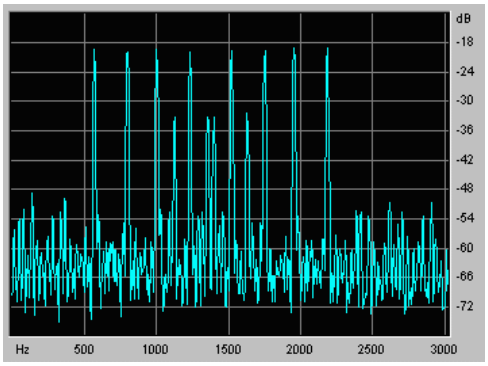


Figura 2 - Exemplo de um espectrograma de som [121]

Este método de reconhecimento de voz envolve uma série de processos que passam pela conversão do sinal analógico recebido pelo microfone para o formato digital, que é depois convertido num espectrograma de som[1], particionado em pequenas partes denominadas por "*acoustic frames*", cada uma contendo uma parte do som. De seguida, comparam-se essas partes com o dicionário fonético[2] e tenta-se identificar a palavra mais provável de ter ocorrido [7]. A maioria dos sistemas ou aplicações que recorrem a este método já permitem a introdução de *feedback* por parte do utilizador. Ou seja, permitem ao utilizador corrigir eventuais erros de

[1] Espectrograma de Som - Gráfico que mostra a intensidade do som ao longo do tempo.
[2] Dicionário Fonético - lista de palavras ou pequenas partes do som já conhecidas pelo sistema.

conversão de voz para texto, fazendo com que o sistema reaja melhor na próxima vez que a palavra for dita.

2.1.2.3 Modelo de Linguagem e Análise Estatística

Como já foi referido, reconhecer o discurso humano é bastante mais complexo do que a simples tarefa de reconhecimento de tons descrita anteriormente. Como se sabe, existem diferentes pronúncias para a mesma palavra (as pronúncias podem mudar de região para região), existem palavras que se "ouvem" da mesma forma, mas que têm diferentes significados (homónimos tais como "acerto" e "asserto", "coser" e "cozer", "acento" e "assento", etc.) que tornam a tarefa de reconhecimento mais difícil para o sistema.

Para, de certa forma, contornar parte destes problemas recorre-se a um modelo de linguagem que é um modelo estatístico onde figura a probabilidade da existência de certas sequências de palavras [8]. Por exemplo, é bastante provável que no discurso, depois da palavra "boa" venha a palavra "pessoa", "noite", "tarde". Desta forma, o sistema terá isso em conta ao analisar o discurso e fará uma predição mais acertada. A maioria dos sistemas atuais têm esta característica presente, pois recorrem ao *Modelo Oculto de Markov* (ferramenta estatística bastante utilizada na inteligência artificial que serve para modelar diversas sequências) que, de certa forma, é a base deste conceito [9].

2.1.2.4 Redes Neuronais

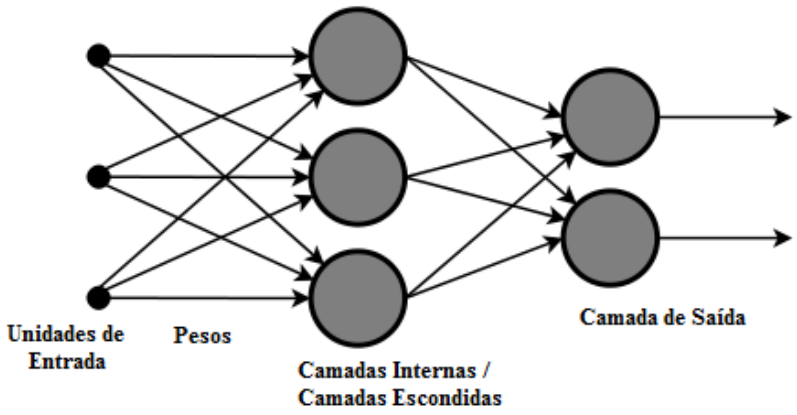


Figura 3 - Representação de uma rede neuronal

As redes neuronais são sistemas baseados no funcionamento do cérebro humano, ou seja, funcionam de forma análoga aos neurónios no cérebro humano [10]. São compostas por várias camadas de "neurónios" (ou unidades) interligadas entre si, subdivididas por 3 tipos de camadas diferentes: a **camada de unidades de entrada**, que é responsável por alimentar a rede com os dados de entrada, as **camadas internas** (ou camadas "escondidas"), que são responsáveis pela aprendizagem ou interpretação dos dados que entraram na rede e a **camada de saída** que transmite para o exterior, os resultados obtidos. As conexões entre unidades estão também sujeitas a um "peso" com um determinado significado, ou seja, tal como no funcionamento dos neurónios humanos, o peso entre as suas conexões varia em função da influência que uma unidade tem sobre a outra: quanto maior o significado numa conexão entre unidades, maior é o seu peso e vice-versa. Desta forma, é possível treinar a rede para que esta "aprenda" determinadas funções, tais como no caso do reconhecimento de voz. Para que isto seja possível é necessário que a rede esteja sujeita também a um *feedback* por parte do utilizador ou seja, o utilizador tem que de certa forma, validar o resultado da rede. Dependendo desta validação, caso a rede tenha errado na identificação de um determinado elemento, na próxima vez que esta for sujeita à mesma tarefa, a probabilidade de acerto incrementará [11].

2.1.3 Aplicações do Reconhecimento de Voz

Tal como a maioria das tecnologias, o reconhecimento de voz surgiu devido a determinadas necessidades. Hoje em dia esta tecnologia é cada vez mais utilizada em determinadas áreas e sectores de modo a facilitar a interação entre a pessoa e a máquina ou a automatizar algumas tarefas rotineiras. Enumeram-se nos parágrafos seguintes algumas das aplicações principais do reconhecimento de voz.

2.1.3.1 Reconhecimento de Voz Aplicado à Área da Saúde

O reconhecimento de voz na área da saúde é talvez das mais antigas aplicações deste tema e sem dúvida uma das que mais contribuiu para o desenvolvimento do mesmo. Uma das primeiras pessoas que pegou neste tema foi *Alexander Graham Bell* que, movido pela necessidade de fazer algo para ajudar na comunicação com a sua esposa que era deficiente auditiva, tentou desenvolver um dispositivo capaz de interpretar palavras e transformá-las em imagens visíveis (investigação esta que acabou por levar *Bell* à invenção do telefone) [12].

Hoje em dia, o reconhecimento de voz na área da saúde está bastante ligado à acessibilidade, ou seja, facilitar a utilização de diversas ferramentas ou sistemas a utilizadores com algum tipo de incapacidade física que dificulte a sua interação. Existem por exemplo, aplicações de reconhecimento de voz que permitem a um utilizador efetuar determinadas tarefas num

computador, ditando apenas alguns comandos para um microfone, incluindo também a escrita de texto, eliminando de certa forma a necessidade de interação com o teclado e/ou rato. Por outro lado, a acessibilidade no mundo do reconhecimento de voz não fica só pelo uso do computador, mas vai também ao encontro de outro tipo de necessidades. Existe por exemplo, uma aplicação para *smartphones* que facilita o início da interação com as pessoas deficientes auditivas, que consiste em enviar um alerta sob a forma de aviso sonoro (nos casos em que o nível de surdez não seja total) ou luminoso, quando chamam pelo nome do indivíduo, para que este saiba quando pretendem comunicar com ele [13].

Outro caso de utilização do reconhecimento de voz nesta área, é no meio cirúrgico, onde podem existir meios para, sob a forma de um comando de voz, durante uma cirurgia, por exemplo aceder aos registos clínicos do paciente, ditar dados relevantes para serem guardados ou até chamar outras funções, evitando assim possíveis interrupções ou perdas tempo por parte dos médicos [14].

2.1.3.2 Reconhecimento de Voz Aplicado ao Marketing e a Usos Corporativos

Como já foi referido anteriormente, o reconhecimento de voz é usado já há alguns anos a nível das centrais de atendimento telefónico, de modo a automatizar por exemplo, os processos de seleção do tipo de serviço pretendido sob a forma de menus interativos que obedecem a comandos simples por voz, diminuindo assim a mão-de-obra necessária para esse serviço e aumentando a produtividade [15].

O reconhecimento de voz é também utilizado para fins de *marketing* e, para isso existe uma técnica denominada por *audio mining*. Esta técnica consiste em procurar ocorrências de determinadas palavras ou frases num discurso guardado em ficheiro de áudio ou vídeo. É possível, por exemplo, efetuar controlo de qualidade nas chamadas em *call centers* verificando se os funcionários utilizam o discurso imposto pela empresa, ou até verificar quais os assuntos abordados mais frequentemente de modo a melhorar o serviço [16].

Em termos de utilização do reconhecimento de voz a nível corporativo, têm surgido soluções que permitem às empresas efetuar determinadas tarefas de uma forma mais automatizada, como escrever documentos ou *e-mails* através de programas de reconhecimentos de voz, tomar notas automaticamente em reuniões sem ter que recorrer à escrita manual, ou até guardar o conteúdo dessas reuniões (por exemplo, o *software Nuance Dragon Enterprise*) [17].

2.1.3.3 Reconhecimento de Voz Aplicado à Segurança

Nos últimos tempos tem-se falado do tema da segurança utilizando o reconhecimento de voz. Um dos maiores exemplos que surgem nesta temática é o caso da autenticação por voz, que

consiste em o utilizador ditar uma determinada palavra ou frase que permite ao sistema reconhecer a identidade do mesmo, evitando as habituais *passwords* e facilitando por exemplo a autenticação remota, utilizando um telefone ou um computador sem ser necessário adquirir nenhum *hardware* adicional [18].

Num outro extremo do tema da segurança, tem-se utilizado também o reconhecimento de voz para a identificação de criminosos. Através de uma base de dados com registos de voz de criminosos ou suspeitos, é possível verificar e identificar através de escutas telefónicas a pessoa que está por detrás do telefone e, a partir daí efetuar as ações necessárias [19].

Já para uso militar, tal como foi referido anteriormente, a DARPA (Defense Advanced Research Projects Agency), apostou fortemente na tecnologia de reconhecimento de voz. Esta seria uma tecnologia que, caso aplicada com sucesso, apresentaria grandes vantagens a nível militar já que seria possível, por exemplo intercetar comunicações de inimigos, identificar palavras-chave e verificar qual seria a informação relevante [20]. No entanto, dado à natureza confidencial deste tipo de informação, não existem muitos detalhes acerca da mesma, apenas se sabe que possivelmente esta tecnologia pode ser aplicada.

2.1.3.4 Reconhecimento de Voz no Quotidiano

O reconhecimento de voz já é utilizado há algum tempo em versões mais simples, como por exemplo o *voice dialer* dos antigos telemóveis em que era possível através da voz, transmitir ao telefone o nome da pessoa a quem desejávamos telefonar. Hoje em dia, nos *smartphones* já é possível mandar executar esta e muitas outras tarefas através da voz. Temos como exemplo o caso dos assistentes digitais disponíveis nos s*martphones* (*Google Now, Cortana e Siri*), onde já é possível questionar esses assistentes virtuais e obter a resposta correspondente, ou mandar executar determinada tarefa como abrir uma aplicação, enviar um *e-mail* ou SMS (*Short Message Service*) completamente escritos através da voz com bastante eficiência. Isto também veio transformar, de certo modo, a antiga definição de sistemas de mãos livres para os automóveis, não sendo necessário comprar nenhum dispositivo adicional. É possível efetuar todas as tarefas anteriores sem necessariamente ter que tirar as mãos do volante.

O reconhecimento de voz também fez surgir novas formas de sistemas de mãos livres integrados nos automóveis mais recentes: sistemas de navegação GPS (*Global Positioning System*) que obedecem a comandos por voz, ou até mesmo, o próprio automóvel estar equipado com um sistema capaz de interpretar alguns comandos para realizar determinadas funções sem ser necessário intervenção manual por parte do condutor [21].

Em termos de utilidade doméstica, o reconhecimento de voz também tem algumas vertentes. Hoje em dia, cada vez mais se ouve falar em *Smart Homes* ou casas inteligentes. Existem

já sistemas controlados por voz que permitem ligar ou desligar luzes de determinadas divisões, ligar ou desligar eletrodomésticos, ou, em sistemas mais avançados ter um assistente virtual similar ao dos *smartphones* capaz de efetuar as mesmas tarefas [22]. Ainda em termos de dispositivos domésticos, um exemplo que se tem destacado nesta matéria são as *Smart TV's* que conseguem realizar operações como por exemplo mudar de canal ou aumentar/diminuir o volume, aceder à internet, etc. apenas através de comandos de voz, sem ser necessário recorrer ao comando [23].

Finalmente temos também, à semelhança dos programas corporativos descritos anteriormente, programas para uso doméstico que permitem ao utilizador, escrever texto, navegar na internet ou efetuar determinadas operações no computador ditando apenas comandos com a voz, sem ter que recorrer ao teclado ou rato [24].

2.1.4 Problemas e Limitações no Reconhecimento de Voz

Como se sabe, o reconhecimento de voz ainda não conseguiu, por si só, chegar a uma taxa de acerto de 100%. Isto deve-se a diversos fatores que hoje em dia, apesar do grande avanço tecnológico, ainda não são passíveis de contornar.

Em cada país existem determinadas regiões que pronunciam algumas palavras de forma diferente (sotaques diferentes), o que de certa forma dificulta a tarefa de aprendizagem ou capacidade de acerto do sistema, já que existirão várias formas de representar a mesma palavra em que, algumas delas poderão ser desconhecidas do sistema. O padrão de fala também varia de pessoa para pessoa, as pausas entre palavras e as velocidades do discurso são diferentes, a entoação também é diferente, fazendo com que o sistema possa interpretar pausas longas ou outros sons que o utilizador reproduza entre palavras enquanto está a pensar, ou até omitir palavras devido a uma velocidade muito rápida no discurso.

Praticamente todas as línguas tem homónimos, ou seja, palavras que soam de forma idêntica, mas com significados diferentes que no discurso se identificam apenas através do contexto. Apesar de já existirem algumas soluções capazes de identificar o contexto (palavras que surgem juntas mais frequentemente), é ainda impossível identificar com precisão qual a palavra apropriada a transcrever. Existem também palavras ou nomes que não são tão comuns e que o sistema desconhece, o que mais uma vez dificulta na obtenção de uma boa precisão.

Também é impossível ao sistema de reconhecimento de voz, ter conhecimento sobre como empregar determinadas regras gramaticais como a pontuação, de forma autónoma. A maioria destes sistemas necessitam que o utilizador verbalize a pontuação ou intervenha manualmente.

Existe ainda o facto da eventual existência de ruído de fundo no momento que se dita o discurso ao sistema. Esse ruído pode ser interpretado como palavras e causar erros na transcrição.

É também uma tarefa bastante complicada para o sistema, decifrar conversas entre vários utilizadores. Para além de existirem na linguagem humana modos de comunicação não-verbais que obviamente não são reconhecidos pelo sistema (tais como o acenar de cabeça, gestos, etc.), existe também o facto de ser comum os intervenientes interromperem-se ou falarem simultaneamente, o que dificulta a tarefa de perceber qual dos intervenientes está a falar em determinado momento e interpretar interrupções ou discursos simultâneos. Neste tipo de sistemas, ainda é necessário efetuar um discurso claro para que este possa ser interpretado com uma maior taxa de sucesso [25].

Uma das limitações mais relevantes no reconhecimento de voz prende-se também com o facto de ser necessário, na maioria dos casos, algum treino do sistema para que a taxa de acerto seja maior. Isto exige que o utilizador despenda algumas horas a treinar o sistema, não garantida nunca uma total cobertura de todo o tipo de discursos. A tarefa de corrigir eventuais erros de interpretação pode ser bastante trabalhosa [26].

Alguns sistemas, apesar de hoje em dia já não ser um entrave tão comum dado a atual velocidade de processamento, têm ainda algum atraso na interpretação e escrita dos itens que lhes são ditados, o que por vezes, entre outros fatores, pode ainda levar a que o utilizador prefira utilizar os meios tradicionais como o teclado e o rato ao invés dos sistemas de reconhecimento de voz.

Finalmente, sublinha-se ainda o facto de que o conhecimento e bases de dados que abrangem os diversos dialetos mundiais são desproporcionais, ou seja, o nível do desenvolvimento do reconhecimento de voz em determinados dialetos é bastante superior a outros e vice-versa.

2.1.5 Software de Reconhecimento de Voz

Com a evolução tecnológica que se tem feito sentir no decorrer dos últimos anos e que tornou a utilização do computador e outros dispositivos eletrónicos parte da rotina, foi surgindo a necessidade de explorar outros meios alternativos para a interação com estes dispositivos para além dos meios tradicionais, por vezes por uma questão de conforto, conveniência ou acessibilidade. Foi-se então criando valor em torno desta temática, o que levou à criação de algumas soluções comerciais, cujas mais relevantes são descritas a seguir.

2.1.5.1 Philips FreeSpeech 2000

Esta solução foi sem dúvida uma das mais marcantes na história do software de reconhecimento de voz. Surgiu e começou a ser distribuído comercialmente em 1999 e já permitia ao utilizador ditar texto para qualquer processador de texto no computador, executar comandos no *Windows* com a voz e até navegar na *web*. Suportava na altura 6 línguas diferentes, incluindo a

Portuguesa e já era capaz de "aprender" à medida que ia sendo utilizada, adaptando-se aos padrões de fala do utilizador [27]. Apesar de por vezes apresentar alguns erros de interpretação, a sua taxa de acertos era já bastante aceitável, existindo ainda uma opção apresentada através de uma interface simples que permitia ao utilizador corrigir eventuais erros. Este *software* necessitava também de um treino inicial de, no mínimo uma hora para conseguir obter resultados significativos [28]. No entanto, o utilizador podia optar por continuar a treinar o *software* por um período de tempo mais longo, recorrendo à leitura de textos integrados no programa que permitiam uma aprendizagem bastante mais alargada e, consequentemente uma maior taxa de sucesso na transcrição.

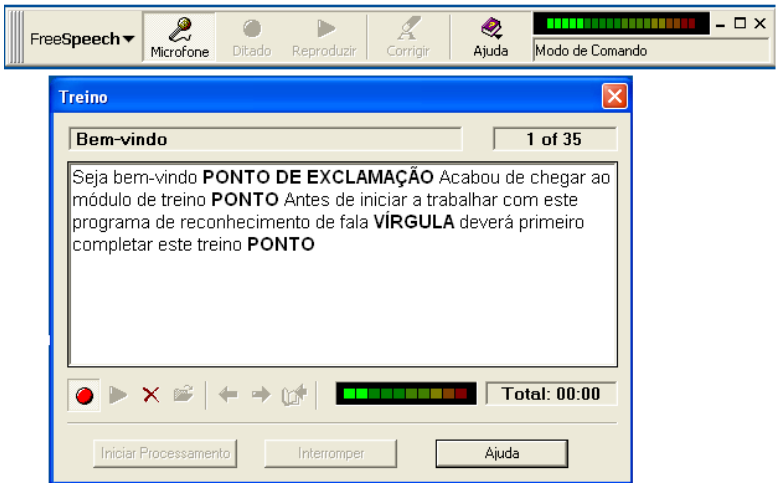


Figura 4 - Interface do *Freespeech 2000* em modo de treino

2.1.5.2 Dragon NaturallySpeaking

Esta é a solução comercial de referência atual das aplicações para o reconhecimento de voz. Esta solução apresenta uma taxa de sucesso na transcrição da fala para texto bastante aceitável quando utilizada em condições ideais (requer um microfone de razoável qualidade). Verifica-se, por exemplo, que as taxas de acerto para os microfones integrados nos computadores portáteis não são tão altas [29]. Desenvolvida atualmente pela *Nuance*, a *Dragon NaturallySpeaking* já vai na sua 13ª versão e apresenta várias versões que se adaptam ao modo de utilização pretendido. A versão *Home*, permite realizar tarefas como enviar *e-mails*, pesquisar na *Web*, escrever documentos, utilizar redes sociais e *chats*, etc. [30]. No caso da versão Premium, esta vem com um sistema de precisão melhorado. Permite, para além de efetuar as tarefas descritas anteriormente, realizar várias tarefas em simultâneo, ouvir o discurso ditado durante a sua utilização e oferece ainda uma ferramenta extra para síntese de voz que transforma o texto em discurso, o qual pode depois ser ditado ao utilizador [31]. Na sua versão *Professional*, esta solução mais uma vez traz as funcionalidades anteriores, juntando-lhes ainda outras mais direcionadas ao

ambiente empresarial, como por exemplo um sistema baseado na *cloud*[3], que permite ao utilizador ditar os seus textos onde quer que esteja, através do seu *Smartphone*. Permite também criar determinados atalhos por voz que remetem para processos repetitivos de modo a automatiza-los, e ainda adicionar palavras personalizadas para os diversos tipos de indústria (palavras não tão usuais no discurso do dia-a-dia, mas recorrentes em determinadas indústrias), ou termos únicos que se utilizam regularmente na empresa. Para além disto, conta ainda com mais uma funcionalidade que permite transformar ficheiros de áudio em texto, necessitando apenas de uma aprendizagem de 90 segundos para atingir uma taxa de sucesso aceitável [32].

Existem também duas versões gratuitas para *Smartphones* (*Dragon Search* e *Dragon Dictation*), que permitem efetuar pesquisas nos motores de busca comuns [33], transcrever voz para texto para utilizar na escrita de SMS, *e-mails* e aplicações de redes sociais [34].

Neste momento, este *software*, nas suas versões para PC não apresenta suporte para a língua Portuguesa. No entanto, nas versões mobile esta já está incluída.

2.1.5.3 Tazti

Este software foi concebido de um modo mais direcionado para o reconhecimento de comandos pela voz. Tal como os anteriores, é um programa disponibilizado comercialmente, no entanto tem uma opção de teste por 14 dias. É possível, através deste *software*, efetuar as operações básicas de utilização do computador que já vêm predefinidas (ex. consultar *websites*, abrir e controlar programas, etc.), assim como adicionar outras operações personalizadas. O *Tazti* funciona apenas em Inglês, englobando 4 tipos de pronúncias desta linguagem: Inglês Americano, Inglês do Reino Unido, Canadiano e Australiano.

Como já foi referido anteriormente, este programa funciona apenas à base de comandos, não sendo incluído nenhum processamento de texto mais complexo que inclua por exemplo o processamento de texto através da voz. No entanto, a característica que o diferencia dos anteriores é o facto de se conseguir controlar jogos de computador sem recurso ao rato e teclado, ou seja, apenas com a voz. Traz já consigo um perfil predefinido para o jogo *Warcraft*, sendo possível construir e adicionar novos perfis para outros jogos [35].

O estagiário efetuou um pequeno teste a este *software* onde se notou que o mesmo obedece aos comandos de forma bastante satisfatória. No entanto o processamento é um pouco demorado, o que pode incidir na viabilidade da sua utilização em alguns tipos de jogos que exijam ações rápidas.

[3] Cloud – permite o armazenamento de dados online, sendo deste modo possível aceder aos mesmos independentemente da localização do utilizador

2.1.5.4 Braina

Esta aplicação, à semelhança da anterior, permite controlar várias funcionalidades no computador através da voz, incluindo ainda o processamento de texto. O que distingue esta aplicação das anteriores, é o tipo de interação sob a forma de assistente virtual, que nos faz lembrar o modo de interação com os assistentes das plataformas móveis. Segundo a página oficial desta plataforma, o *Braina* é fruto de uma grande investigação no campo da inteligência artificial, tendo como objetivo tornar esta aplicação cada vez mais inteligente, aprendendo pela experiência como se de um cérebro humano se tratasse [36]. Esta aplicação é também capaz de "entender" a linguagem e aprender através da conversa [37]. O utilizador pode fazer perguntas por voz ou texto à aplicação (por exemplo perguntar o estado do tempo) e obter respostas instantâneas baseadas em serviços *online*.

Outra das características desta aplicação é a sua integração com dispositivos móveis *Android*. Ao instalar uma aplicação desenvolvida pela mesma empresa (Remote PC Voice Control) num *Smartphone*, é possível utilizar o seu microfone para enviar comandos para o computador através da rede *wifi* doméstica, a partir de qualquer local da casa ou efetuar qualquer uma das outras tarefas que requerem a utilização do microfone [38].

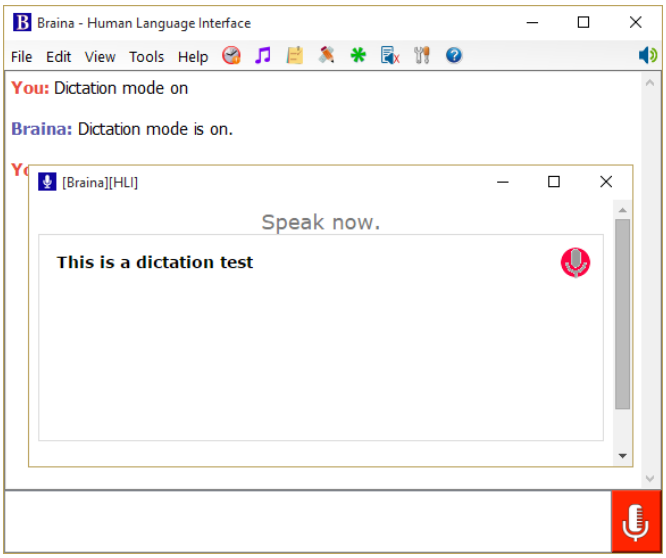


Figura 5 - Plataforma de reconhecimento de voz da *Braina* [122]

2.1.5.5 Assistentes Virtuais

Ao falar sobre sistemas de reconhecimento de voz, não se pode deixar de referir os assistentes virtuais dos principais sistemas operativos móveis atuais. Recuando uns anos atrás, ao momento que a *Siri* (assistente virtual dos sistemas da *Apple*) foi lançada, eram poucos os que acreditavam na credibilidade ou utilidade de um sistema deste tipo, tanto que acabaram por utilizá-

lo apenas pontualmente e, por vezes acabava por ser visto mais como uma forma de entretenimento. Depois de alguns anos sem existir novidades neste âmbito, tanto a *Google* como mais recentemente a *Microsoft*, acabaram por lançar também os seus assistentes virtuais, relançando novamente toda esta temática e também a do reconhecimento de voz em si [39].

Todos estes assistentes são capazes de entender linguagem natural, interpretando corretamente a maioria dos comandos que lhes são enviados, encontrando-se todos ao mesmo nível no que diz respeito ao reconhecimento de voz em Inglês. Todos são capazes de lançar aplicações, procurar algo na internet, fornecer informações requisitadas pelo utilizador e até "contar uma piada". Em termos de resultados, a *Siri*, talvez devido a uma mais longa presença no mercado e consequentemente uma maior exploração destas funcionalidades, interpreta e apresenta melhores resultados que as outras duas plataformas [40], tendo até uma "personalidade" um pouco mais humana que por exemplo o *Google Now*. Falando agora deste último, destaca-se o facto desta aplicação conseguir agregar vários serviços numa só aplicação, utilizando o reconhecimento de voz como complemento [41], e de ter um papel de "assistente" ligeiramente melhor do que as outras aplicações. Já no caso da *Cortana*, destaca-se o facto de, apesar deste sistema ser bastante recente, não fica muito atrás dos seus concorrentes, podendo isto dever-se ao histórico que a *Microsoft* tem no mundo do reconhecimento de voz. Em termos de "personalidade", esta aplicação é a que mais se aproxima do modelo humano [42], tendo uma "voz" mais natural e sendo até capaz de interpretar por exemplo o humor em certas frases, respondendo igualmente com humor.

Como já foi referido, todas estas aplicações têm uma excelente taxa de sucesso no reconhecimento de voz em Inglês. No caso da língua Portuguesa, neste momento só existe em Português Brasileiro na *Siri* e no *Google Now*, apresentando uma taxa de sucesso significativamente inferior à versão em Inglês. A *Microsoft* anunciou também a chegada da *Cortana* em Português no final de 2015, mas mais uma vez em Português do Brasil [43].

2.1.5.6 Windows Speech Recognition

Esta é uma solução integrada no sistema operativo *Windows* que já o tem vindo a acompanhar desde que foi lançado o *Windows 7*. Esta solução, um pouco à semelhança das soluções comerciais descritas anteriormente, permite controlar diversas funcionalidades nativas do *Windows* e até editar documentos. No entanto, no que toca ao software de terceiros (por exemplo o *browser Google Chrome*), a sua funcionalidade fica bastante mais reduzida.

Também à semelhança das aplicações anteriores, esta requer algum treino e vai aprendendo através dos *inputs* e correções efetuadas pelo utilizador [44].

No entanto, mais uma vez, este *software* não suporta a língua Portuguesa, suportando apenas Inglês, Francês, Espanhol, Alemão, Japonês e Chinês [45].

2.1.5.7 Ferramentas Online de Transcrição de Voz para Texto

Existem também algumas ferramentas *online* que permitem ao utilizador converter voz para texto de um modo bastante simples como um editor de texto online (por exemplo as aplicações *SpeechTexter* [46], *Dictation Online Speaking* [47], *TalkTyper* [48], etc.), não tendo qualquer outro tipo de funcionalidade para além desta. Todas estas aplicações são baseadas nas funcionalidades da *Web Speech API* (Application Programming Interface) da *Google* (encontra-se descrita no capítulo **3.3.3**), suportando uma larga quantidade de idiomas que inclui o Português de Portugal. Como já se esperava, o reconhecimento de voz em Inglês nestas aplicações, tem uma melhor taxa de acerto. No entanto, a sua performance a nível da transcrição da língua Portuguesa também é bastante aceitável, reconhecendo a maioria das palavras sem problemas.

Existem também algumas aplicações deste tipo na *webstore* da Google que se integram facilmente no *Google Chrome*. No entanto existe uma que se destaca pelo facto de permitir também transcrever voz para texto em qualquer aplicação do *Windows* (por exemplo *Word, Excel*, etc.) e, para além disso permite também preencher campos de texto (*text fields*) em qualquer *website* através da voz, transcrever ficheiros de áudio para texto e ainda efetuar traduções instantâneas para outro idioma. Neste momento esta aplicação suporta uma grande variedade de idiomas, incluindo o Português de Portugal [49].

2.1.5.8 Ambientes de Desenvolvimento

A oferta de *software* de reconhecimento de voz grátis é bastante reduzida, sendo que as melhores alternativas neste campo se encontram em sistemas integrados em ambientes de desenvolvimento, ou seja, versões de código aberto em várias linguagens de programação que vão sendo atualizadas pelos seus colaboradores.

Antes de avançar é então necessário definir alguns conceitos:

Modelos de Linguagem: Listas de distribuições probabilísticas de certas palavras ocorrerem em sequência [50].

Modelos Acústicos: Representação dos sons distintos de cada palavra presente no modelo de linguagem. Relação entre os sinais de áudio e a fonética ou outras unidades linguísticas que constituem as palavras [51].

Decoder: Sistema que procura no modelo acústico, sons equivalentes aos introduzidos pelo utilizador e os traduz para palavras [52].

Existem então algumas bibliotecas gratuitas para desenvolvimento que não contêm propriamente uma interface gráfica como era o caso das alternativas descritas anteriormente. As mais populares são: a *CMU Sphinx*, desenvolvida em J*ava* para ambiente *Linux*; a *HTK* e *Julius*, desenvolvidas em linguagem C sendo sistemas multiplataforma; o *Kaldi*, desenvolvido em C++ e igualmente multiplataforma; e a *iATROS* desenvolvida em linguagem C para *Linux*. Como foi referido anteriormente, estes sistemas ou bibliotecas não têm uma interface gráfica, existindo, no entanto, dois projetos interessantes que fornecem esse tipo de interface ao sistema: o *Simon*, desenvolvido em C++ para sistemas multiplataforma com suporte às bibliotecas *CMU Shpinx*, *HTK* e *Julius* e o *Jasper Project* desenvolvido em *Python* para *Raspberry Pi* com suporte às bibliotecas *CMU Sphinx* e *Julius* [53]. À semelhança das aplicações comerciais, estas conseguem também efetuar tarefas de edição de documentos através de comandos por voz.

Todas as alternativas anteriores se centram na mesma lógica de funcionamento que consiste em modelos de linguagem, modelos acústicos e *decoders* que necessitam de ser integrados entre si. Todos estes sistemas apresentam apenas bibliotecas em Inglês e, no caso do *Julius*, também em Japonês.

2.2 Síntese de Voz

Pode entender-se como síntese de voz, a capacidade de um dispositivo eletrónico simular a voz ou o discurso humano a partir de texto. O dispositivo recebe o discurso sob a forma de texto, transforma-o na sua representação fonética[4] e converte-a de seguida em sons similares aos do discurso humano [54].

Os primeiros esforços no âmbito desta tecnologia permitiram através de instrumentos mecânicos, simular apenas os sons mais simples do discurso humano, ou seja, as vogais. No entanto, com o avançar da tecnologia e principalmente com o surgir dos sistemas elétricos, foram-se desenvolvendo sistemas cada vez mais eficientes e com um modo de discurso cada vez mais próximo ao do ser humano [55].

[4] Representação Fonética – Representação das palavras em símbolos que equivalem a sons segundo o Alfabeto Fonético Internacional (AFI). O AFI é constituído por símbolos que representam os sons básicos mais frequentes das línguas existentes no mundo.

2.2.1 Revisão Histórica

O despontar da tecnologia da síntese de voz deu-se bastante mais cedo que a do reconhecimento de voz, ainda que de um modo bastante arcaico. Em 1779, *Christian Kratzenstein*, um professor Russo, de modo a conseguir explicar as diferenças físicas entre as 5 vogais existentes, construiu um dispositivo constituído por 5 instrumentos similares ao trato vocal humano no momento em que se reproduz o som de cada uma das vogais. Estes instrumentos, ou *ressoadores acústicos de Kratzenstein* como foram mais tarde intitulados, funcionavam à base do sopro e reproduziam cada um, um som diferente, fazendo vibrar palhetas em diferentes estruturas de modo similar ao funcionamento de uma flauta. As 5 diferentes combinações de palhetas e estruturas construídas por *Kratzenstein* permitiram, portanto, simular o som de cada uma das vogais existentes [56].

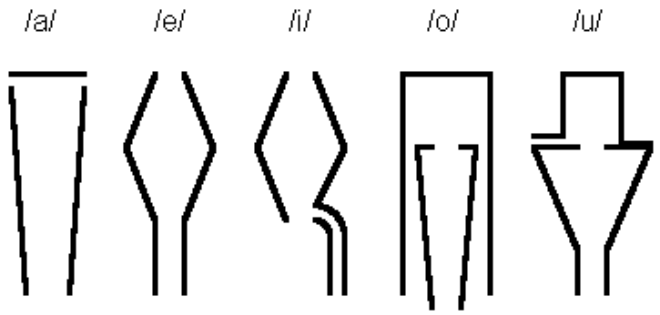


Figura 6 – Representação da estrutura dos ressoadores acústicos de *Kratzenstein* [123]

Uns anos mais tarde, *Wolfgang von Kempelen*, um inventor Austro-Húngaro, apresentou em Viena um dispositivo mecânico que para além de ser capaz de reproduzir o som das vogais, conseguia também reproduzir outras combinações de sons mais complexos. Este dispositivo era constituído por diversos módulos que permitiam simular o comportamento de todos os órgãos humanos que possibilitam a fala, sendo possível deste modo simular também o som das consoantes. As peças-chave desta máquina centravam-se essencialmente numa câmara de pressão (semelhante aos acordeões) que simulava o comportamento dos pulmões, uma palheta vibratória que agia similarmente ao funcionamento das cordas vocais, quatro cavidades diferentes controladas pelos dedos para simular o som das consoantes e ainda um modelo em couro, em forma de cilindro, que permitia simular o trato vocal, incluindo uma "língua" articulada e uns "lábios" móveis que possibilitavam sons oclusivos[5] [57].

No entanto, nesse momento era ainda impossível reproduzir palavras completas. Este cenário acabou por ser ultrapassado uns anos mais tarde, a meio do século XIX, por *Charles*

[5] Sons Oclusivos – Representam uma consoante que é produzida quando se para o fluxo de ar através dos lábios, dentes ou língua, seguido de uma libertação de ar repentina (por exemplo, as letras "p" ou "t").

Wheatstone, outro inventor do Reino Unido, que através da simplificação e melhoramento do *design* do dispositivo de *Kempelen*, conseguiu que este fosse capaz de reproduzir mais consoantes (19 no total) e, por consequência, já era possível construir algumas palavras [58].

Com o surgir da eletricidade e dos sistemas elétricos, passou a ser possível também a utilização de novas abordagens para a construção de novos dispositivos de síntese de voz. Foi então em 1939, na Feira Mundial de Nova Iorque, que *Homer Dudley*, um investigador dos laboratórios *Bell*, apresentou ao mundo a sua invenção: o VODER (*Voice Operating Demonstrator*). Este dispositivo foi considerado o primeiro sistema elétrico capaz de simular o discurso humano e consistia essencialmente num teclado e numa série de alavancas que controlavam os vários componentes capazes de gerar sons, sílabas e palavras. Era ainda possível, com este sistema, aplicar entoação à voz, simular vozes diferentes e até "cantar". Segundo *Dudley*, o VODER era capaz de falar em qualquer linguagem desde que o seu operador a conhecesse [59].

Figura 7- Demonstração do VODER [124]

Mais tarde, a meio do século XX, contruíram-se mais alguns sistemas para síntese de voz. Um dos dispositivos mais famosos foi construído por *Cooper, Haskins e Borst* e foi apelidado de "*Pattern Play-back Machine*". Este consistia na leitura de pequenos furos através da luz, num pequeno disco de plástico. A luz, ao passar por estes pequenos furos, e dependendo da intensidade da transparência que era aplicada a cada furo, transmitia ao sistema as diferentes frequências correspondentes a cada tipo de som. Ou seja, a intensidade da luz recebida era interpretada de forma proporcional tal como a intensidade da frequência de determinado som. Estes sinais elétricos eram então reunidos e amplificados de forma a produzir o som do discurso [60]. Mais tarde, em

1979, no MIT (*Massachusetts Institute of Technology*), foi desenvolvido um sistema denominado por *MITalk*, que serviu como base para outros sistemas mais avançados similares aos que se utilizam hoje em dia [61]. Até então estes sistemas ainda funcionavam com uma voz simulada e de carácter robótico. Mas, com o surgimento dos computadores, esta tarefa ficou bastante facilitada pois já era possível um melhor processamento e tratamento da conversão de texto para voz. Com o surgimento de novos algoritmos, de uma melhor capacidade de processamento e ainda das tecnologias *web*, foi possível chegar ao patamar onde estamos hoje, em que já existem sistemas capazes de simular o discurso humano de forma quase natural. Hoje em dia, já é possível escutar uma transcrição quase perfeita de texto para voz utilizando sistemas artificiais. Temos como exemplo os atendedores telefónicos automáticos ou aplicações para pessoas com dificuldade ou impedidas de falar. Um caso bastante conhecido é o do físico consagrado *Stephen Hawking* que, apesar das condições que o impediam de falar, graças a um sistema deste tipo, este era capaz de falar e transmitir o seu conhecimento.

2.2.2 Modo de Funcionamento

O funcionamento da tecnologia de síntese de voz, resume-se essencialmente a três fases: **Texto para Palavras**, em que é analisado o contexto das palavras de modo a diminuir a probabilidade de erros no discurso; **Palavras para Representação Fonética**, em que o texto é convertido para a sua representação fonética e **Representação Fonética para Som**, em que se converte o resultado obtido na fase anterior para som audível e percetível pelo ser humano como discurso. No entanto, atualmente existem três abordagens principais que se podem seguir para realizar esta última operação: a *Síntese Concatenativa*, a *Síntese Formante* e finalmente a *Síntese Articulatória*, sendo esta última considerada a mais complexa [55].

2.2.2.1 Texto para Palavras

O principal objetivo desta fase é reduzir a ambiguidade em termos do contexto em que determinadas palavras são lidas. Esta que é a primeira fase do processo de síntese de voz, também chamada de fase de normalização ou pré-processamento, foca-se em analisar o texto e verificar o contexto de determinadas palavras de modo a que a leitura seja realizada da forma mais correta possível.

Como é de conhecimento geral, na linguagem existem palavras homógrafas[6], assim como números ou símbolos. Portanto, o sistema não se poderá limitar simplesmente a pronunciar cada palavra do mesmo modo, independentemente do seu contexto.

Consideremos por exemplo o número "1987". Dependendo do contexto, existem várias formas de ler este número: pode dizer respeito a um ano ("… nasci no ano de mil novecentos e oitenta e sete."), ou a uma quantidade que pode ser lida no contexto do género masculino ou feminino ("existem mil **novecentos** e oitenta e sete exemplares…" ou "… estiveram presentes mil **novecentas** e oitenta e sete pessoas."), ou ainda um conjunto de números ("…insira o código um, nove, oito, sete".). Sendo assim, deverá ser feita uma análise preliminar ao texto, verificando (quando aplicável) qual a forma mais correta de pronunciar determinada palavra, tendo em conta o modo como esta se insere no restante texto. Para esta tarefa, um pouco à semelhança dos métodos de reconhecimento de voz, utilizam-se as já referidas técnicas de *Redes Neuronais* ou *Modelo Oculto de Markov*, os quais, recorrendo à análise de padrões em comparação com o contexto, determinam qual a pronúncia mais provável para a palavra pretendida [55].

2.2.2.2 Palavras para Representação Fonética

Após o término da fase anterior, o sistema necessita de converter o texto para a sua representação fonética, ou seja, transformar as palavras nos fonemas[7] que as compõem.

Esta tarefa em teoria seria bastante trivial já que, partindo do princípio que existe uma biblioteca com todos os fonemas, o único trabalho a desempenhar seria o mapeamento das letras ou conjunto de letras que constituem a palavra nos fonemas correspondentes. No entanto, todas as linguagens têm as suas particularidades, particularidades essas que o sistema tem que ter em conta no momento da transcrição das palavras para a sua representação fonética. Existem, por exemplo, letras ou sílabas que apesar de serem escritas da mesma forma em determinadas palavras, soam de modo diferente (por exemplo, a letra "e" das palavras "venho" e "termino" é lida de forma diferente), ou seja, dizem respeito a diferentes fonemas [62].

Outro fator a ter em conta nesta fase, centra-se no facto de que na linguagem natural humana expressam-se emoções, enfatizam-se determinadas palavras consoante o que é dito ou sentido e até são aplicadas diferentes velocidades de discurso [55]. Algumas destas emoções são possíveis de transmitir por texto, recorrendo à pontuação (por exemplo, recorrer ao ponto de exclamação para representar surpresa). No entanto também a pontuação pode ter várias interpretações no contexto da linguagem (tendo em conta o exemplo anterior, o ponto de exclamação tanto pode representar surpresa como também, por exemplo entusiasmo). Mais uma vez, para que o discurso

[6] Palavras Homógrafas – Palavras que apesar de se escrevem da mesma forma, pronunciam-se de modo diferente, tendo também significados diferentes.

[7] Fonemas – Unidades sonoras que representam a pronúncia de determinadas letras ou conjunto de letras na linguagem.

simulado seja o mais natural possível, o sistema tem que saber lidar com estas situações. Caso contrário, todo o discurso é realizado com o mesmo tom de voz tornando-o demasiado artificial. Deste modo, o método mais comum para contornar este problema passa pela utilização de SSML (*Speech Synthesis Markup Language*). Este método consiste em, de forma bastante similar ao XML (*eXtensive Markup Language*), utilizar *tags*[8] com diversos elementos e/ou atributos que caracterizam o seu conteúdo (neste caso, o discurso). Ou seja, especificam-se os elementos e atributos do discurso (tais como a velocidade, voz, tipo de entoação, etc.) através de *tags* em SSML as quais serão aplicadas ao conteúdo que se encontra dentro dessas *tags* [63].

```
That is a <emphasis> big </emphasis> car!
That is a <emphasis level="strong"> huge </emphasis> bank account!
```

Figura 8 – Exemplo de Utilização das *tags* SSML

2.2.2.3 Representação Fonética para Som

Depois de concluída a fase anterior, é necessário converter a representação fonética obtida para discurso percetível ao ser humano. Para isso, têm vindo a ser desenvolvidas diversas abordagens, das quais se distinguem três que são descritas a seguir.

Síntese Concatenativa

Este tipo de abordagem recorre a gravações de fragmentos de voz humana, os quais podem ser frases ou palavras completas ou até pequenas frações de som (fonemas) retirados dessas mesmas palavras, que podem ser reorganizados para formar outras palavras diferentes. No decorrer da transformação da representação fonética para som, o sistema irá recorrer a estes pequenos fragmentos de voz humana guardados em bases de dados e, dependendo da organização da representação fonética de cada palavra, esses fragmentos serão reordenados de modo a formar palavras completas. Dado que neste processo é utilizada uma quantidade mínima de sinal digital, este tipo de síntese é considerada como sendo a mais natural, ou seja, a que mais se aproxima ao discurso humano [64]. No entanto, a principal desvantagem deste processo centra-se no facto de que na maioria dos sistemas deste tipo, a síntese de voz é geralmente limitada a apenas uma voz e consequentemente apenas um idioma [55].

[8] Tags – Em SSML, uma *tag* é utilizada para definir as características dos elementos de discurso abrangidos.

Síntese Formante

O método utilizado por esta abordagem considera o discurso como sendo formado da mesma forma que é gerado o som nos instrumentos musicais, ou seja, o discurso é visto como um padrão de som sujeito a diversas variações que são capazes de formar diferentes vozes. Deste modo, recorrendo à combinação artificial das várias frequências de ressonância comuns à voz humana, o discurso é totalmente gerado em tempo real, não existindo a necessidade de recorrer a bases de dados como acontece na abordagem anterior. Estas características permitem que o sistema seja capaz de efetuar a síntese de voz de qualquer palavra em qualquer linguagem, tornando esta abordagem a mais flexível. No entanto, dada a natureza totalmente artificial deste método, a voz gerada neste tipo de sistemas distancia-se da voz humana, apresentando ainda muitos traços de caráter robótico [55].

Síntese Articulatória

Esta abordagem é considerada uma das mais complexas e, por consequência, também uma das menos exploradas [55]. O método por detrás desta, foca-se na produção de discurso artificial simulando todos os aspetos da produção de discurso natural. Recorrendo a uma representação física similar ao trato vocal humano, que conta com diversas características que simulam o comportamento dos vários órgãos responsáveis pela fala, é possível reproduzir qualquer tipo de discurso. Este último é produzido combinando diferentes tipos de comportamento nos vários componentes que fazem parte do sistema de modo a reproduzir todos os fonemas que constituem a fala. Teoricamente a voz produzida por este tipo de sistemas seria a que mais se assemelharia à voz humana. No entanto, hoje em dia a produção de discurso é ainda um processo bastante complexo cujos detalhes ainda não foram totalmente explorados e entendidos, o que torna este método ainda pouco viável [65].

2.2.3 Aplicações da Síntese de Voz

O discurso e a linguagem foram desde sempre as tarefas mais importantes na comunicação entre seres humanos. Com o despontar da era informática foram surgindo, para além dos computadores, diversos dispositivos eletrónicos que cada vez mais nos vão proporcionando uma maior interatividade. Deste modo, foi surgindo também a necessidade de alargar os processos de comunicação que estes dispositivos utilizam para comunicar connosco. Um desses processos é a síntese de voz, que hoje em dia, cada vez mais, vai marcando presença em diversos momentos da nossa rotina. Enumeram-se a seguir algumas das aplicações principais da síntese de voz.

2.2.3.1 Síntese de Voz como Ferramenta de Acessibilidade

À semelhança do reconhecimento de voz, a acessibilidade foi um dos fatores que mais contribuiu para o desenvolvimento da síntese de voz na era moderna. As ferramentas de acessibilidade utilizadas por deficientes visuais ou por pessoas com problemas fonéticos, são talvez dos mais importantes exemplos da utilização desta tecnologia.

Antes da existência da síntese de voz, era necessário aos deficientes visuais recorrer ao braile caso pretendessem ler um livro ou, utilizando uma abordagem diferente, recorrer a *audiobooks*[9]. No entanto, estas duas soluções nem sempre estavam disponíveis no idioma nativo de cada utilizador e, no caso dos *audiobooks*, eram necessários vários meses para que os mesmos fossem gravados, sendo esta tarefa de produção bastante dispendiosa [66]. Surgiu então uma nova abordagem que consiste em dispositivos capazes de digitalizar e interpretar texto em livros físicos através de OCR[10] (*Optical Character Recognition*) e, de seguida, produzir discurso com o conteúdo do livro. Este método, para além de alargar os meios de leitura disponíveis para estes indivíduos, veio resolver os problemas referidos, possibilitando ainda a leitura de livros que não se encontrem em formato digital (que neste caso já seria possível a sua leitura através de um software de síntese de voz) [67]. Um dos mais famosos dispositivos deste tipo foi produzido pela *Intel* e denominado por *Intel Reader*.

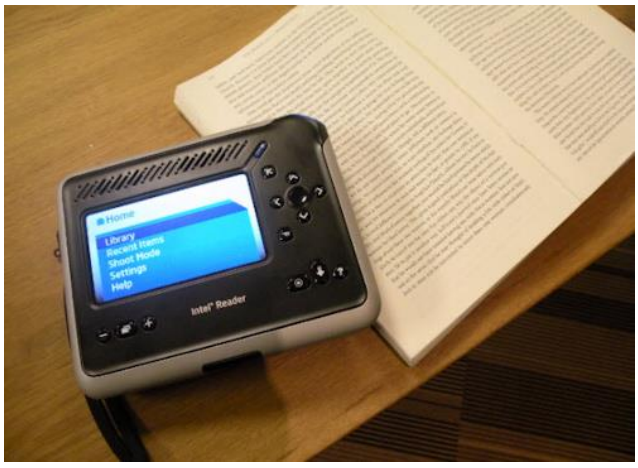


Figura 9 - *Intel Reader* [68]

Atualmente, tornou-se implícita a utilização do computador ou do *smartphone* como parte da rotina diária, assim como em grande parte das tarefas profissionais na maioria dos sectores. De modo a que estes dispositivos possam também ser utilizados por pessoas com deficiência visual, têm vindo a ser criadas diversas aplicações que através da síntese de voz, vão descrevendo o conteúdo do ecrã auxiliando assim o utilizador na realização de determinadas tarefas (por exemplo,

[9] Audiobook – Gravação do conteúdo de um livro que é narrado em voz alta.
[10] OCR – Tecnologia que reconhece texto escrito fisicamente e o converte para formato digital.

ler o conteúdo de uma página *web*, ler o *e-mail*, relatar o local onde o ponteiro do rato se encontra, etc.) [69].

Já no caso das pessoas com deficiências ao nível fonético, é-lhes possível recorrer, por exemplo às aplicações de síntese de voz mais comuns (em que o utilizador digita o texto para que o mesmo seja convertido em síntese). No entanto, existem soluções especializadas que se adaptam conforme as necessidades especiais do utilizador e, englobando as tecnologias de síntese e de reconhecimento de voz derrubam as barreiras físicas que impedem estes utilizadores de uma interação completa com o sistema [70].

2.2.3.2 Síntese de Voz no Quotidiano

Tem-se vindo a tornar cada vez mais comum, cruzarmo-nos com sistemas de síntese de voz em algumas situações do nosso dia a dia. Um exemplo bastante recorrente são os sistemas de anúncios utilizados em algumas estações ferroviárias e em alguns comboios. Estes sistemas utilizam a síntese de voz para anunciar automaticamente as informações relativas às partidas, chegadas, atrasos, estações de paragem, etc. dos comboios, não sendo necessário deste modo, recorrer a mão de obra humana. A maioria destes sistemas utiliza o método da síntese concatenativa de modo a que o discurso pareça mais natural e percetível para os passageiros, já que neste método é utilizada uma voz humana pré-gravada. No entanto, tendo em conta o funcionamento deste método de síntese de voz, estes sistemas não se limitarão apenas a um conjunto de palavras chave mais comuns, já que será possível criar novos anúncios introduzindo texto no sistema que, imediatamente, é convertido para voz [71].

Foram já descritos os assistentes digitais dos *smartphones* no âmbito do tema do reconhecimento de voz. Contudo, estes sistemas contam também com uma funcionalidade de voz por síntese automática que é utilizada para responder aos pedidos do utilizador. Mais uma vez, estes sistemas utilizam o método da síntese concatenativa para proporcionar uma experiência mais agradável e uma "conversa" mais natural entre o utilizador e o assistente digital. Desde que estes sistemas surgiram, têm vindo a ser realizadas melhorias à voz e à lista de fonemas disponíveis de modo a que o assistente digital seja capaz de reproduzir praticamente qualquer palavra que surja. Por exemplo, no caso da *Siri*, têm vindo a ser gravados e armazenados vários fonemas pronunciados de diferentes formas, aumentando deste modo a abrangência do discurso [72].

A síntese de voz está presente também em alguns sistemas de inquéritos, leitura de SMS's e avisos automáticos realizados por chamadas telefónicas. Mais uma vez, estes sistemas substituem a mão de obra humana no tratamento de grandes quantidades de dados (por exemplo na realização de inquéritos telefónicos em grandes amostras de população), agilizando e automatizando este tipo de tarefas. No entanto, em casos onde não é necessária a construção de discurso adaptativo (por exemplo menus telefónicos interativos onde figuram sempre as mesmas frases), basta

simplesmente recorrer-se a trechos de frases gravadas que são depois reproduzidas dependendo do momento ou resposta do utilizador.

2.2.3.3 Síntese de Voz como Plataforma Educacional

Recentemente, com a crescente importância que a educação vai tendo no mundo atual e, principalmente, dada à necessidade de aprender cada vez mais e melhor novas linguagens (como por exemplo, o Inglês), a síntese de voz tem sido adotada também em novas abordagens ligadas à aprendizagem.

Algumas plataformas de cursos *online* adotam esta tecnologia, de modo a automatizar e rentabilizar o processo da criação dos cursos (o conteúdo do curso pode ser transcrito por síntese de voz, em vez de estar disponível apenas em texto). Este método, para além de promover a compreensão dos alunos, vai de encontro a determinadas necessidades que alguns dos estudantes possam ter, tais como dificuldades na leitura da língua visada ou até dificuldades ao nível da acessibilidade, as quais já foram descritas.

Esta tecnologia pode ser também adotada para a aprendizagem de novas linguagens, de forma a melhorar a leitura, a compreensão e a pronúncia das mesmas, por exemplo fazendo com que exista a possibilidade de escrever determinadas palavras e ouvi-las de seguida, pronunciadas da forma correta [73].

2.2.4 Problemas e Limitações da Síntese de Voz

Atualmente a síntese de voz encontra-se a uma distância considerável de atingir a perfeição. Existem determinados fatores que, com a tecnologia e abordagens atuais, ainda não são possíveis de contornar o que, por sua vez, levam a que a síntese de voz se afaste do discurso natural humano.

Um dos problemas mais notórios nesta tecnologia, prende-se com o facto da falta de naturalidade presente em todos os métodos existentes. Foi anteriormente referido que existem determinados métodos que possuem uma maior naturalidade que outros. No entanto, dependendo do método utilizado, cada um dos mesmos tem sempre as suas desvantagens neste aspeto. Alguns métodos apresentam uma voz robótica, enquanto outros, apesar de apresentarem uma voz humana, contêm também elementos que retiram a naturalidade ao discurso. Geralmente, num discurso real os intervenientes variam a entoação quando falam. Ou seja, mesmo que pronunciem a mesma frase, esta dificilmente será transmitida exatamente com a mesma entoação, o que não acontece no discurso simulado.

Outro dos problemas mais notórios desta tecnologia encontra-se na dicção de homógrafos (já explicados na secção 3.2.2.1) e determinadas palavras ou expressões menos comuns. Por exemplo, quando surgem acrónimos ou nomes estrangeiros a meio de um texto, a probabilidade

do sistema errar a sua síntese é bastante alta dado que a síntese de voz geralmente está apenas preparada para a fonética caraterística do dialeto. O mesmo acontece com alguns carateres especiais ou símbolos que poderão estar presentes no texto e que poderão não ter uma transcrição válida, ou com a leitura errónea de números já referida anteriormente [74].

Finalmente, mas não menos importante, na comunicação humana existem também os complementos não verbais que o sistema não aborda. Há sentimentos que podem ser expressados pelo tom de voz e, apesar de existir SSML em alguns sistemas, este método acaba também por ter as suas falhas, principalmente pela notória artificialidade na ênfase do discurso. Por outro lado, estes sentimentos são também demonstrados pela postura ou linguagem gestual dos interlocutores, algo que é impossível de demonstrar através dos habituais sistemas de síntese de voz.

2.2.5 Software para Síntese de Voz

Quando se pretende uma interação mais natural com os dispositivos eletrónicos, é necessário que os mesmos comuniquem do modo o mais aproximado possível da comunicação humana. Posto isto, de forma a tornar esta experiência possível, foram sendo desenvolvidas soluções que permitem a aplicação da síntese de voz em diversos dispositivos, as quais são descritas de seguida.

2.2.5.1 Skype Translator (Tradução Instantânea do Skype)

A barreira linguística, apesar de hoje em dia já não ser tão significativa como no passado, ainda representa um obstáculo à comunicação entre indivíduos que falem diferentes idiomas. Foi a pensar nisso que se desenvolveu uma ferramenta que possibilita aos utilizadores de um dos softwares de comunicação mais utilizados a nível mundial, o Skype, comunicarem entre si, em tempo real, sem que se imponham estas barreiras linguísticas. Esta ferramenta permite, através de reconhecimento e, posteriormente através de síntese de voz, traduzir em tempo real todas as frases proferidas na linguagem natural de um interlocutor para a linguagem natural do outro e vice-versa [75].

Neste momento, esta ferramenta suporta 12 idiomas (entre os quais o Português Brasileiro) na sua vertente de chamadas de voz/vídeo e cerca de 50 na vertente de mensagens instantâneas [76].

2.2.5.2 Google Tradutor

A famosa plataforma de traduções *online* da Google, o Google Tradutor, disponibiliza também uma funcionalidade interessante que recorre à síntese de voz. Este recurso tem como finalidade a sintetização do texto escrito pelo utilizador, tanto no idioma a traduzir, como no idioma traduzido. Esta funcionalidade, permite ao utilizador ouvir todo o texto introduzido, assim como excertos do mesmo, sendo deste modo uma excelente forma de o utilizador se familiarizar com a correta pronúncia de determinadas palavras na linguagem pretendida.

Esta ferramenta suporta, neste momento, mais de 100 idiomas e está disponível via *browser* e via aplicação móvel (*Android* ou *iOS*) [77].

2.2.5.3 Nuance Vocalizer

Tal como a solução para reconhecimento de voz da *Nuance* já apresentada, "Dragon NaturallySpeaking", a mesma marca oferece também uma plataforma para síntese de voz. Esta plataforma é mais direcionada para o desenvolvimento de soluções para empresas que queiram, por exemplo, automatizar o processo de atendimento ao cliente.

Segundo a página oficial desta plataforma, o *Nuance Vocalizer* oferece um mecanismo para síntese de voz que permite aos utilizadores (neste caso, aos clientes da empresa), uma interação mais personalizada e aproximada ao modelo humano. Para tornar isto possível, esta ferramenta recorre a uma poderosa rede neuronal, que vai sendo aperfeiçoada e adaptada ao modelo de negócio da empresa à medida que o tempo passa. Ainda segundo a mesma página, o discurso apresentado pela solução oferece uma naturalidade e expressividade bastante avançada, sendo possível moldar a sua ênfase e entoação como acontece numa conversação normal.

Esta ferramenta tem, neste momento, suporte para 53 idiomas em 119 vozes diferentes, e possibilita o desenvolvimento de soluções IVR *(Interactive Voice Response*) para diversos tipos de canais digitais [78].

2.2.5.4 Windows Narrator

Esta é uma funcionalidade integrada no Windows 10 que foi desenvolvida essencialmente para utilizadores com deficiência visual. Ao ativar esta funcionalidade, juntamente com o teclado e rato, o utilizador passa a ter acesso à narração dos itens com que interage, leitura de documentos ou *e-mails* e apoio à redação de texto (narração de informações sobre formatações de texto e pontuação). É possível ainda ao utilizador, alterar a velocidade, tom e volume da voz do narrador, assim como alterar a voz ou o idioma pretendido [79].

Atualmente, esta solução suporta 47 idiomas diferentes que podem ser instalados gratuitamente, sendo que em alguns deles, é possível escolher entre uma voz masculina ou uma voz feminina para a narração [80].

2.2.5.5 Ferramentas Online

À semelhança do que acontece com o reconhecimento de voz, existem também algumas ferramentas *online* que possibilitam a síntese de texto para voz. A funcionalidade principal destas ferramentas limita-se apenas à conversão do texto introduzido pelo utilizador para voz, tendo todas elas disponível um conjunto de vozes e idiomas predefinidos que são possíveis de utilizar. Algumas destas ferramentas como a *Naturalreaders* [81], *Festvox* [82] e *iSpeech* [83] têm também a opção de alterar a velocidade do discurso e efetuar o *download* do resultado em formato áudio. No entanto, estão limitadas a uma determinada quantidade de conversões diárias ou limite de palavras nas suas versões gratuitas, não podendo ser utilizadas para fins comerciais. Já nas suas versões *Premium*, estas ferramentas têm suporte para conversão direta de documentos (em diferentes formatos como .doc, .pdf, etc.) para voz, leitura integral de *websites On-Demand* e integração da funcionalidade de síntese de voz no *website* do subscritor para que qualquer visitante tenha acesso à mesma. No caso da *Naturalreaders* é ainda fornecido ao subscritor, um editor de pronúncias onde é possível editar e corrigir a pronúncia de determinadas palavras, sendo esta uma funcionalidade que se revela bastante vantajosa, nos casos em que, por exemplo, seja necessário mencionar palavras num idioma diferente ou em que a pronúncia das palavras se releve diferente do habitual.

2.3 Gravação e Reprodução sonora

A gravação de voz, também denominada de modo mais geral como gravação sonora, consiste num processo pelo qual é possível capturar e armazenar a informação de áudio existente através de um determinado sistema de captação de áudio, de modo a que a mesma possa ser reproduzida mais tarde por meio de um sistema de reprodução sonora [84].

Os primórdios da gravação sonora remontam a meados do século XIX, tendo passado por várias fases e evoluções desde então, até aos dias de hoje, onde é já uma tecnologia estabelecida e massificada. Atualmente, a gravação e reprodução sonora fazem parte do ambiente que nos rodeia, encontrando-se em praticamente todos os meios de comunicação social, cinema, música, etc.

2.3.1 Revisão Histórica

Os primeiros registos da gravação sonora datam de 1857, quando o inventor francês *Édouard-Léon Scott de Martinville* inventou o fonoautógrafo, um dispositivo capaz de captar as ondas de som transmitidas pelo ar e registar as suas oscilações em papel. No entanto, ainda não era possível traduzir os registos obtidos novamente para som, o que levou a que este instrumento não tenha alcançado a visibilidade dos que surgiram a seguir [85].

Cerca de 20 anos depois, o famoso inventor *Thomas Edison*, numa tentativa de melhorar o seu trabalho no campo da telefonia e telegrafia, introduziu o fonógrafo. O modo de captura de som deste mecanismo foi fortemente inspirado pelo conceito do fonoautógrafo de *Martinville*. No entanto, foi utilizada uma abordagem diferente para registar o som, de modo a que o mesmo pudesse ser novamente reproduzido: registavam-se as vibrações provocadas pelo som num cilindro embrulhado em papel de parafina, para que fosse possível reproduzi-las novamente através de uma agulha que lia e transmitia essas mesmas vibrações para um transdutor[11] que, por sua vez, traduzia as vibrações para áudio [86].

Figura 10 – Fonógrafo de *Edison* [87]

À medida que o tempo passava, estes mecanismos atraiam cada vez mais utilizadores que se iam rendendo à tecnologia da gravação/reprodução sonora para fins lúdicos. Isto levou a que grande parte dos utilizadores, depressa se apercebesse que o método de armazenamento (em papel de cera) do fonógrafo de *Edison* era bastante frágil e limitado. *Edison*, na altura envolvido em projetos de maior dimensão (a lâmpada incandescente e a eletrificação da cidade) acabou por deixar estagnar o desenvolvimento do fonógrafo, o que criou uma oportunidade para *Alexander*

[11] Transdutor – Dispositivo que converte informações provenientes de fenómenos físicos para sinais detetáveis (como por exemplo sinais elétricos, sonoros, etc.).

Graham Bell introduzir um novo método de armazenamento: um cilindro de cera que suportava uma utilização repetitiva e com uma maior capacidade de armazenamento. Este método deu origem a um novo dispositivo similar ao fonógrafo, o grafofone, que, no entanto, apresentava maiores vantagens como por exemplo, a gravação e reprodução automática. Com a chegada da eletricidade, este instrumento foi mais tarde equipado com motores elétricos e o seu método de armazenamento foi alterado para o formato de discos planos, fruto da descoberta de um outro inventor, *Emile Berliner*, renomeando este dispositivo para Gramofone [88].

Apesar do Gramofone persistir no mercado, com o passar do tempo foram surgindo novos métodos de captação e reprodução do som, resultantes da inovação proveniente dos campos da eletricidade e magnetismo. É então que, por volta dos anos 30, surge um novo método de gravação magnética que consiste em magnetizar fitas revestidas de óxido de ferro com o sinal analógico recebido, ficando o mesmo registado de forma permanente nesse material, podendo ser depois lido e reproduzido através de um dispositivo capaz de descodificar os dados magnetizados [89]. Ao mesmo tempo que ia surgindo a gravação magnética, iam-se substituindo também os métodos tradicionais de captura e reprodução de som, que geralmente consistiam numa estrutura em forma de cone, por novas abordagens que recorriam à eletricidade. Surgiram assim o microfone e as colunas de som, que permitiam uma captura e reprodução sonora ainda mais clara [90].

Entretanto, em 1963, surgiu um dos instrumentos que mais prevaleceram no mundo da gravação e reprodução sonora: a *cassete*. Lançado pela *Philips*, este dispositivo utilizava a tecnologia de gravação magnética e veio revolucionar o mercado que se vivia na altura, dando aso ao desenvolvimento de novos instrumentos de reprodução e gravação sonora como foi o caso dos *walkmans*, autorrádios equipados com leitor de cassete, sistemas caseiros de reprodução e gravação de cassetes, etc. [91].

Com o passar do tempo e com o despontar da era digital, começam a surgir alternativas mais práticas e eficientes que as anteriores, como foi o caso dos *Compact Disks*, usualmente conhecidos pela sua sigla, os CD's. Esta tecnologia resultou da união entre duas empresas, a Philips e a Sony, e desde cedo começou a conquistar o mercado, causando o declínio na utilização dos outros meios de armazenamento sonoro devido às vantagens que apresentava em relação aos métodos anteriores. A leitura dos dados armazenados em CD's era realizada através de *lasers*, não existindo contacto direto com a superfície, o que removia completamente os ruídos causados pelos métodos de leitura que até à época se utilizavam. A mudança de faixa musical era também direta, não era necessário percorrer toda a superfície até chegar a determinada música, incluindo ainda uma maior capacidade de armazenamento que superava a maioria dos métodos anteriores. Todas as estas vantagens levaram a que esta tecnologia perdurasse até hoje, ainda que atualmente esteja a cair em desuso graças a novos e mais práticos métodos de compactação e armazenamento digital que não dependem inteiramente de dispositivos físicos (como por exemplo os leitores de cassetes ou CD's) [92].

Nos dias de hoje, após grandes evoluções tecnológicas, nomeadamente ao nível de soluções de armazenamento e partilha de informação, a grande maioria do conteúdo sonoro é produzido e consumido de um modo totalmente digital. Tudo começou com o início da globalização da internet, altura em que surgia o formato *mp3* (*MPEG-1/2 Audio Layer 3*). A tecnologia inerente a este formato permitia comprimir ficheiros de áudio volumosos para ficheiros mais pequenos, suprimindo alguns bits de informação, de modo a facilitar a sua partilha e transmissão através da internet e de outros canais digitais [93]. Para além do formato *mp3*, é importante referir que ao longo do tempo foram surgindo também outros formatos de compressão de ficheiros áudio, como por exemplo o *wma* da Microsoft, similar ao *mp3* ou o *flac*, formato este que utiliza um algoritmo que permite comprimir ficheiros de áudio sem causar nenhuma perda de informação, entre outros [94]. Não tardaram a surgir equipamentos portáteis dedicados à leitura deste tipo de áudio, os leitores de *mp3*, que permitiam armazenar e ouvir centenas ou até milhares de músicas num pequeno dispositivo que facilmente cabia no bolso. No entanto, foram caindo em desuso à medida que surgiam os smartphones e as operadoras começaram a fornecer planos de dados que permitem uma utilização intensiva da internet. Neste momento, vive-se a era do *streaming*[12], onde é possível aceder a qualquer conteúdo sonoro, a qualquer momento, desde que exista uma ligação à internet, através de plataformas como, por exemplo o *Youtube*, *Spotify*, *Apple Music*, etc [95].

2.3.2 Software de Gravação Sonora

Estão disponíveis no mercado atual, inúmeras soluções para gravação sonora. Devido a essa extensa oferta, e dado que as funcionalidades oferecidas pelas mesmas acabam por ser bastante similares, apresentam-se de seguida apenas algumas das ferramentas mais populares.

2.3.2.1 Audacity

Esta é uma das mais populares ferramentas gratuitas para gravação e edição sonora. É uma escolha habitual entre os utilizadores que pretendem iniciar-se neste tipo de plataformas, dado que lhes permite uma experiência aproximada daquela que é oferecida pelas ferramentas profissionais, sem que lhes sejam necessários investimentos avultados.

Esta plataforma é compatível com os principais sistemas operativos existentes (excluindo-se as plataformas móveis), oferecendo uma interface gráfica bastante simplificada, bem organizada e fácil de dominar. O utilizador tem à sua disposição uma larga panóplia de controlos e ferramentas de edição de áudio, que lhe permite realizar tarefas como por exemplo, ajustes do tom, graves, agudos, etc. No entanto, esta ferramenta também apresenta algumas desvantagens, sendo que, a

[12] Streaming – Metodologia de receção de ficheiros multimédia sem interrupções através da internet. À medida que o dispositivo vai recebendo os dados, estes vão sendo processados e transmitidos.

que mais se destaca é a edição destrutiva, que não permite a reversão das alterações efetuadas ao ficheiro áudio trabalhado [96].

2.3.2.2 Adobe Audition

Considerada uma das mais completas ferramentas deste género no mercado, o *Adobe Audition* oferece uma interface limpa e personalizável, em conjunto com poderosas funcionalidades de edição de áudio. Uma das grandes características desta aplicação é a capacidade da captura de som a partir de várias fontes de áudio de uma só vez, sendo possível a sua edição separadamente.

Para além destas funcionalidades, o *Audition* oferece uma larga diversidade de efeitos e restauração de áudio. É possível, por exemplo, importar uma gravação antiga proveniente de um leitor de vinil e retirar-lhe os ruídos causados pela fricção da agulha no disco, sem causar perdas no áudio principal. Esta ferramenta conta ainda com suporte para a maioria dos formatos de ficheiros áudio, permitindo a sua conversão.

A grande desvantagem desta ferramenta, centra-se na necessidade de pagamento por subscrição que, ao final de alguns meses acaba por ser bastante dispendioso [97].

2.3.2.3 WavePad

O *WavePad* é outra das mais populares aplicações para edição sonora, o qual conta com uma das maiores ofertas em termos de efeitos e filtros áudios para utilizar na edição. Conta ainda com compatibilidade com a maioria dos ficheiros áudio, permitindo ao utilizador trabalhar com o seu formato preferido.

Esta ferramenta oferece funcionalidades como por exemplo, análise de frequências, restauração de áudio, conversão de múltiplos ficheiros de áudio em simultâneo, permitindo ainda o *download* de várias amostras de áudio úteis a partir da sua biblioteca.

Apesar de oferecer uma quantidade considerável de manuais e ferramentas de suporte no seu *website*, caso o utilizador pretenda um atendimento mais personalizado ou especializado, terá que pagar por esse serviço, o que acaba por se tornar a desvantagem que mais se destaca neste produto [98].

3 Ferramentas e Tecnologias

Neste capítulo descrevem-se as diversas ferramentas, linguagens de programação e tecnologias que foram adotadas no decorrer das várias fases de desenvolvimento associadas ao estágio. Foi dada total liberdade ao estagiário para escolher as mais adequadas a cada prova de conceito ou desenvolvimento, desde que as mesmas fossem compatíveis e integráveis com a solução final (o STA). O estagiário foi também instruído sobre a política de preferência por ferramentas gratuitas, sendo que, durante a análise e seleção de ferramentas e tecnologias, procurou sempre as que estivessem de acordo com estes parâmetros.

3.1 Ferramentas de Desenvolvimento e Suporte

3.1.1 Netbeans

É um IDE[13] gratuito e de código aberto disponibilizado pela Sun Microsystems, detida agora pela Oracle, que permite desenvolver aplicações para *desktop*, *web* ou *mobile* multiplataforma. Originalmente este ambiente de desenvolvimento foi projetado para linguagem Java, no entanto, hoje em dia já suporta uma grande variedade de linguagens de programação, tais como PHP (*Hypertext Preprocessor*), C/C++, JavaScript, etc. [99].

Este IDE foi utilizado para o desenvolvimento e integração das provas de conceito que envolviam a linguagem JAVA. Os motivos que levaram o estagiário à adoção desta ferramenta foram a natureza gratuita da mesma e o conhecimento adquirido durante o percurso académico.

3.1.2 Visual Studio Code

É um editor de código desenvolvido pela Microsoft, disponibilizado de forma gratuita e que inclui suporte para *debugging*[14], *intelligent code completion*[15], *code refactoring*[16], permitindo

[13] IDE – do Inglês Integrated Development Environment ou Ambiente de Desenvolvimento Integrado, é uma aplicação para computador onde estão reunidas determinadas ferramentas ou funcionalidades que apoiam e agilizam o processo de desenvolvimento de software.
[14] *Debugging* – é o processo de encontrar e resolver problemas ou defeitos numa aplicação de software.
[15] *Intelligent code completion* – é uma funcionalidade existente em várias ferramentas de desenvolvimento que permite completar automaticamente determinadas linhas de código, recorrendo a uma base de dados que vai sendo gerada durante o desenvolvimento, que guarda classes, nomes de variáveis, nomes de funções, etc. e que as vai apresentando consoante a ação executada pelo programador, diminuindo assim a probabilidade de ocorrerem erros de sintaxe.
[16] *Code refactoring* – é o processo de reestruturar o código existente de modo a melhorar a sua estrutura interna sem alterar o seu comportamento externo.

ainda efetuar tarefas de Git diretamente a partir do editor. Esta ferramenta suporta uma larga lista de linguagens de programação, tais como JavaScript, PHP, C#, C++, Java, JSON (*JavaScript Object Notation*), *Python*, etc. [100].

Este IDE foi utilizado durante a fase de integração dos desenvolvimentos obtidos pelas provas de conceito na solução final (o STA). Esta era a ferramenta utilizada uniformemente por todos os membros da equipa, o que levou o estagiário a adotar a mesma.

3.1.3 Atom

É um editor de código desenvolvido pela GitHub que suporta *plug-ins*[17] Node.js, que à semelhança do anterior, também permite efetuar tarefas de Git diretamente a partir do editor. Este inclui a funcionalidade de *intelligent code completion* e possibilita ainda a organização dos ficheiros de código por pastas ou pacotes de modo quase similar ao ambiente de um IDE. Esta ferramenta é também de código aberto e gratuita, bastante fácil de personalizar e suporta também uma extensa lista de linguagens de programação [101].

Esta ferramenta foi utilizada durante o desenvolvimento das provas de conceito que envolviam JavaScript. O estagiário optou por esta ferramenta pelo facto da mesma ser gratuita e fornecer diversas *frameworks* úteis para desenvolvimento *web*.

3.1.4 Notepad++

É um dos editores de código mais antigos e mais adotados no mundo da programação pela sua simplicidade, rapidez de execução e "leveza", bem como pelo facto de suportar a maioria das linguagens de programação mais populares. Esta ferramenta foi escrita em C++ por Don Ho e disponibilizada gratuitamente à comunidade. As suas características incluem as funcionalidades de *intelligent code completion*, gravação de macros[18] e suporte para um largo número de dialetos que podem ser adicionados à ferramenta como *plug-ins* [102].

Esta ferramenta foi utilizada durante todas as fases de desenvolvimento que decorreram durante o estágio. Esta ferramenta era utilizada para, de forma rápida, efetuar pequenos desenvolvimentos ou testes que não exigissem a utilização de um IDE.

3.1.5 Node.js

É uma plataforma direcionada para o desenvolvimento de aplicações *web*, desenvolvida originalmente por Ryan Dahl, utilizando o JavaScript Engine V8[19] da Google. Este ambiente de desenvolvimento proporciona um modo fácil de construir aplicações rápidas e escaláveis, focando-

[17] *Plug-ins* – são extensões que permitem adicionar determinadas funcionalidades a um software já existente.
[18] *Macros* – permitem automatizar sequências de ações pré-definidas
[19] *JavaScript Engine V8* – motor *JavaScript* desenvolvido pela *Google* para o seu browser *Google Chrome*

se na *performance*, no baixo consumo de memória e numa abordagem baseada em eventos. O Node.js tem a particularidade de, ao contrário das aplicações de servidor mais "tradicionais" (como por exemplo o *Apache*), ser possível implementar JavaScript na implementação de componentes num servidor de uma aplicação *web* [103].

Esta plataforma foi utilizada durante a fase de desenvolvimento / integração dos componentes de reconhecimento e gravação de voz na solução final. Esta é a plataforma na qual o STA está assente.

3.1.6 Npm

É um gestor de dependências para o Node.js que permite aos programadores de JavaScript, partilhar facilmente os seus módulos de código que podem depois ser reutilizados por outros programadores. Estes módulos poderão ser ferramentas ou funcionalidades úteis que ao serem adicionados a determinados projetos, evitam por exemplo, a necessidade de se desenvolver algumas funcionalidades que deste modo já estão implementadas e prontas a integrar no projeto [104].

Foi utilizado durante a fase de desenvolvimento da solução final e permitia a partilha e instalação de módulos necessários ao desenvolvimento dos componentes da mesma.

3.1.7 Vagrant

É uma ferramenta que permite a construção de um ambiente de desenvolvimento pronto a usar. Isto é, ao invés de se ter que configurar cada máquina para determinado projeto, configura-se apenas um ambiente virtual do Vagrant que é utilizado por todos os membros da equipa. Uma das maiores vantagens desta ferramenta encontra-se no facto de se poder partilhar o *workspace*[20] local do projeto com o ambiente virtualizado. Ou seja, por exemplo, o código pode ficar alojado localmente na máquina e, no entanto, pode ser utilizado no ambiente virtualizado já configurado [105].

Esta ferramenta foi utilizada durante a fase de desenvolvimento e integração dos componentes de reconhecimento e gravação de voz. Permitia ao estagiário desenvolver, simular e testar as funcionalidades que iam sendo produzidas.

3.1.8 Git

É uma ferramenta grátis e de código aberto para controlo de versões, desenhada de modo a ser capaz de manipular projetos independentemente da sua dimensão, com eficiência e rapidez.

[20] *Workspace* – grupo de ficheiros de código que constituem o projeto

O Git é projetado para facilitar o desenvolvimento de projetos em equipa, para que seja viável o trabalho simultâneo de membros diferentes no mesmo projeto, sem que exista o risco de determinadas alterações serem sobrescritas [106].

Esta era a ferramenta de controlo de versões\repositório utilizada pela equipa durante a fase de desenvolvimento.

3.2 Linguagens de Programação

3.2.1 HTML (HyperText Markup Language)

É a principal e mais básica linguagem da *internet*. O HTML utiliza um conjunto universal pré-definido de elementos ou *tags*[21] que servem como instruções aos *browsers* para que os mesmos, ao interpretá-las, construam a página *web*. Esta linguagem segue uma estrutura que separa o conteúdo (por exemplo imagens, texto, etc.) das instruções que coordenam esses elementos [107].

Esta linguagem foi utilizada no desenvolvimento das diversas *interfaces web* das provas de conceito.

3.2.2 CSS (Cascading Style Sheets)

É a tecnologia responsável por definir o modo como os componentes de uma determinada página *web* são exibidos. Esta tecnologia surgiu da necessidade de contornar o aspeto "cru" de uma simples página *web* em HTML, aplicando-lhe estilos e cor. Hoje em dia já se encontra na terceira versão (CSS3) que está ainda em processo de estandardização no W3C[22] (*World Wide Web Consortium*), [108].

À semelhança do HTML, foi utilizada no desenvolvimento das *interfaces web* das provas de conceito.

3.2.3 JavaScript

É uma linguagem de programação orientada a objetos direcionada para o desenvolvimento de páginas *web*, a qual tem como principal característica o facto de ser executada do lado do cliente, utilizando os recursos dessa máquina e poupando processamento adicional do lado do servidor. O JavaScript é utilizado tipicamente para manipular o HTML e CSS de uma página w*eb*,

[21] *Tags* – Em HTML, uma *tag* é utilizada para criar um elemento.
[22] *W3C* – comunidade internacional responsável por desenvolver *standards* para a *Web*.

fazendo com que as mesmas se tornem mais interativas e dinâmicas. No entanto, esta linguagem de programação tem-se tornado tão versátil, que já é utilizada em outro tipo de ambientes como por exemplo o já referido exemplo do Node.js [109].

Esta linguagem de programação foi utilizada no desenvolvimento da lógica dos componentes de reconhecimento e síntese de voz. É também a linguagem de programação onde a ferramenta utilizada pelo módulo de reconhecimento de voz (WEB Speech API) está assente.

3.2.4 Java

É uma linguagem de programação orientada a objetos que tem como principais atributos, a sua elevada portabilidade, simplicidade e o vasto número de bibliotecas com que a mesma é distribuída. O Java, ao contrário de outras linguagens de programação convencionais que são compiladas para código nativo, é executado numa máquina virtual (JVM) que deverá estar instalada no sistema. Deste modo, independentemente do sistema operativo instalado, é possível correr aplicações Java sem problemas de compatibilidade [110].

Esta linguagem foi utilizada no desenvolvimento das provas de conceito para gravação de voz e integração das mesmas no STA. As principais razões para a utilização do Java passaram pela experiência já adquirida na mesma pelo estagiário em ambiente letivo e pela sua vasta compatibilidade nos sistemas operativos existentes.

3.2.5 JSON

É um formato leve para troca de dados baseado em JavaScript, de fácil interpretação e geração pelas máquinas e, de igual modo, de fácil leitura e escrita pelo ser humano. Esta tecnologia, utilizando convenções similares às das linguagens baseadas em C e outras derivadas, torna-se completamente independente da linguagem em que é inserida, facilitando assim por exemplo, a integração e comunicação de módulos que sigam linguagens diferentes [111].

Foi utilizado como modo de geração de objetos para a comunicação entre aplicações Java e JavaScript (neste caso entre o código responsável pela interface do STA e a aplicação Java para gravação de voz), aproveitando a vantagem de ser interpretado por ambas as linguagens.

3.2.6 Angular2

É uma *framework*[23] concebida para tornar as páginas *web* mais dinâmicas. Esta tecnologia permite alargar a tradicional sintaxe HTML para novas e mais completas instruções, permitindo uma melhor manipulação dos componentes da página, trazendo novas funcionalidades que até então eram impossíveis de implementar recorrendo apenas ao HTML. Esta tecnologia permite

[23] *Framework* – fornece funcionalidades adicionais a um sistema ou ferramenta existente

também reduzir a quantidade e a complexidade do código que se utiliza para construir determinadas funções, quando comparada com as tecnologias mais tradicionais (como por exemplo HTML, CSS e JavaScript) [112].

Sendo esta uma das tecnologias na qual o STA está assente, o estagiário efetuou pequenos desenvolvimentos na mesma, ao nível gráfico da aplicação.

3.2.7 TypeScript

É uma linguagem de programação baseada em JavaScript desenvolvida pela Microsoft, direcionada tanto para o desenvolvimento de aplicações do lado do cliente como do lado do servidor (por exemplo Node.js). Esta linguagem foi concebida para auxiliar na construção de aplicações *web* extensas, as quais se têm vindo a tornar cada vez mais comuns e com um crescente grau de complexidade. O TypeScript, de certo modo, reinventa o JavaScript aumentando a modularidade e simplicidade do código, trazendo novas funcionalidades e reduzindo a complexidade necessária para a implementação de determinadas funcionalidades. O código TypeScript compila para código JavaScript, o que permite a sua interpretação por todos os *browsers* e torna ainda possível a utilização de código JavaScript em simultâneo [113].

Mais uma vez, esta é uma das linguagens de programação utilizadas no desenvolvimento do STA. Foi utilizada durante a integração das funcionalidades de reconhecimento de voz no STA, neste caso traduzindo grande parte dos desenvolvimentos escritos em JavaScript para TypeScript.

3.3 Bibliotecas e API's[24]

3.3.1 WebSockets

É um sistema que permite a existência de uma sessão interativa de comunicação bidirecional entre a aplicação que corre no *browser* do cliente e um servidor. Com esta API torna-se possível que a ligação entre o cliente e o servidor esteja constantemente ativa até que um dos intervenientes a interrompa e, deste modo, conseguem-se enviar mensagens para o servidor ou receber respostas do mesmo baseadas em eventos, sem existir necessariamente uma solicitação explícita por parte do cliente [114].

Este sistema foi utilizado de modo a permitir a comunicação / partilha de objetos JSON entre a aplicação *web* do STA e a aplicação Java para gravação de voz que corre em *background*.

[24] API – conjunto de rotinas, protocolos ou funcionalidades pré-definidos que auxiliam o desenvolvimento de aplicações

É o modo pelo qual se enviam as instruções (Iniciar Gravação, Parar Gravação, etc.), entre as aplicações.

3.3.2 Gson

É uma biblioteca Java desenvolvida pela Google que tanto permite converter objetos Java na sua representação em JSON, como também converter s*trings* JSON nos seus objetos Java equivalentes. Esta biblioteca apresenta características bastante vantajosas que se focam numa implementação bastante simples das suas funcionalidades, possibilita a conversão para JSON de objetos Java já existentes, mesmo que não exista acesso ao seu código, e permite ainda personalizar a representação dos objetos em JSON [115].

Esta biblioteca foi utilizada devido à sua natureza gratuita e fácil de utilizar. Graças à mesma, é possível converter os objetos JSON com as instruções, enviados a partir da aplicação *web*, para objetos interpretáveis pela aplicação Java para gravação de voz.

3.3.3 Web Speech API

A Web Speech API é uma API para JavaScript que tem vindo a ser desenvolvida desde 2012 por um grupo de investigadores e colaborados da Google e Mozilla. Esta permite incorporar dados de voz em aplicações *web*, sendo constituída por dois componentes principais: o Reconhecimento de Voz e a Síntese de Voz. Tem já disponíveis diversas interfaces e eventos para tratar alguns acontecimentos principais destes dois componentes (transcrição de voz em texto e vice-versa, alternativas em caso de transcrição com pouca percentagem de confiança, erros gerados, etc.), os quais permitem aumentar, por exemplo, o nível de acessibilidade e mecanismos de controlo das páginas *web*. As maiores vantagens desta API estão no facto da mesma ser gratuita e de fácil integração no JavaScript, ter suporte para língua Portuguesa de Portugal (apesar de existir apenas no módulo de reconhecimento de voz), não ser necessária a utilização de um ambiente de desenvolvimento especializado (não é necessário treino nem manipulação direta de modelos de linguagem ou modelos acústicos) e recorrer aos algoritmos e bases de dados da Google para a geração de resultados [116].

Esta API foi utilizada, mais uma vez, devido a ser uma tecnologia gratuita, sem necessidade de investimento em treino, e que reúne a maioria das condições exigidas para o reconhecimento de voz a integrar no STA, nomeadamente o suporte para a língua Portuguesa e a fácil integração em aplicações *web*.

3.3.4 Voice RSS

A API Voice RSS oferece um modo bastante acessível para conversão de conteúdos em texto para voz. Recorrendo a poucas linhas de código, é possível transcrever remotamente (num servidor dedicado) alguns parágrafos de texto de forma bastante satisfatória. Esta API oferece suporte para 26 línguas (incluindo o Português de Portugal que não está disponível na Web Speech API) e permite ainda personalizar as características da resposta de áudio obtida em termos de velocidade do discurso, *codecs*[25] de áudio (*mp3, WAV*, etc.) e formato que determina a qualidade do áudio. Apesar de ter várias modalidades que são pagas, esta API também está disponível de forma gratuita, no entanto restringida a 350 pedidos diários por conta de utilizador [117].

Esta API foi utilizada no âmbito das provas de conceito que envolviam a síntese de voz. As razões que levaram à adoção da mesma foram essencialmente a não existência do Português de Portugal nas funcionalidades de síntese de voz da Web Speech API e por ser praticamente a única forma gratuita existente (mas limitada) para síntese de voz, possível de satisfazer o requisito anterior.

3.3.5 JAVE (Java Audio Video Encoder)

É uma biblioteca Java gratuita e de código aberto desenvolvida pela Sauron Software que permite converter áudio e vídeo de um determinado formato para outro. Com esta API é possível por exemplo, converter ficheiros áudio que se encontrem em formato WAV para *mp3*, tarefa esta impossível de realizar recorrendo apenas às bibliotecas Java nativas. Outra das grandes vantagens desta biblioteca encontra-se no facto da mesma ser multiplataforma, ou seja, compatível com a maioria dos sistemas operativos [118].

Esta biblioteca foi utilizada no desenvolvimento e integração dos resultados da prova de conceito para gravação de voz no STA. O facto de a mesma ser gratuita e possibilitar a conversão de grandes formatos áudio (como o WAV) para formatos mais pequenos não suportados nativamente pela linguagem Java, foram as principais razões para sua adoção.

[25] *Codecs* – servem para codificar ou descodificar dados digitais. São bastante utilizados por exemplo, na conversão de áudio e vídeo para formatos mais leves.

4 Metodologia de Trabalho

Neste capítulo apresenta-se uma breve descrição sobre a metodologia de desenvolvimento adotada durante o estágio. Descreve-se também o modo pelo qual a mesma foi adaptada à realidade das diferentes fases do estágio. Finalmente são descritas as responsabilidades de cada um dos elementos da equipa e detalhadas as durações de cada uma das etapas do estágio.

4.1 Metodologia Scrum

A metodologia de desenvolvimento adotada pela entidade acolhedora é a **Scrum**. Esta consiste num processo de desenvolvimento ágil, iterativo e incremental, bastante utilizado em projetos cujos objetivos estejam sujeitos à mudança.

Existem 3 papéis principais nas equipas Scrum:

Membro da Equipa: responsável pelo desenvolvimento da solução e implementação dos objetivos definidos

Product Owner: responsável pela definição e priorização dos requisitos. Geralmente este é um papel desempenhado pelo cliente.

Scrum Master: responsável pela gestão do projeto. Tem como papel principal, assegurar a prática correta e eficiente da metodologia, assim como remover os obstáculos com que a equipa se depara.

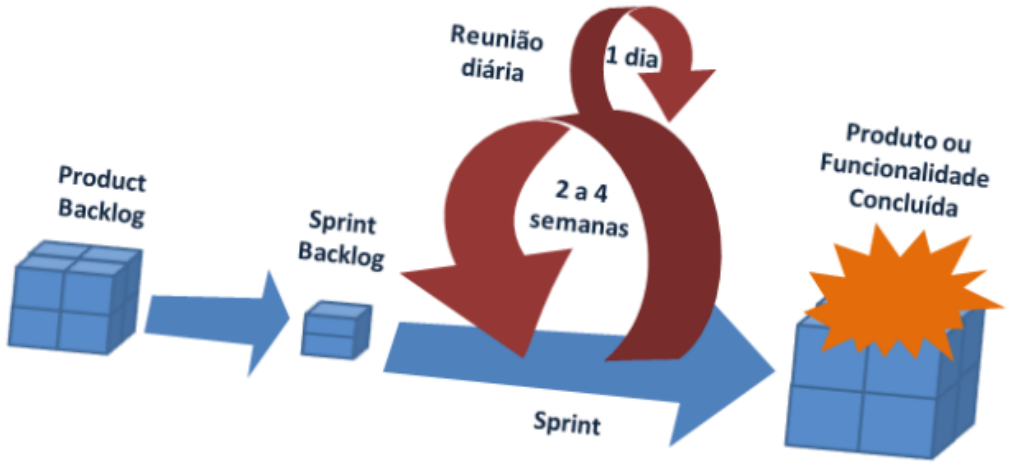


Figura 11 – Esquema da Metodologia Scrum [119]

Esta metodologia baseia-se em ciclos de desenvolvimento denominados por *Sprints*, que tipicamente podem durar de uma semana a um mês. Durante estes ciclos trabalha-se para atingir os objetivos, ou *User Stories*, definidos no início de cada *Sprint* numa lista denominada por *Sprint Backlog*. Estes objetivos fazem parte e são retirados de uma lista mais extensa, o *Product* Backlog, onde estão definidas todas as *User Stories* a implementar durante as diversas *Sprints* que compõem as fases de desenvolvimento do produto.

Durante a *Sprint*, a equipa reúne todos os dias com o *Scrum Master*, no mesmo horário, de modo a realizar um resumo sobre o que foi feito no dia anterior, o que se pretende fazer no dia atual e eventuais dificuldades ou impedimentos detetados.

No final de cada *Sprint*, a equipa demonstra os resultados alcançados ao *Product Owner* e realiza uma *Sprint Retrospective* onde discute o que correu bem durante a *sprint*, o que pode eventualmente ser melhorado e aquilo que se comprometem a melhorar durante a próxima *sprint* [119].

Durante a maioria do tempo de estágio, utilizou-se uma metodologia *Scrum* similar à descrita anteriormente. Nos primeiros meses, em que o estagiário estava encarregue de investigar os temas abordados pelo estágio e testar diversas soluções, não existia ainda requisitos totalmente definidos, pelo que, à exceção das reuniões frequentes com a equipa e reuniões de ponto de situação com o *Product Owner*, não se seguia exatamente esta metodologia.

Logo após a apresentação dos resultados da investigação, começaram a definir-se requisitos e já existiam objetivos concretos. A partir deste momento e até ao final do estágio já foi possível trabalhar sob a forma da metodologia *Scrum*.

4.2 Equipa

A seguinte tabela detalha os membros e funções correspondentes da equipa onde o estagiário esteve integrado.

Tabela 2 – Estrutura da equipa

Membro	Função
Jorge Coimbra	*Product Owner*
Marta Cunha	*Scrum Master*
Ivo Santos	Tutor
João Fidalgo	*Developer*
Telmo Raimundo	Estagiário
Bruno Oliveira	Estagiário

4.3 Fases de Desenvolvimento

A tabela seguinte ilustra as diferentes etapas de investigação e implementação e respetiva duração.

Tabela 3 – Duração das etapas durante o estágio

Fase	Duração
Investigação sobre reconhecimento de voz	Entre Dezembro/2015 e Fevereiro/2016
Prova de Conceito 1: Conversão de Voz para Texto	Entre Fevereiro/2016 e Março/2016
Prova de Conceito 2: Comandos por Voz / Converter Texto para Voz (incluindo investigação sobre síntese de voz)	Entre Março/2016 e Abril/2016
Desenvolvimento de Aplicação *Java* para Gravação de Voz	Abril/2016
Prova de Conceito 3: Comunicação entre Aplicação *Java* e Aplicação Web via *websockets*	Maio/2016
Integração do Módulo de Reconhecimento de Voz (Comandos por Voz)	Julho/2016
Integração do módulo de Gravação de Voz	Entre Julho/2016 e Agosto/2016

5 Desenvolvimento

No decorrer do estágio, foi proposto ao estagiário, o desenvolvimento de diversas provas de conceito de modo a testar e explorar as várias ferramentas e tecnologias encontradas. Ao todo, foram realizadas 3 provas de conceito, sendo que a grande maioria do trabalho investido foi depois convertido para a solução final. Nas secções seguintes explica-se mais detalhadamente o trabalho realizado nessas provas de conceito, assim como a integração dos resultados obtidos a partir das mesmas na solução final.

Ao contrário da habitual organização deste tipo de capítulos que apresentam os detalhes do desenvolvimento de uma aplicação muitas vezes já previamente definida, devido à natureza experimental e ao carácter de investigação levado a cabo através de provas de conceito ao longo do período de estágio, este capítulo segue uma ordem cronológica determinada por essas provas de conceito culminando nos detalhes da integração do resultado das mesmas na solução final.

5.1 Prova de Conceito 1: Converter Voz para Texto

5.1.1 Objetivos

Esta prova de conceito teve como objetivo central demonstrar as funcionalidades principais do reconhecimento de voz da Web Speech API. Durante o desenvolvimento desta prova de conceito, o estagiário focou-se em testar e explorar a API em ambiente *web*, analisar as várias funcionalidades e implementar várias abordagens de reconhecimento de voz, dando uso aos diferentes métodos que esta oferece.

5.1.2 Investigação

Para utilizar o reconhecimento de voz nesta API, recorre-se a uma função denominada “SpeechRecognition”. Esta permite reconhecer a voz do utilizador através de um *input* de áudio (por exemplo um Microfone) e transcrever o discurso para texto.

Quando invocada, esta função devolve um objeto onde constam os resultados em bruto do reconhecimento de voz. Posteriormente, é possível aplicar a esse objeto diversos eventos e propriedades que nos permitem manipular a informação obtida pelo serviço de reconhecimento de voz. Reúnem-se, de seguida, os eventos e propriedades mais relevantes:

SpeechRecognitionResultList

Permite aceder à lista de possíveis palavras que o reconhecimento de voz detetou para cada resultado. Aplicando-lhe a propriedade "length", é possível saber quantas palavras possíveis existem para cada resultado ("SpeechRecognitionResult").

Interim/Final

Através da propriedade *Interim* é possível aceder aos resultados intermédios do reconhecimento de voz, à medida que os mesmos vão sendo recebidos. No entanto estes ainda não foram sujeitos a análise de contexto pelo que existe uma maior probabilidade de os mesmos não serem corretos. Já no caso da propriedade *Final*, a mesma só apresenta os resultados finais com maior probabilidade de acerto, contudo a obtenção destes resultados é mais morosa do que a anterior, já que o sistema terá que processar todo o excerto escutado e sujeitá-lo a análise de contexto de modo a obter os resultados mais corretos.

Transcript

Esta propriedade é utilizada para obter a *string* onde constam as palavras que o reconhecimento de voz transcreveu para texto.

Onaudiostart / Onaudioend

São eventos associados ao início e término da deteção da voz do utilizador. Geralmente utilizados para invocar e terminar o serviço de reconhecimento de voz quando os mesmos ocorrem.

Onsoundstart / Onsoundend

São eventos bastantes similares aos anteriores, no entanto não se limitam apenas a detetar a voz do utilizador, mas todos os sons recebidos.

Onspeechstart / Onspeechend

São eventos ligados ao início/fim do serviço de reconhecimento de voz. Estes eventos são despoletados apenas quando a voz do utilizador é efetivamente reconhecida e possível de ser transcrita pelo serviço.

Onresult

É um evento desencadeado no momento em que o serviço de reconhecimento de voz devolve os seus resultados, ou seja, é devolvida pelo menos uma palavra reconhecida com sucesso.

Onnomatch

É um evento despoletado quando o serviço de reconhecimento de voz devolve uma palavra ou frase, cuja percentagem de confiança é baixa.

SpeechRecognitionAlternative

Representa uma das palavras alternativas que foi reconhecida pelo serviço. Esta palavra será uma alternativa à palavra reconhecida, no entanto com uma menor percentagem de confiança. Aplicando a propriedade *confidence* a esta alternativa, é possível aceder à estimativa de confiança que o serviço atribuiu a esta palavra.

SpeechGrammarList

Cria uma lista exclusiva de palavras que o sistema pode reconhecer. Todas as palavras detetadas que não pertencerem a esta lista não farão parte dos resultados.

Start / End

Despoleta e termina o serviço de reconhecimento de voz.

5.1.3 Implementação

A ideia por detrás desta prova de conceito passou por simular de forma simples o ecrã de uma sala de chat onde seria possível enviar mensagens para o ecrã principal através do reconhecimento de voz. Este ecrã consistia numa janela principal para onde eram enviadas as mensagens escritas manualmente ou através do serviço de reconhecimento de voz, uma janela mais pequena onde era possível escrever ou observar o resultado obtido através do reconhecimento de voz, um botão *Enviar* para remeter o texto para a janela principal, um botão *Eliminar* para limpar a janela principal e um botão *Clique para falar/Clique para parar*, para iniciar e terminar o serviço de reconhecimento de voz.

Foram implementadas 3 abordagens diferentes que serviram para demonstrar algumas vertentes da utilização do reconhecimento de voz através da API.

Figura 12 – Interface gráfica da primeira prova de conceito

5.1.3.1 Reconhecer Voz e Apresentar os Resultados no Final

Esta implementação focou-se apenas nos resultados finais do reconhecimento de voz, ou seja, o utilizador pressiona o botão *Clique para Falar*, inicia o seu discurso e, quando termina pressiona o botão *Clique para Parar*. Nesse momento são-lhe apresentados os resultados finais do reconhecimento de voz na caixa de edição de texto.

```
var recognizing;
var recognition = new webkitSpeechRecognition();
recognition.continuous = true;
reset();
recognition.onend = reset();

recognition.onresult = function (event) {
  for (var i = event.resultIndex; i < event.results.length; ++i) {
    if (event.results[i].isFinal) {
      messageBox.value += event.results[i][0].transcript;
    }
  }
}

function reset() {
  recognizing = false;
  button.innerHTML = "Clique para Falar";
}

function toggleStartStop() {
  if (recognizing) {
    recognition.stop();
    reset();
  } else {
    recognition.start();
    recognizing = true;
    button.innerHTML = "Clique para Parar";
  }
}
```

Figura 13 – Excerto de código das funções responsáveis pelo reconhecimento de voz da primeira abordagem

5.1.3.2 Escrita Contínua do Reconhecimento de Voz

Nesta implementação utilizaram-se os resultados intermédios do reconhecimento de voz, resultados esses que vão aparecendo na caixa de texto durante o discurso do utilizador. À medida que os mesmos são sujeitos a análise de contexto são depois atualizados com os resultados finais que irão sobrepor os anteriores. Utilizando esta abordagem é possível observar os diferentes resultados do reconhecimento de voz até se chegar ao resultado final.

```
var recognizing;
var recognition = new webkitSpeechRecognition();
recognition.continuous = true;
recognition.interim = true;
reset();
recognition.onend = reset;

recognition.onresult = function (event) {
  var final = "";
  var interim = "";
  for (var i = 0; i < event.results.length; ++i) {
    if (event.results[i].final) {
      final += event.results[i][0].transcript;
      document.getElementById('messageBox').value = final;
    } else {
      interim += event.results[i][0].transcript;
      document.getElementById('messageBox').value = interim;
    }
  }

}

function reset() {
  recognizing = false;
  button.innerHTML = "Clique para Falar";
}

function toggleStartStop() {
  if (recognizing) {
    recognition.stop();
    reset();
  } else {
    recognition.start();
    recognizing = true;
    button.innerHTML = "Clique para Parar";
  }
}
```

Figura 14 - Excerto de código das funções responsáveis pelo reconhecimento de voz da segunda abordagem

5.1.3.3 Utilização de Eventos para Despoletar Métodos

Esta abordagem é em tudo similar à anterior. No entanto testou-se a utilização do evento *Onspeechend* da API. A funcionalidade extra desta abordagem consiste então em utilizar este evento de modo a detetar a ausência de discurso e, interpretando isso como término da dicção, invocar a função responsável por enviar a mensagem ditada (não sendo necessário clicar no botão enviar).

```
recognition.onspeechend = function(){
    reset();
    sendMessage();
}
```

Figura 15 – Excerto do código responsável por detetar o evento de final de discurso na terceira abordagem

5.1.4 Arquitetura

A arquitetura da solução que compõe esta prova de conceito é bastante simples, consistindo essencialmente nos seguintes módulos:

- **Vistas –** Páginas HTML para as diferentes abordagens
- **Folha de Estilos –** Estilos CSS para as 3 páginas
- **Funcionalidades Gerais –** Código JavaScript para as funções partilhadas entre as páginas (botões *enviar*, *eliminar*, etc.).
- **Funcionalidades de Reconhecimento de Voz –** Código JavaScript das funcionalidades de reconhecimento de voz para cada abordagem
- **Serviço de Reconhecimento de Voz –** Serviço remoto de reconhecimento de voz utilizado pela Web Speech API

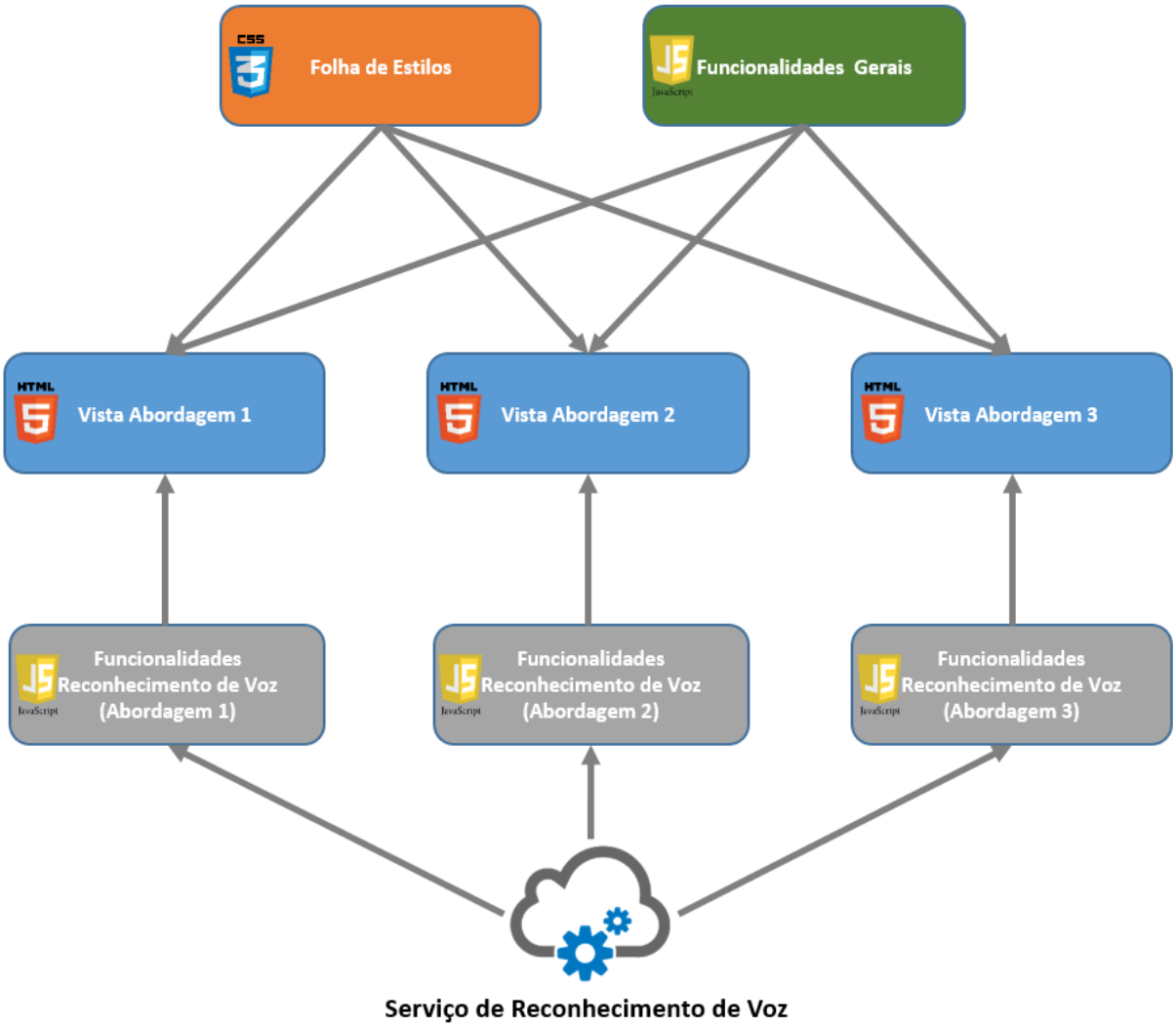


Figura 16 – Arquitetura da primeira prova de conceito

5.1.5 Resultados e Conclusões

Durante a análise desta prova de conceito, constatou-se que o reconhecimento de voz oferecido por esta API obtinha resultados bastante satisfatórios, apresentando uma elevada taxa de sucesso no reconhecimento de voz. Após apresentação e testes, a equipa decidiu continuar a apostar na investigação e na aplicação da solução de reconhecimento de voz oferecida pela Web Speech API.

Foi então solicitado ao estagiário que investigasse mais aprofundadamente os métodos e eventos desta API adaptados a um contexto de comandos ditados pelo utilizador e que os apresentasse numa nova prova de conceito.

5.2 Prova de Conceito 2: Comandos por Voz / Converter Texto para Voz

5.2.1 Objetivos

Um dos objetivos desta prova de conceito passou pela demonstração do funcionamento da Web Speech API para reconhecimento de comandos ditados por voz numa plataforma *web*, neste caso para possível aplicação no projeto do STA. Pretendia-se também demonstrar o funcionamento da síntese de voz em Português como resposta a determinados comandos, erros ou simplesmente para efetuar leitura de texto. Após alguma análise das ferramentas existentes, recorreu-se à plataforma *Voice RSS* que permite converter texto em voz *On-demand*. Para síntese ou relato por voz sobre acontecimentos mais frequentes (por exemplo erros), recorreu-se a ficheiros de som pré-gravados de modo a otimizar a utilização da plataforma *Voice RSS*, já que o acesso à sua vertente gratuita se encontra limitado a uma pequena quantidade de transcrições por dia.

5.2.2 Investigação

Constatou-se que a Web Speech API possuía também capacidade para síntese de voz, e já que a mesma tinha sido adotada para o desenvolvimento das funcionalidades de reconhecimento de voz do STA, esta seria também a escolha mais óbvia para implementar a síntese de voz. No entanto, após uma curta análise das suas funcionalidades, observou-se que não existia capacidade de síntese de voz em Português de Portugal e, sendo isto um dos requisitos principais, tiveram que ser estudadas alternativas.

Após análise de algumas alternativas, verificou-se que não existem ferramentas gratuitas para síntese de voz que suportassem a língua Portuguesa. No entanto, existe uma quantidade limitada de ferramentas pagas que satisfazem esta condição, mas apresentam quase sempre planos de preço bastante dispendiosos. Opondo-se tais ferramentas à política de adoção de *software* gratuito que foi transmitida ao estagiário, procurou-se então a solução mais económica possível: a API Voice RSS.

O Voice RSS é uma API externa cujo funcionamento consiste fundamentalmente no envio de um *link* em formato NVP (*Network Voice Protocol*) para o seu serviço de síntese de voz. No interior desse *link* especificam-se parâmetros, tais como o texto pretendido e critérios de discurso (idioma, velocidade, qualidade). Após o envio do *link*, o resultado é retornado em formato de voz. Esta API tem a particularidade da existência da síntese de voz em Português de Portugal, que não existe na Web Speech API. No entanto, cada conta em formato gratuito dispõe apenas de 350 *requests* diários, pelo que na necessidade de um maior número de *requests* terá que se optar por uma conta paga.

5.2.3 Implementação

Foi desenvolvida uma aplicação que tem como objetivo demonstrar a utilização de comandos por voz para efetuar determinadas tarefas, assim como a utilização da síntese de voz para relatar determinados acontecimentos ou textos. A aplicação utiliza a Web Speech API para as funcionalidades de reconhecimento de voz e a API Voice RSS em conjunto com ficheiros de áudio pré-gravados para a parte da síntese de voz.

A aplicação consiste numa página *web* habitual (HTML, CSS e JavaScript) onde é simulado o comportamento das funcionalidades mencionadas, de modo a demonstrar o funcionamento das duas API's em simultâneo.

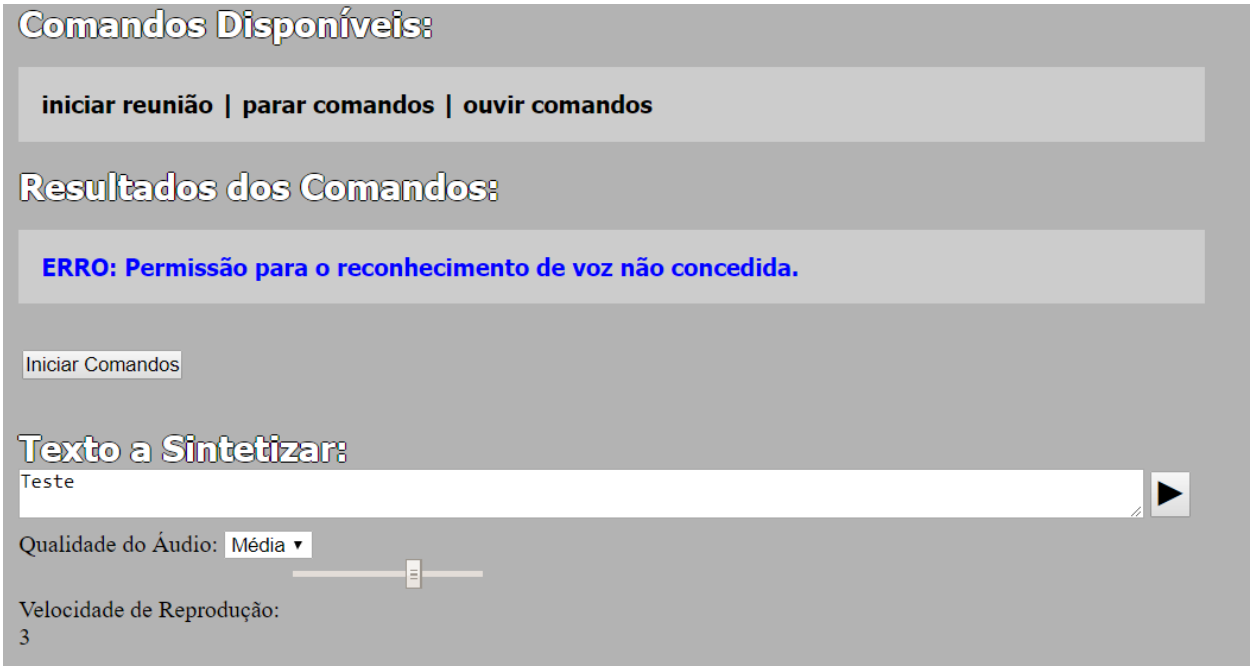


Figura 17 – Interface gráfica da segunda prova de conceito

5.2.3.1 Reconhecimento de Comandos por Voz

O reconhecimento de voz, após ser ativado e durante o decorrer da utilização é chamado em ciclo contínuo invocando funções conforme os resultados interpretados. Esta funcionalidade termina apenas quando recebe o comando para esse efeito.

Início da Interação

Na parte do reconhecimento de comandos por voz, a primeira interação terá que ser efetuada manualmente, clicando no botão "Iniciar Comandos". Os comandos disponíveis nesse momento são apresentados na figura 16, de modo a simular o facto de existirem determinados comandos que só podem ser utilizados em determinada situação. Neste momento, portanto, estão apenas disponíveis 3 comandos: "Iniciar reunião", "Parar comandos" e "Ouvir comandos", estando os dois últimos também presentes noutra qualquer situação durante a execução da aplicação.

Comando "Iniciar Reunião"

Para iniciar a reunião, o utilizador deve ditar o comando "Iniciar reunião" e, logo que o mesmo for interpretado pelo sistema, deverá conduzir o utilizador a uma nova funcionalidade onde será apresentada uma nova lista de comandos disponíveis, de modo a simular a lista de comandos possíveis depois de iniciada a reunião. Neste caso, e apenas para efeitos de demonstração, é possível enviar qualquer comando da lista a qualquer momento desde que a "reunião" já tenha sido iniciada. Numa implementação real, estes comandos terão que ser adaptados de modo a serem só possíveis de utilizar quando a situação atual assim o permitir: um comportamento semelhante à separação dos comandos disponíveis antes e depois de iniciar a reunião.

Comandos Disponíveis Durante a Reunião

A lista de comandos disponíveis durante a reunião, nesta implementação é a seguinte:

- "Abrir ata";
- "Ler ata";
- "Ler ponto";
- "Colocar para discussão";
- "Votação";
- "Votar por alguém";

- "Bloquear voto";
- "Adicionar secção";
- "Adicionar ponto";
- "Apagar ponto";
- "Adiar ponto";
- "Suspender ponto";
- "Encerrar ponto";
- "Finalizar reunião";
- "Encerrar reunião";
- "Ler anexo";
- "Declaração de voto";
- "Passar controlo";
- "Parar comandos";
- "Ouvir comandos".

Ao enviar qualquer um dos comandos anteriores, quando reconhecidos com sucesso, a aplicação deve apresentar uma mensagem do tipo “Resultados: Comando X”, de modo a simular o seu funcionamento e perceber se o comando foi realmente interpretado com sucesso. Caso não tenham sido detetadas palavras válidas, será apresentada uma mensagem de erro que alerta o utilizador para este facto.

Comando “Ouvir Comandos”

Ao utilizar o comando “Ouvir comandos”, será apresentada ao utilizador, a lista de comandos disponíveis no contexto atual. Estes comandos serão também ditados sob a forma de síntese de voz.

Comando “Encerrar Reunião”

Ao ser ditado este comando, o utilizador será redirecionado para o estado inicial da página, onde só estarão disponíveis os comandos iniciais.

Comando “Parar Comandos”

Para terminar o reconhecimento de comandos por voz, basta o utilizador ditar o comando “Parar comandos”. A partir desse momento inativam-se as funcionalidades de reconhecimento de voz.

Erros e Exceções

Durante o reconhecimento de comandos por voz, poderão existir situações que despoletem eventuais erros, tais como a inexistência de um determinado comando, inexistência de um dispositivo de captura de áudio, o cancelamento inesperado da utilização do microfone, a falta de permissão para utilização do microfone pela aplicação *web*, erro de acesso à rede, erro ao não ser detetada voz, entre outros. O utilizador é alertado sobre estes erros tanto por escrito, como por voz.

Comportamento em Caso de Falha na Interpretação de Comandos

Caso o comando não tenha sido reconhecido com sucesso (detetada confiança inferior a 70%), será iniciada uma nova função que irá verificar se foram ditas palavras válidas e, neste caso, analisar possíveis parecenças com os comandos disponíveis. No caso de existirem comandos similares ao que foi dito, estes serão sugeridos sob a forma de uma lista numerada. Nesse momento, para se escolher o comando pretendido, o utilizador terá apenas que ditar o número correspondente ao comando. Se o utilizador não disser nenhum número válido, os comandos disponíveis passam ao estado anterior.

5.2.3.2 Síntese de Voz

De modo a simular diferentes utilidades para a síntese de voz na aplicação, construíram-se três exemplos diferentes de utilização desta funcionalidade. Dois desses exemplos recorrem à API Voice RSS e consistem na síntese de texto escrito pelo utilizador numa caixa de texto e relato de acontecimentos aleatórios (erros menos comuns e sujeitos a uma maior dinâmica ou descrição da lista de comandos similares possíveis nos momentos em que o reconhecimento de voz não interpreta corretamente os comandos ditados). O terceiro exemplo é dedicado ao relato de acontecimentos mais comuns recorrendo a ficheiros de áudio locais (como por exemplo, erros comuns e descrição da lista de comandos habitual). A terceira alternativa foi pensada como um modo de minimizar a quantidade de *requests* realizados pela API, colocando a síntese de acontecimentos frequentes a recorrer a ficheiros de som locais já gravados.

Síntese de Texto Escrito pelo Utilizador

De modo a demonstrar o comportamento desta API na leitura de um determinado texto, criou-se uma *Textfield* na aplicação, onde é possível escrever algumas linhas de texto que, ao clicar no botão "Play", serão transcritas pela API e ditadas sob a forma de discurso.

De modo a testar os parâmetros da síntese de voz desta API, permite-se que o utilizador escolha a velocidade de discurso (de -10 a 10, sendo 0 a velocidade normal) através de um *slider* e a qualidade do áudio recebido (fraca, média e alta), sendo deste modo possível simular a adaptabilidade da qualidade de som à largura de banda disponível ao utilizador que está conectado. Todos estes parâmetros serão convertidos na forma de um *request* NVP e enviados ao servidor responsável pela conversão de texto em voz, que por sua vez envia um ficheiro áudio com o resultado da conversão que será reproduzido automaticamente pelo *browser*.

```
http://api.voicerss.org/?key=[Código_Key]&amp;src='[TEXTO_A_SINTETIZAR]'
&amp;hl=[IDIOMA]&f='[QUALIDADE]'&r='[VELOCIDADE]'"type="audio/mpeg"
```

Figura 18 – Exemplo de utilização de um *request* NVP

Síntese/Relato de Acontecimentos Aleatórios

Quando ocorrem acontecimentos menos comuns (erros capturados através do evento *OnError* da *Web Speech API* ou na síntese de comandos similares disponíveis quando a confiança do reconhecimento é menor que 70%), utiliza-se a API *Voice RSS* de um modo semelhante ao do comportamento da síntese do texto escrito pelo utilizador, no entanto, passando a descrição do erro como parâmetro para o link NVP. Deste modo, recorre-se a esta API apenas em casos como os descritos, minimizando o número de pedidos efetuados pela mesma.

Síntese/Relato de Acontecimentos Frequentes

Para o relato de acontecimentos fixos ou frequentes na aplicação (ouvir comandos disponíveis, erros ou descrições comuns), recorre-se à utilização de ficheiros áudio gravados previamente, que ficarão disponíveis localmente e serão utilizados de acordo com o acontecimento despoletado.

5.2.4 Arquitetura

A arquitetura da solução que compõe esta prova de conceito consiste nos seguintes módulos:

- **Vista –** Página HTML onde o utilizador pode testar e verificar as funcionalidades demonstradas
- **Folha de Estilos –** Estilos CSS para a página
- **Funcionalidades Gerais –** Código JavaScript para as funcionalidades gerais da página
- **Funcionalidades de Reconhecimento de Voz –** Código JavaScript das funcionalidades de reconhecimento de voz e interpretação de comandos
- **Serviço de Reconhecimento de Voz –** Serviço remoto de reconhecimento de voz utilizado pela Web Speech API
- **Funcionalidades de Síntese de Voz –** Código JavaScript das funcionalidades de síntese de voz (inclui as funcionalidades que decidem o tratamento de erros via *API* ou ficheiros de áudio locais)
- **Serviço de Síntese de Voz –** Serviço remoto de síntese de voz utilizado pela Voice RSS
- **Repositório de Ficheiros de Áudio –** Armazenamento de ficheiros de áudio pré-gravados com a síntese da descrição dos erros mais comuns

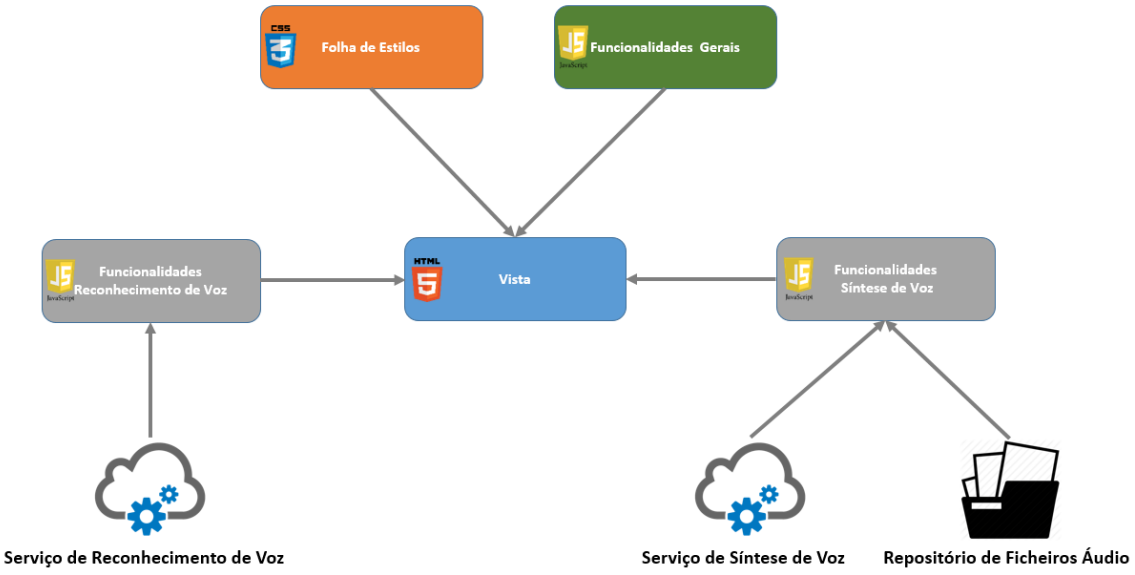


Figura 19 – Arquitetura da segunda prova de conceito

5.2.5 Resultados e Conclusões

De modo a evitar más interpretações dos comandos pelo reconhecimento de voz, depois de efetuados alguns testes, verificou-se que colocando a percentagem de confiança da interpretação em valores acima dos 70%, evitar-se-iam em larga escala essas más interpretações. Testou-se ainda com sucesso uma abordagem que, nos casos em que a confiança é menor, é ativada uma funcionalidade que apresenta comandos alternativos ao utilizador com base nas palavras transcritas. Verificou-se ainda, que a abordagem geral de interpretação de comandos por voz utilizada nesta prova de conceito funcionou com bastante sucesso, tendo sido a mesma aprovada para futura integração no STA.

O funcionamento geral da abordagem para a síntese de voz desenvolvida nesta prova de conceito correspondeu também às expetativas. A combinação entre as funcionalidades que recorriam à síntese de voz com chamadas à API e as funcionalidades que usavam os ficheiros de áudio pré-gravados revelou-se bastante vantajosa, reduzindo amplamente o número de utilizações da API, tal como era pretendido.

Embora esta abordagem tenha sido implementada com sucesso, a utilização da API *Voice RSS* nos vários ambientes de produção futuros iria acabar por exigir um elevado fluxo de pedidos de serviço, sendo deste modo necessário recorrer aos planos de pagamento da API, o que iria tornar a solução final mais dispendiosa. Foi então transmitido ao estagiário que a investigação e o desenvolvimento das funcionalidades de síntese de voz iram ficar suspensas temporariamente, enquanto não surgissem alternativas.

5.3 Desenvolvimento de Aplicação *Java* para Gravação de Voz

5.3.1 Enquadramento

Durante a implementação do STA surgiu a necessidade de desenvolver uma funcionalidade que permitisse a gravação sonora das reuniões. Apesar de esta atividade não constar no plano inicial da proposta de estágio, como naquele momento não era ainda possível efetuar a integração das funcionalidades de reconhecimento de voz na solução final e as atividades para as funcionalidades de síntese de voz estavam suspensas, foi solicitado ao estagiário que desenvolvesse uma solução que tornasse possível esta abordagem.

5.3.2 Investigação

Ao testar as possíveis abordagens para a implementação desta aplicação, verificou-se que iriam ser necessários dois sistemas distintos para captura de voz. Isto porque não seria possível a utilização do mesmo microfone por parte do controlador da reunião para as atividades de reconhecimento de comandos por voz em conjunção com a voz dos restantes participantes. Ao ser utilizado o mesmo microfone para estas duas tarefas, o discurso dos participantes que se encontram a uma maior distância do microfone torna-se praticamente impercetível e, ao ser utilizada a funcionalidade de reconhecimento de comandos por voz, a mesma seria afetada também pelo discurso dos outros participantes, influenciando negativamente a sua taxa de acerto.

Deste modo, uma possível solução para este problema passaria pela utilização de dois microfones, um deles direcionado para o controlador da reunião efetuando a tarefa de reconhecimento de comandos por voz, e outro dirigido aos restantes intervenientes para a gravação do áudio da reunião (a gravação de áudio poderia recorrer, por exemplo, ao sistema de microfones que geralmente é utilizado nas assembleias, enquanto que o reconhecimento de voz poderia ter um microfone dedicado).

Posto isto, depois de efetuados alguns testes e pesquisas, verificou-se que atualmente os *browsers* não suportam a utilização de dois microfones diferentes em simultâneo. Pelo facto, a solução para a gravação da reunião teria que passar por uma funcionalidade que, de certo modo corresse de forma independente do *browser*. Pensou-se então em desenvolver a aplicação em *Java*, integrando-a depois na janela do STA. No entanto, a maioria dos *browsers* já deixou de suportar aplicações Java integradas, pelo que esta não seria de todo a melhor abordagem.

Entretanto, após mais alguma investigação, o estagiário sugeriu à equipa, a utilização do JNLP (ou *JNLP* ou *Java Network Launch Protocol*), o qual permite que uma aplicação seja lançada no computador cliente, utilizando os recursos hospedados num servidor. O modo de funcionamento do JNLP consiste no descarregamento de um pequeno ficheiro que, depois de inicializado, irá lançar a aplicação pretendida no computador cliente automaticamente. Este método acabaria por resolver, de certo modo, os problemas descritos anteriormente, já que permite à aplicação correr independentemente do *browser*.

Considerando esta sugestão, foi proposto ao estagiário desenvolver uma primeira versão, do que viria a ser a funcionalidade para gravação sonora das reuniões a ser posteriormente integrada no STA.

5.3.3 Implementação

5.3.3.1 User Stories

Como se tratava de uma tarefa de desenvolvimento já com tarefas previamente delineadas, foi sugerido ao estagiário que utilizasse *User Stories* para definir as várias funcionalidades da aplicação. Assim sendo, o estagiário preparou a seguinte lista de *User Stories* onde se expressam os diversos itens de trabalho a realizar.

Lançamento da Aplicação Via JNLP

Como Secretário, pretendo que a aplicação seja lançada através do *browser*, via JNLP de modo a ser compatível com todos os *browsers*.

Descrição:

- Inserir especificações em “Project Properties”
- Habilitar o “Web Start”
- Especificar servidor
- Especificar “key” – gerar a partir da licença
- Habilitar “Software Protections”

Layout

Como Secretário, pretendo visualizar um *layout* simples e em conformidade com o STA, de modo a que as funcionalidades sejam evidentes e a aplicação seja entendida como parte integrante do STA.

Descrição:

- Criar ícones
- Criar elementos do *layout*
- Criar Botões
- Criar caixa de *logs*
- Criar espaços para logótipo, informações e *timer*

Obtenção da Diretoria Predefinida para os Ficheiros da Gravação

Como Secretário, pretendo que exista uma diretoria predefinida para guardar os ficheiros gerados pela gravação, de modo a agilizar a utilização da aplicação.

Escolha da Diretoria para os Ficheiros de Gravação

Como Secretário, pretendo poder escolher uma diretoria diferente da diretoria predefinida, para guardar os ficheiros gerados pela gravação, de modo a personalizar a localização dos mesmos.

Descrição:

- Criar botão para o efeito
- Criar uma chamada a um "JFileChooser" em modo diretorias
- Gravar caminho na classe correspondente
- Enviar informações para o *log*

Escolha do Microfone

Como Secretário, pretendo poder escolher o microfone a ser utilizado para a gravação da reunião, de modo a poder selecionar um microfone diferente do utilizado para o reconhecimento de voz.

Descrição:

- Importar biblioteca JAVE
- Criar *layout* para o menu de seleção
- Criar funcionalidade para obter todos os microfones do sistema
- Criar funcionalidade que permita a escolha do microfone
- Gravar escolha na classe correspondente
- Enviar informações para o *log* no final da alteração

Aviso Caso Existam Menos de Dois Microfones

Como Secretário, pretendo ser alertado caso sejam detetados menos de dois microfones no sistema de modo a que seja possível gravar a reunião e utilizar as funcionalidades de reconhecimento de voz.

Descrição:

- Importar biblioteca JAVE
- Criar funcionalidade para detetar a presença de menos de dois microfones
- Despoletar aviso através de uma "MessageDialog"

Saída da Aplicação

Como Secretário, pretendo ter uma opção para poder sair da aplicação, de modo a libertar o ecrã para outras tarefas.

Descrição:

- Criar funcionalidade para sair da aplicação

Confirmação de Saída da Aplicação

Como Secretário, pretendo que me seja apresentada uma mensagem de confirmação sempre que tentar sair da aplicação de modo a evitar saídas involuntárias.

Descrição:

- Criar funcionalidade que deteta a saída da aplicação
- Despoletar aviso através de um "Confirm Dialog"

Gravação da Reunião

Como Secretário, pretendo ter uma opção para gravar a reunião, de modo a que consiga obter um registo sonoro da mesma.

Descrição:

- Importar biblioteca JAVE
- Obter dados de formato áudio e microfone a ser utilizado
- Obter caminho para gravação dos ficheiros
- Criar funcionalidade de gravação áudio
- Enviar informações para o *log* no início da gravação

Pausa na Gravação

Como Secretário, pretendo ter uma opção que permita colocar a gravação da reunião em pausa, de modo a poder dar continuidade à gravação, mais tarde.

Descrição:

- Importar biblioteca JAVE
- Criar função que pare e converta automaticamente a última gravação para *mp3*
- Criar funcionalidade que mude os ícones do botão para gravar/pausar conforme a sua utilização
- Enviar informações para o *log* sobre a pausa na gravação

Parar Gravação

Como Secretário, pretendo ter uma opção que permita parar a gravação da reunião, de modo a que possa aceder ao registo sonoro gerado.

Descrição:

- Importar biblioteca JAVE
- Criar função que pare e converta automaticamente a última gravação para *mp3*
- Criar funcionalidade que mude os ícones do botão para gravar/pausar conforme a sua utilização
- Enviar informações para o *log* no final da gravação

Bloqueio de Funcionalidades Durante a Gravação

Como Secretário, pretendo que todas as funcionalidades da aplicação, exceto a Pausa e a Paragem da Gravação, estejam bloqueadas durante a gravação, de modo a evitar a sua utilização incorreta.

Descrição:

- Funcionalidade que bloqueia todos os botões durante a gravação, exceto o botão parar e o botão pausar.

Criação de Ficheiros de Áudio a cada Minuto

Como Secretário, pretendo que sejam criados ficheiros de áudio a cada minuto, de modo a agilizar o processo de conversão em *mp3*.

Descrição:

- Definir tempo de interrupção para um minuto
- Criar um novo *timer*
- Chamar a função para gravação áudio
- Chamar a função de conversão para *mp3*
- Enviar informações para o *log* no final da gravação

Conversão dos Ficheiro de Áudio para mp3

Como Secretário, pretendo que os ficheiros WAV gerados pela aplicação sejam automaticamente convertidos em *mp3*, de modo a otimizar o espaço dos ficheiros em disco.

Descrição:

- Importar biblioteca JAVE
- Importar caminho da pasta da gravação
- Definir "Codecs de áudio, Bitrate, Canais e Sampling rate"
- Criar funcionalidade para conversão dos ficheiros WAV para *mp3*
- Adicionar o nome do ficheiro a um "Array" de "Strings" de modo a ser utilizado na playlist
- Enviar informações para o *log* no final da conversão

Nome dos Ficheiros

Como Secretário, pretendo que o nome dos ficheiros *mp3* gerados sigam uma determinada ordem, de modo a que seja possível ordená-los cronologicamente.

Descrição:

- Criar funcionalidade para atribuição de nomes únicos, ordenados cronologicamente, aos ficheiros áudio

Criação de Playlists

Como Secretário, pretendo que no final da gravação seja criada uma *playlist* com todos os ficheiros gerados, de modo a poder centralizar todos os ficheiros.

Descrição:

- Importar a localização dos ficheiros de som
- Criar um novo ficheiro na localização indicada com a extensão "m3u"
- Obter nomes dos ficheiros áudio necessários
- Criar funcionalidade que escreve no ficheiro cada faixa a ele pertencente
- Enviar informações para o *log* no final

Visualização de Ocorrências

Como Secretário, pretendo ser informado das ocorrências durante a execução da aplicação através de um log, de modo a ter uma visão do estado da execução.

Descrição:

- Criar um "Text Field" em branco
- Aplicar a propriedade não editável ao "Textfield"

Visualização do Tempo de Gravação Decorrido

Como Secretário, pretendo ser informado do tempo de gravação decorrido, de modo a ter noção da dimensão do áudio que vai sendo guardado.

Descrição:

- Criar funcionalidade para obter o tempo atual
- Criar funcionalidade que soma os tempos ao longo da gravação
- Criar funcionalidade que converte os tempos em "bruto" para uma estrutura do tipo "HH:MM:SS"
- Criar funcionalidade que apresenta o tempo decorrido no layout da aplicação

5.3.3.2 Mockups

Como forma de demonstrar e simular o funcionamento do *layout* da aplicação a desenvolver, elaboraram-se *mockups* que retratavam as diversas fases e/ou ecrãs disponíveis na aplicação. Na imagem seguinte consta, como exemplo, um dos *mockups* construídos.

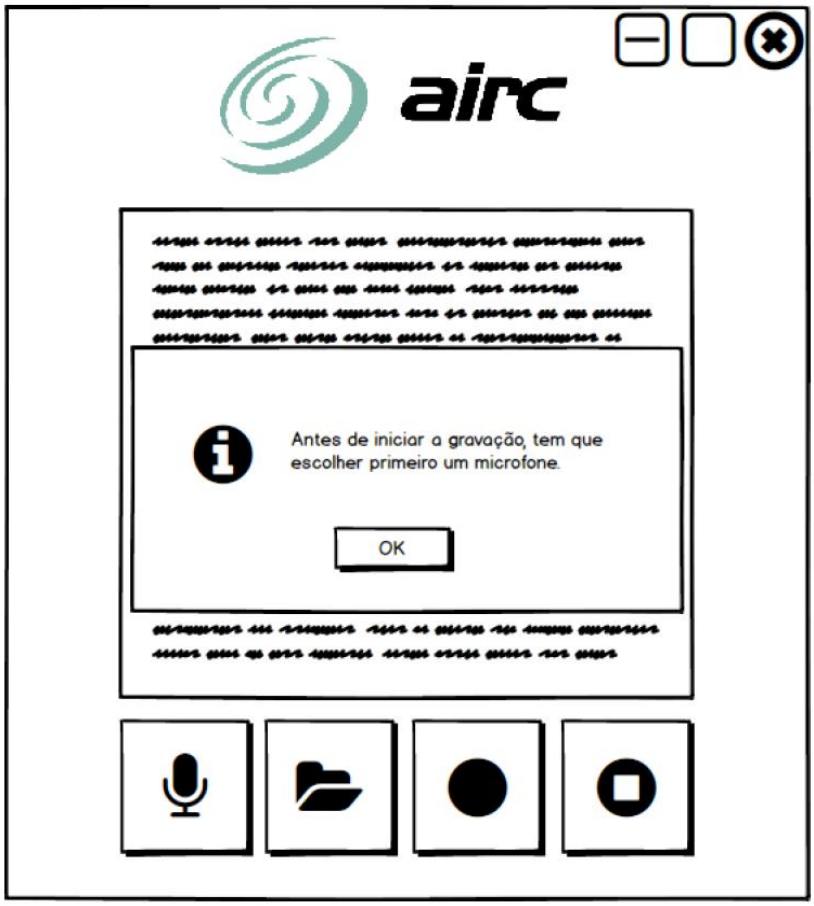


Figura 20 – *Mockup* da aplicação desenvolvida com mensagem informativa

5.3.3.3 Modo de Funcionamento

Depois de lançada a aplicação, esta irá verificar se o sistema possui mais que um microfone. Caso não sejam detetados microfones ou se existir apenas um, será apresentada uma mensagem de alerta informando o utilizador sobre este facto. Depois disto, de modo a proceder á gravação, o utilizador terá que escolher a localização onde pretende que os ficheiros de som sejam gravados e o microfone a ser utilizado. Caso estes requisitos não tenham sido cumpridos, ao carregar no botão destinado á gravação, será apresentada uma mensagem que alerta o utilizador para este facto, impedindo também o início da gravação.

Depois de escolhidos a localização pretendida para os ficheiros de som e o microfone destinado á gravação da reunião, ao ser pressionado o botão de gravação procede-se então à gravação de som. No decorrer da gravação vão sendo automaticamente criados ficheiros *WAV* com a duração de um minuto cada, para que seja possível ao sistema converter os ficheiros *WAV* já gravados, em *mp3*, durante a gravação, evitando que essa tarefa (normalmente morosa) seja realizada apenas no final da mesma.

Os ficheiros *WAV* necessitam de muito espaço em disco o que, para a gravação completa de uma reunião, exigiria bastante do sistema. A solução passaria então por utilizar ficheiros de áudio comprimidos, por exemplo, ficheiros *mp3*. Deste modo, tendo em conta que a linguagem *Java* não suporta nativamente esses ficheiros, optou-se pela utilização de uma técnica de conversão recorrendo a uma biblioteca externa capaz desta tarefa: a JAVE (*Java Audio Video Encoder*). Assim, a cada minuto de gravação é chamada uma função desta biblioteca que converte o último ficheiro gravado para *mp3*. Estes ficheiros vão sendo criados e organizados alfabeticamente (o nome é composto pela data e hora atual de modo a que seja único e possível de ordenar cronologicamente).

```
//função para converter os ficheiros wav para mp3
private static void toMp3(String pathWav){
    String saveMessages = UIRecorder.textLog.getText();
    UIRecorder.textLog.setText(saveMessages+"\nA converter "+pathWav+" para mp3.");
    String pathMp3 = pathWav.replace(".wav", ".mp3"); //obtem o mesmo caminho que o ficheiro wav e altera a extensao para mp3
    FileHandling.mp3Files.add(pathMp3); //adiciona o caminho à classe de modo a que fique disponível para ser recolhido na playlist
    String targetFile = pathMp3;
    String sourceFile = pathWav;

    //atribuir codecs e qualidade de conversão
    int samplingRate = 8000;// pode ser 8000, 16000 mono ou 16000 stereo - em 8000 reduz-se muito o tempo de conversão e concatenação
    int channels = 2;// 1 para mono / 2 para stereo
    int bitRate = 190000;// pode ser 128, 160, 190 kbps, etc..

    AudioAttributes audio = new AudioAttributes();
    audio.setCodec("libmp3lame"); //atribuir codec da biblioteca
    audio.setBitRate(bitRate);
    audio.setChannels(channels);
    audio.setSamplingRate(samplingRate);
    EncodingAttributes ea = new EncodingAttributes();
    ea.setAudioAttributes(audio);
    ea.setFormat("mp3"); //atribuir formato de conversão
    File f = new File(sourceFile);
    Encoder e = new Encoder();


    try {
        e.encode(f, new File(targetFile), ea); //conversão
    } catch (IllegalArgumentException ex) {
        Logger.getLogger(Recorder.class.getName()).log(Level.SEVERE, null, ex);
    } catch (EncoderException ex) {
        Logger.getLogger(Recorder.class.getName()).log(Level.SEVERE, null, ex);
    }

    saveMessages = UIRecorder.textLog.getText();
    UIRecorder.textLog.setText(saveMessages+"\n"+pathWav+" convertido para mp3.");
}
```

Figura 21 – Código da função responsável pela conversão dos ficheiros *WAV* em *mp3*

No final da gravação é apresentada uma mensagem ao utilizador que o questiona se pretende concatenar todos os ficheiros de áudio num só ficheiro ou criar uma playlist com todo o áudio da reunião. As razões que levaram a esta implementação prendem-se com o facto de a concatenação de um largo número de ficheiros ser bastante morosa, já que esta tarefa requer a utilização de muitos recursos da máquina. Assim, o recurso à *playlist* seria uma alternativa mais leve e imediata.

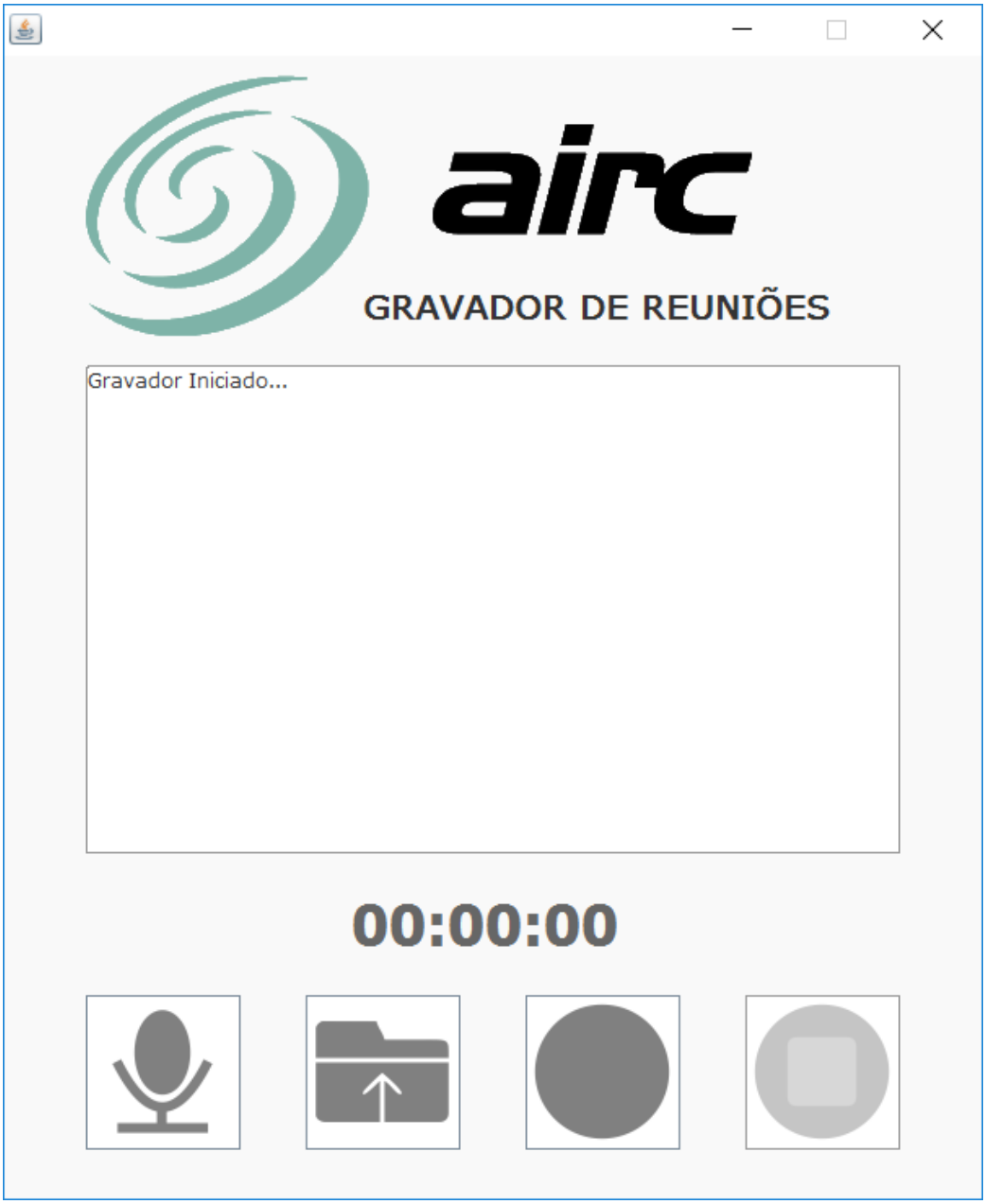


Figura 22 – Janela principal da aplicação *JAVA* para gravação sonora das reuniões

5.3.3.4 Arquitetura

A arquitetura da aplicação *Java* desenvolvida para gravação sonora das reuniões consiste nos seguintes módulos:

- **Vista Página Web –** Página *web* que simula o ambiente do STA (a aplicação Java será invocada via JNLP a partir daqui)
- **UI Java –** Interface gráfica da aplicação "Gravador"
- **Gestão de Ficheiros –** Código Java para as funcionalidades que manipulam os ficheiros
- **Gestão do Mixer –** Código Java para as funcionalidades de gestão de microfones
- **Funcionalidades para Gravação –** Código Java para as funcionalidades responsáveis pela gravação do áudio
- **Gestão do Tempo de Gravação –** Código Java para as funcionalidades que gerem os tempos de gravação
- **Biblioteca Jave –** Biblioteca responsável pelas funções que permitem a conversão dos ficheiros *WAV* para *mp3*

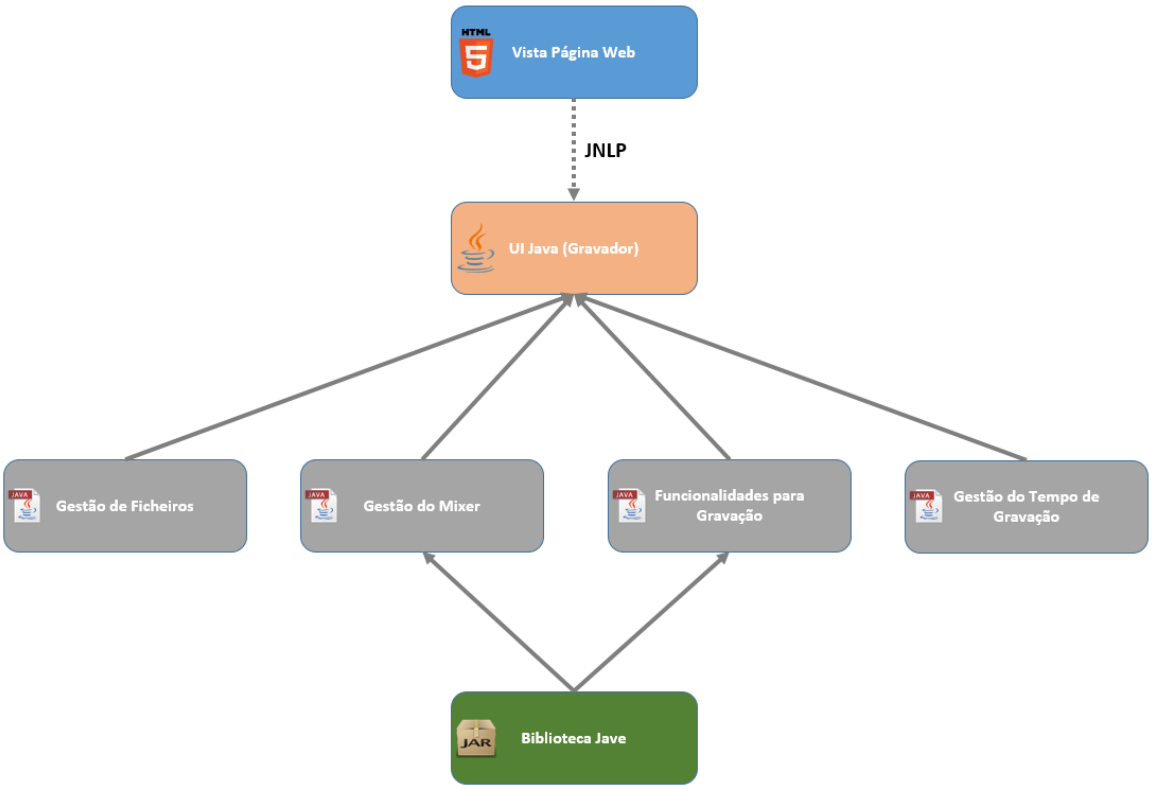


Figura 23 – Arquitetura da aplicação para gravação sonora das reuniões

5.3.4 Resultados e Conclusões

Durante o desenvolvimento da aplicação, o estagiário constatou alguns factos que o levaram à adoção de algumas abordagens, das quais se destacam as seguintes:

- **Conversão *WAV* para *mp3*** – tal como já foi descrito anteriormente, a linguagem Java conta apenas com métodos predefinidos de gravação áudio em formato *WAV*. Dado que este tipo de ficheiros ocupa grandes quantidades de espaço em disco, o arquivo dos mesmos iria acabar por exigir uma grande capacidade de armazenamento. A solução para esta questão passa então por compactar estes ficheiros em *mp3*, exigindo assim uma menor capacidade de armazenamento.

- **Gravação áudio separada em ficheiros de um minuto cada** – optou-se por esta abordagem devido à duração da tarefa de conversão *WAV* em *mp3*. Caso se procedesse à conversão total de um ficheiro *WAV* gerado durante uma reunião (que pode contar com várias horas de gravação), no final da mesma, a tarefa de conversão poderia prolongar-se por bastante tempo. Deste modo, optou-se por um ciclo de gravação de pequenos ficheiros de um minuto, que vão sendo convertidos para *mp3* à medida que são gravados. Deste modo, garante-se que no final da gravação, os ficheiros gerados já se encontram totalmente convertidos em *mp3*.

- **Seleção de microfone –** este método permite ao utilizador escolher o microfone que pretende utilizar para realizar a gravação da reunião. Deste modo, torna-se possível escolher uma fonte diferente da que é usada pelo sistema de reconhecimento de voz ou escolher a fonte mais adequada para a gravação pretendida.

- **Seleção entre *playlist* e ficheiro concatenado –** A tarefa de concatenação dos ficheiros já convertidos em *mp3*, acaba por exigir algum tempo de processamento que, dependendo da quantidade e duração total dos ficheiros a concatenar, pode-se tornar longo. Deste modo, o estagiário decidiu proporcionar ao utilizador da aplicação, a possibilidade de escolher entre concatenar os ficheiros gerados esperando pelo final do processamento, ou gerar instantaneamente uma *playlist* ordenada cronologicamente que permite ouvir os ficheiros de modo contínuo, como se de um único ficheiro se tratasse.

No final do desenvolvimento, o estagiário em conjunto com a restante equipa, reuniu com o *Product Owner* com o intuito de analisar a aplicação produzida. A mesma foi testada pelos intervenientes e a maioria das abordagens foi aprovada. No entanto, o facto de esta aplicação ter que executar, de certo modo, descentralizada do sistema principal, era algo que não estava de acordo com alguns dos princípios e objetivos fundamentais que levaram ao desenvolvimento do STA. Neste caso, a simplicidade e a conveniência de reunir as várias funcionalidades necessárias num só sistema. Entretanto, após algum debate, um dos elementos da equipa sugeriu a utilização de *websockets* para capacitar a comunicação entre o *browser* e a aplicação Java, de modo a suprimir a necessidade de recorrer a uma UI *Java* para a interação do utilizador com as funcionalidades desenvolvidas. Esta ideia foi bem recebida por parte do P*roduct Owner* que, por

sua vez, solicitou ao estagiário o desenvolvimento de uma pequena prova de conceito que permitisse testar a abordagem sugerida com as funcionalidades já desenvolvidas.

5.4 Prova de Conceito 3: Comunicação entre Aplicação *Java* e Aplicação *Web* via *WebSockets*

5.4.1 Enquadramento e Objetivos

Tal como descrito na secção anterior, o grande objetivo desta prova de conceito passou por demonstrar que as funcionalidades desenvolvidas para a gravação de voz, funcionariam igualmente através da utilização de *websockets*. Como já existiam alguns desenvolvimentos na AIRC que consistiam nesta abordagem, foi fornecida ao estagiário uma implementação-exemplo em que o mesmo se pudesse basear.

Este novo desenvolvimento teria que contar então com as funcionalidades de gravação, pausa e seleção de microfone já desenvolvidas, no entanto as mesmas teriam que ser invocadas a partir de uma página *web* (tal como se do ambiente do STA se tratasse). Para além destes requisitos, as funcionalidades de conversão *WAV* em *mp3* e geração de *playlists* seriam também para manter (o *Product Owner* optou por anular a funcionalidade para concatenação dos ficheiros, utilizando apenas a abordagem de geração de *playlists*). O estagiário foi também instruído para desenvolver uma funcionalidade que, no final da gravação, colocasse todos os ficheiros *mp3* gerados conjuntamente com a sua *playlist*, numa pasta zipada.

5.4.2 Implementação

5.4.2.1 User Stories

À semelhança da implementação anterior, o estagiário elaborou uma lista de *User Stories* que traduziriam o trabalho a realizar durante a fase de desenvolvimento que se seguia. Seguidamente é apresentada a lista mencionada (as funcionalidades que constam na implementação anterior e não fazem parte desta lista de *User Stories*, não necessitaram de ser alteradas ou adaptadas).

Ambiente de interação Web

Como Secretário, pretendo ter disponíveis opções para escolher um microfone, gravar, pausar e parar a gravação, de modo a que me seja possível controlar o registo sonoro da reunião.

Descrição:

- Criar ambiente gráfico

- Criar botões “Gravar”, “Pausar”, “Parar” e “Escolher Microfone”
- Criar caixa de *logs*

Comunicação entre a Interface *Web* e a Aplicação Java

Como Secretário, pretendo aceder às funcionalidades de seleção de microfone, gravar, pausar e parar a reunião a partir da interface *web*, de modo a que o sistema permaneça centralizado.

Descrição:

- Importar biblioteca GSON
- Adaptar funcionalidades Javascript para utilizar *websockets*
- Adaptar funcionalidades Java para utilizar *websockets*
- Criar funcionalidade para receção, envio e interpretação de mensagens JSON do lado do servidor
- Criar redireccionamento dos comandos provenientes nas mensagens JSON interpretadas no lado do servidor para os métodos correspondentes
- Criar funcionalidade para gerar e interpretar mensagens JSON de comunicação entre a interface *web* e o servidor local
- Criar funcionalidade para interpretar mensagens de estado do servidor e coloca-las na caixa de *logs*

Seleção do Microfone

Como Secretário, pretendo poder escolher o microfone a ser utilizado para a gravação da reunião, de modo a poder selecionar um microfone diferente do utilizado para o reconhecimento de voz.

Descrição:

- Importar biblioteca JAVE
- Criar funcionalidade para apresentar a lista de microfones na interface *web* a partir da mensagem JSON proveniente do servidor
- Criar método para enviar o microfone escolhido para o servidor

Gravação da Reunião

Como Secretário, pretendo ter uma opção para gravar a reunião, de modo a que consiga obter um registo sonoro da mesma.

Descrição:

- Importar biblioteca JAVE
- Criar método para enviar a instrução JSON de inicio de gravação via *websockets* para o servidor
- Adaptar método existente

Pausa na Gravação

Como Secretário, pretendo ter uma opção que permita colocar a gravação da reunião em pausa, de modo a poder dar continuidade à gravação, mais tarde.

Descrição:

- Importar biblioteca JAVE
- Criar método para enviar a instrução JSON de pausa na gravação via *websockets* para o servidor
- Adaptar método existente

Parar Gravação

Como Secretário, pretendo ter uma opção que permita parar a gravação da reunião, de modo a que possa aceder ao registo sonoro gerado.

Descrição:

- Importar biblioteca JAVE
- Criar método para enviar a instrução JSON de parar gravação via *websockets* para o servidor
- Adaptar método existente

Visualização de Ocorrências

Como Secretário, pretendo ser informado das ocorrências durante a execução da aplicação através de um *log*, de modo a ter uma visão do estado da execução.

Descrição:

- Criar método para interpretar mensagens de estado provenientes do servidor e apresentá-las na caixa de texto

Zipar Ficheiros no Final da Gravação

Como Secretário, pretendo que os ficheiros onde consta o áudio da reunião sejam agrupados numa pasta zipada em conjunto com a sua *playlist*, de modo a simplificar a sua manipulação.

Descrição:

- Criar método para zipar os ficheiros gerados
- Criar método para apagar os ficheiros já zipados

5.4.2.2 Modo de funcionamento

Ao abrir a página de interação com o utilizador é executado um *script* que efetua a ligação ao servidor local. Depois de existir comunicação entre a aplicação *web* e a aplicação em servidor, ambas as plataformas ficam a aguardar pedidos via *websockets*. Os pedidos são enviados em formato de objetos JSON, que são depois interpretados e encaminhados de modo a executar a funcionalidade correspondente.

Por exemplo, o utilizador pretende iniciar a gravação de voz e clica no botão destinado ao efeito. No momento em que o utilizador concretiza esta ação, é criado um objeto *JSON* com um parâmetro correspondente ao comando que é enviado para o servidor. Ao receber o objeto, o servidor descodifica-o e interpreta o comando enviado, invocando de seguida o método que inicia a gravação de voz. A tabela seguinte ilustra alguns exemplos de envio e interpretação de pedidos tanto do lado da aplicação *web* como do lado do servidor.

Tabela 4 – Envio e interpretação de pedidos entre a aplicação *web* e o servidor

	Envio de Pedido	Interpretação de Pedido

Aplicação Web	`startRecord: function() {`
	`  var request = JSON.stringify({`
	`  type: "STARTRECORDING"`
	`  });`
	`  this._connection.send(request);`
	`},`
		`this._connection.onmessage = function (e) {`
		`  // parse the message into a javascript object`
		`  var message = JSON.parse(e.data);`
		`  if (message.type === "MICLIST"){`
		`  //...`
		`  }`
		`}`

Aplicação Servidor	`Response response = new Response();`
	`...`
	`response.setRecordDevices(recordDevices);`
	`response.setType("MICLIST");`
	`conn.send(new Gson().toJson(response));`
		`if (frame instanceof TextWebSocketFrame) {`
		`  Request request = new Gson()`
		`  .fromJson(((TextWebSocketFrame) frame)`
		`  .text(), Request.class);`
		
		`  Response response = new Response();`
		`  response.setType(request.getType());`
		
		`  if ("STARTRECORDING".`
		`  equalsIgnoreCase(request.getType())) {`
		`  try {`
		`  Recorder.startRecording();`
		`  response.setSuccess(true);`
		`  ...`

O funcionamento da estrutura do lado do servidor é essencialmente similar ao funcionamento da aplicação de gravação de voz previamente desenvolvida. Excluiu-se apenas a funcionalidade para a concatenação dos ficheiros *mp3* gerados e acrescentou-se uma funcionalidade que, no final da gravação agrupa todos os ficheiros *mp3* e a sua *playlist* num ficheiro zipado.

5.4.2.3 Arquitetura

A arquitetura da prova de conceito realizada para testar esta abordagem consiste nos módulos seguintes:

- **Vista Página Web –** Página *web* que simula o ambiente do STA
- **Comunicação e Interpretação de Comandos –** Código Javascript responsável por enviar os comandos em formato JSON para o servidor via *websockets*, assim como interpretar as mensagens provenientes do mesmo
- **Servidor Virtual Local –** Servidor virtual local responsável pela comunicação com a aplicação cliente e pelas funcionalidades de gravação de voz
- **Interpretação e Redireccionamento dos Pedidos –** Funcionalidades responsáveis pela interpretação dos comandos JSON recebidos e invocação dos métodos correspondentes
- **Gestão de Ficheiros –** Código Java para as funcionalidades que manipulam os ficheiros
- **Biblioteca *Gson* –** Biblioteca responsável pelas funções que permitem interpretar as mensagens em formato JSON para um formato suportado pelo *Java*
- **Gestão do *Mixer* –** Código Java para as funcionalidades de gestão de microfones
- **Funcionalidades para Gravação –** Código Java para as funcionalidades responsáveis pela gravação do áudio
- **Gestão do Tempo de Gravação –** Código Java para as funcionalidades que gerem os tempos de gravação
- **Biblioteca *Jave* –** Biblioteca responsável pelas funções que permitem a conversão dos ficheiros *WAV* para *mp3*

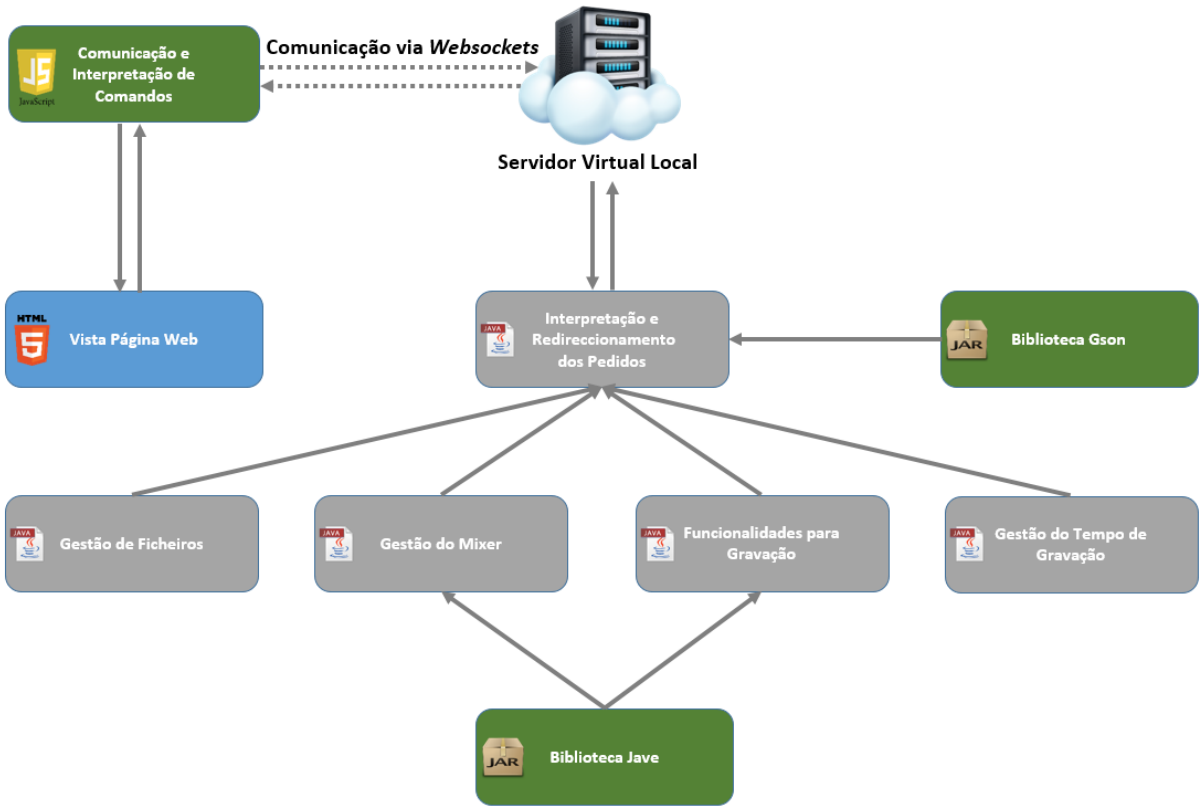


Figura 24 – Arquitetura da Terceira Prova de Conceito

5.4.3 Resultados e Conclusões

Com a realização desta prova de conceito, comprovou-se a possibilidade de atingir o objetivo proposto: era possível integrar as funcionalidades para gravação da reunião já desenvolvidas no STA sem existir necessidade de recorrer a uma interface de utilizador separada do sistema principal. A comunicação entre as duas aplicações realizava-se de forma instantânea e os pedidos efetuados por ambas as instâncias eram satisfeitos com sucesso.

As funcionalidades do sistema de gravação de voz previamente desenvolvidas não careceram de alterações profundas, tendo sido necessário apenas a sua adaptação, de modo a que fosse possível invocá-las a partir do sistema de redireccionamento de pedidos implementado.

Mais uma vez, foi realizada uma reunião onde se demonstrou o funcionamento da prova de conceito à equipa e ao *Product Owner* que, por sua vez, concordou com os desenvolvimentos efetuados e deu instruções para a integração desta abordagem no STA.

5.5 Integração no STA

5.5.1 Enquadramento

Após as demonstrações e provas de conceito realizadas e aprovadas durante as fases anteriores, chegou-se ao momento de integrar as funcionalidades que foram sendo desenvolvidas. Esta etapa teve como principais objetivos, dotar o STA com capacidades de reconhecimento de comandos por voz e gravação sonora das reuniões. Como a maioria das funcionalidades tinha já sido implementada durante as provas de conceito apresentadas nas etapas anteriores, e como estas se aproximavam bastante da estrutura utilizada no STA, a tarefa de integração acabou por se tornar bastante simples, sendo que o maior desafio passou por adaptar as funcionalidades de reconhecimento de voz escritas em Javascript para TypeScript.

5.5.2 Implementação

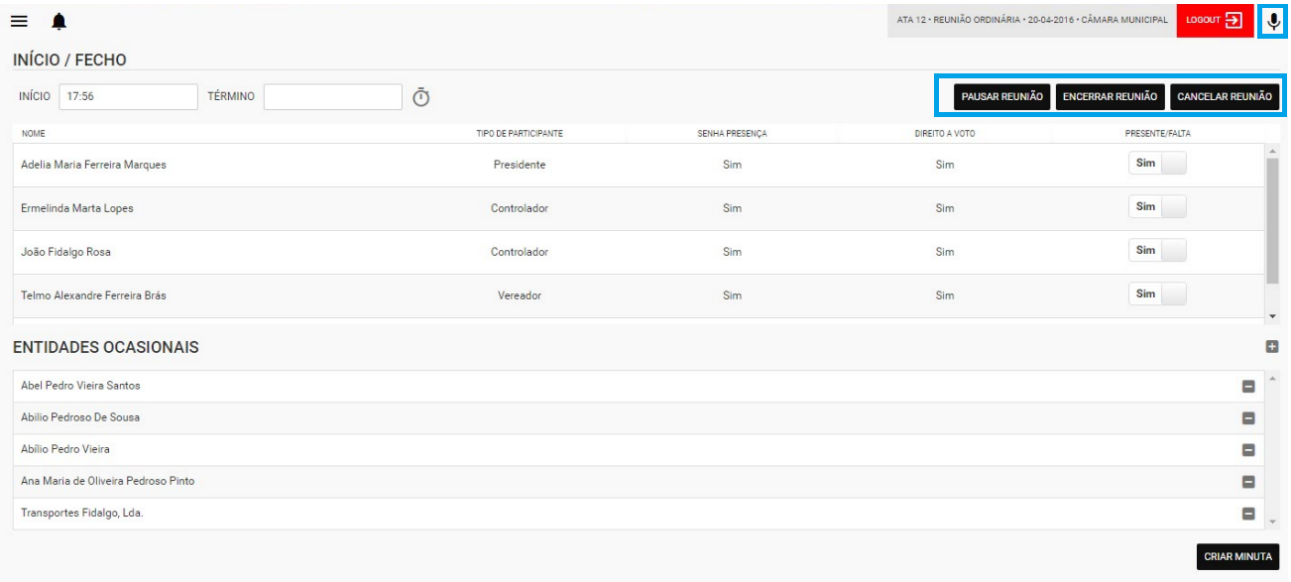
Durante a fase de integração, foi preparada uma máquina virtual no computador do estagiário, através da plataforma Vagrant, tendo-lhe sido concedido acesso ao ambiente de desenvolvimento da aplicação e ao seu repositório, de modo a poder integrar os módulos necessários. À medida que os desenvolvimentos iam sendo realizados, o estagiário ia submetendo as novas funcionalidades ao repositório, as quais eram depois testadas por elementos fora da equipa de desenvolvimento e posteriormente aprovadas pelo *Scrum Master*.

Esta fase durou cerca de um mês e subdividiu-se em duas *sprints* de duas semanas cada: a primeira *sprint* consistiu na integração do módulo de reconhecimento de voz no STA, enquanto que a segunda *sprint* teve como objetivo a integração do módulo de gravação sonora para reuniões. O trabalho realizado consistiu, portanto, em repetir as *User Stories* já implementadas nos anteriores desenvolvimentos e provas de conceito, mas desta vez no contexto da integração no sistema principal.

5.5.3 Modo de Funcionamento dos Módulos Integrados

5.5.3.1 Módulo de Reconhecimento de Comandos por Voz

À semelhança do comportamento da solução apresentada na segunda prova de conceito, para ter acesso às funcionalidades de reconhecimento de voz, o utilizador tem primeiro que clicar no botão que as ativa. A partir desse momento, o utilizador fica habilitado a utilizar comandos por voz de modo a ativar as funcionalidades principais disponíveis no contexto do local da aplicação onde se encontra.



(Em destaque, o botão para ativação/desativação do reconhecimento de voz e opções disponíveis no contexto)

Figura 25 – Ecrã e opções disponíveis durante a reunião

A tabela seguinte demonstra os comandos disponíveis por cada contexto.

Tabela 5 – Lista de comandos disponíveis por contexto

Contexto	Comandos Disponíveis
Página Inicial	- Ouvir Comandos - Entrar Reunião
Contexto da Reunião	- Iniciar Reunião - Pausar Reunião - Retomar Reunião - Encerrar Reunião - Cancelar Reunião - Abrir Votação - Abrir Ata - Abrir Minuta - Passar Controlo - Ouvir Comandos
Contexto de Controlo de Pontos	- Adicionar Ponto - Apagar Ponto - Discutir Ponto - Suspender Ponto - Adiar Ponto - Ouvir Comandos

5.5.3.2 Módulo para Gravação Sonora das Reuniões

As funcionalidades inerentes à gravação sonora das reuniões no STA, são ativadas automaticamente aquando da interação com os botões de controlo de reunião. Ou seja, a gravação é iniciada quando o utilizador clica no botão de início de reunião, é colocada em pausa/retomada quando o utilizador clica nos botões de Pausar/Retomar Reunião, termina e efetua o agrupamento dos ficheiros em pasta zipada quando o utilizador clica no botão para terminar a reunião e é terminada e eliminada quando o utilizador clica no botão para cancelar a reunião. Excetua-se apenas o botão que permite escolher o microfone a utilizar pelo sistema de gravação áudio que se destina apenas para este efeito.

As funcionalidades disponíveis na prova de conceito que antecedeu este desenvolvimento, como por exemplo a conversão contínua do áudio capturado para *mp3*, visualização de ocorrências (desta vez sob a forma de notificações *pop-up*), geração de *playlists* e agrupamento dos ficheiros gerados numa pasta zipada estão também presentes nesta integração.

Como já foi descrito e testado na prova de conceito correspondente a esta integração, as funcionalidades para gravação de voz desenvolvidas em Java, deverão estar a correr num servidor virtual local na máquina do utilizador. Para solucionar esta questão, utilizou-se uma abordagem à qual já se tinha recorrido em outras implementações da empresa. Esta que consiste num instalador executado durante a instalação do STA, o qual dota a máquina destino com uma aplicação que corre um pequeno processo em *background* denominado por *AIRCVoice*, permitindo a execução do servidor virtual onde constam as funcionalidades necessárias.

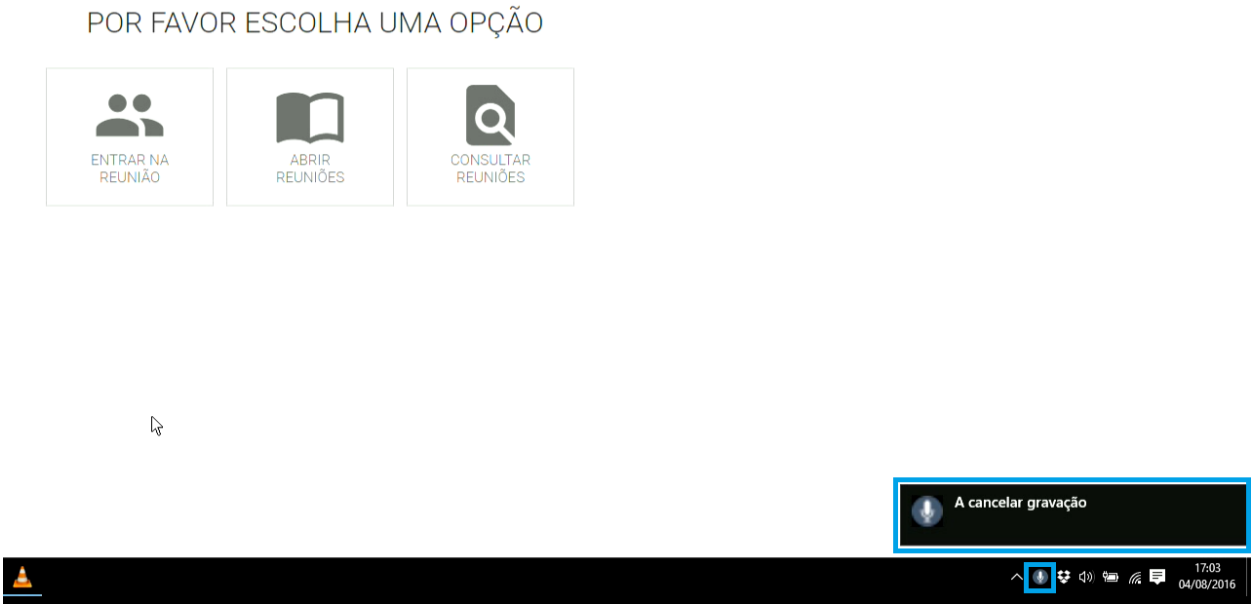


Figura 26 – Demonstração de uma notificação proveniente do *AIRCVoice* (também visível em *Background*)

5.5.4 Arquitetura dos Módulos Integrados

A arquitetura dos módulos integrados na STA consiste nos seguintes módulos:

- **Vista do STA –** Interface *web* desenvolvida em Angular2 que permite a interação do utilizador com o sistema
- **Comunicação e Interpretação de Comandos –** Código Typescript responsável por enviar os comandos em formato JSON para o servidor via *websockets*, assim como interpretar as mensagens provenientes do mesmo
- **AIRCVoice –** Servidor virtual local responsável pela comunicação com a aplicação cliente e pelas funcionalidades de gravação de voz
- **Interpretação e Redireccionamento dos Pedidos –** Funcionalidades responsáveis pela interpretação dos comandos JSON recebidos e invocação dos métodos correspondentes
- **Gestão de Ficheiros –** Código Java para as funcionalidades que manipulam os ficheiros
- **Biblioteca Gson –** Biblioteca responsável pelas funções que permitem interpretar as mensagens em formato JSON para um formato suportado pelo Java
- **Gestão do *Mixer* –** Código Java para as funcionalidades de gestão de microfones
- **Funcionalidades para Gravação –** Código Java para as funcionalidades responsáveis pela gravação do áudio
- **Gestão do Tempo de Gravação –** Código Java para as funcionalidades que gerem os tempos de gravação
- **Biblioteca JAVE –** Biblioteca responsável pelas funções que permitem a conversão dos ficheiros WAV para *mp3*
- **Funcionalidades Reconhecimento de Voz** – Código Typescript responsável pelas funcionalidades de interpretação de comandos e interligação com o serviço de reconhecimento de voz
- **Serviço de Reconhecimento de Voz –** Serviço remoto de reconhecimento de voz utilizado pela Web Speech API

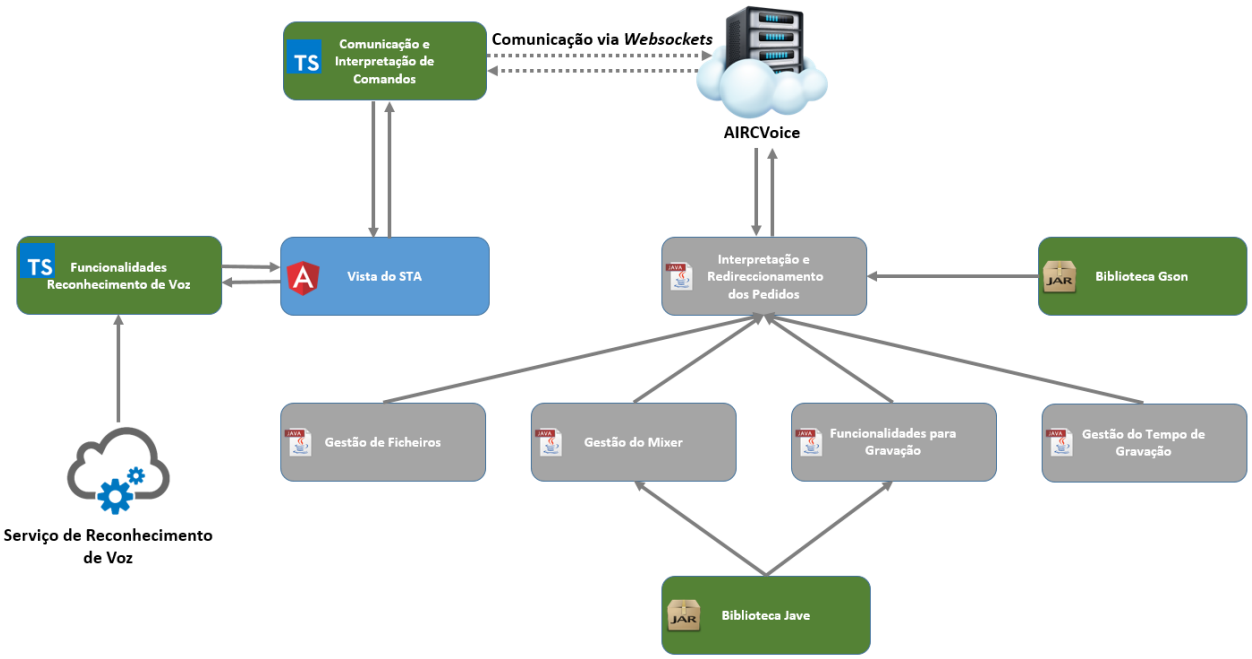


Figura 27 – Arquitetura Resumida dos Módulos Integrados no STA

6 Conclusões

Neste capítulo apresenta-se uma reflexão final relativamente ao trabalho desenvolvido no âmbito do estágio, os seus principais desafios e limitações, assim como uma visão sobre o trabalho futuro.

6.1 Reflexão sobre o Estágio Realizado

O presente estágio veio colmatar a experiência adquirida pelo estagiário no decorrer do Mestrado. Os desafios que iam sendo propostos e ultrapassados, permitiram ao estagiário desenvolver os seus conhecimentos tanto ao nível de ferramentas e linguagens de programação, como ao nível do planeamento e gestão de trabalho.

Outros dos fatores pertinentes a registar, foram o trabalho em equipa e a integração no contexto profissional, que forneceram excelentes bases para o futuro profissional do estagiário.

O facto deste estágio ter exigido um elevado grau de autonomia, devido ao seu teor de investigação e pesquisa de soluções, conferiu também ao estagiário o desenvolvimento de competências na resolução de problemas.

6.2 Revisão sobre os Objetivos Propostos

Recapitulando agora os objetivos propostos, apresenta-se de seguida o seu estado final, assim como algumas conclusões acerca dos mesmos.

- **Análise do estado da arte de *encoders* de reconhecimento de voz em Português (Mantido)** – este objetivo, para além da referida análise, acabou por se alargar a toda a investigação inicial realizada no âmbito do reconhecimento de voz, assim como, mais tarde também no âmbito da síntese e gravação de voz como efeito da alteração de objetivos. Deste modo, toda a informação reunida permitiu à entidade acolhedora adquirir conhecimento numa área que, até então, ainda não tinha sido explorada pela mesma. Esta informação (que faz parte do capítulo 2 deste relatório), foi disponibilizada à entidade acolhedora no final do período de estágio.
- **Seleção do *encoder* a adotar no projeto (Mantido)** – conforme já foi referido neste relatório, a oferta em termos de *encoders* grátis com suporte para a língua Portuguesa é praticamente inexistente. Após pesquisa e testes efetuados através da primeira prova de conceito, constatou-se que recorrendo à ferramenta Web Speech API, os principais requisitos pretendidos

inicialmente seriam satisfeitos sem a necessidade de recorrer a *encoders* complexos e pouco desenvolvidos para a língua Portuguesa.

- **Implementação de uma biblioteca de reconhecimento de voz (Removido)** – devido à inexistência de bibliotecas e *encoders* em Português já exposta anteriormente, esta tarefa requereria a implementação de raiz de uma biblioteca para esse fim. Tarefa essa que exigiria um maior grau de complexidade, tempo e recursos impossíveis de abranger durante o tempo de estágio. Porém, ao recorrer ao Web Speech API, esta implementação deixou de ser necessária, já que esta ferramenta já vem dotada de acesso ás bibliotecas da Google.

- **Possibilidade de transcrição de áudio para texto, inserindo essa informação em dados passíveis de usar na criação de documentos base da reunião (Removido)** – devido aos já mencionados obstáculos que inviabilizaram uma correta aplicação dos requisitos propostos por este objetivo, o mesmo foi substituído pela implementação de um sistema de reconhecimento de voz para interpretação de comandos.

- **Armazenamento do ficheiro de áudio em base de dados (Ajustado)** – este objetivo deixou de fazer sentido como sendo uma funcionalidade do sistema de reconhecimento de voz. No entanto, foi desenvolvido implicitamente no sistema de gravação sonora.

- **Integração na solução AIRC já existente - Sistema de Tratamento de Atas (Mantido)** – este objetivo traduziu-se pela integração das funcionalidades desenvolvidas durante as provas de conceito na solução final, o STA. Foram integrados o sistema de reconhecimento de voz e o sistema de gravação sonora.

- **Estudo das funcionalidades da ferramenta de reconhecimento de voz adotada (Acrescentado)** – o estudo referido neste objetivo foi levado a cabo durante a elaboração das diferentes provas de conceito desenvolvidas durante o período de estágio. Graças a este estudo, foi possível determinar quais as funcionalidades mais relevantes para a solução final, assim como testar a abordagem pretendida, utilizando os recursos disponibilizados pela ferramenta adotada.

- **Implementação de um sistema de reconhecimento de voz para interpretação de comandos (Acrescentado)** – após comprovado o correto funcionamento da ferramenta adotada no contexto de interpretação de comandos, objeto de estudo na segunda prova de conceito, esta funcionalidade foi integrada com sucesso no STA.

- **Análise e implementação de um sistema para síntese de voz (Acrescentado)** -apesar de existirem também funcionalidades para a síntese de voz na Web Speech API, a mesma não suportava o Português de Portugal. Foi realizada uma prova de conceito onde foi testada com sucesso uma outra ferramenta que satisfazia este requisito, o Voice RSS. No entanto, esta ferramenta não era totalmente gratuita, apresentando limites no número de pedidos diários

disponíveis gratuitamente, o que não seria viável numa aplicação em ambiente real. Por este motivo, e face à oferta limitada de ferramentas gratuitas com as características referidas, tendo ainda em conta que eventualmente a Web Speech API irá oferecer também suporte para Português Europeu, após a realização da primeira prova de conceito, foram suspensos os desenvolvimentos neste âmbito, com a intenção de serem retomados quando a ferramenta passar a disponibilizar esta funcionalidade.

- **Implementação de um sistema para gravação sonora (gravar áudio das reuniões) (Acrescentado)** – mais uma vez, após testes bem sucedidos durante as duas provas de conceito que abordaram este tema, as funcionalidades pretendidas por este objetivo foram objeto de integração no STA.

6.3 Desafios e Limitações

No que diz respeito aos desafios encontrados no decorrer do estágio, destaca-se o facto do estagiário ter conduzido a investigação e implementação de soluções sem nenhum ponto de partida. Apesar de inicialmente esta tarefa se ter revelado bastante desafiante, acabou por se tornar numa experiência muito enriquecedora, que fortaleceu o conhecimento e as bases tecnológicas do estagiário.

A limitação da oferta de soluções gratuitas/*open source* para reconhecimento e síntese de voz em Português, possíveis de integrar com a tecnologia do STA, foi também outro dos desafios mais relevantes. A grande maioria das soluções encontradas eram sujeitas a pagamento pelo que as opções de escolha eram bastante limitadas. No caso da síntese de voz, este facto acabou por inviabilizar a integração deste tipo de sistema no STA.

Algumas das linguagens de programação utilizadas no decorrer do trabalho (especialmente o JavaScript *e* Typescript) eram completamente novas para o estagiário, o que levou a que o mesmo investisse em algumas formações *online* durante a fase inicial do estágio. Com o decorrer das etapas de desenvolvimento, esta dificuldade foi sendo ultrapassada graças à experiencia que ia sendo adquirida.

A integração em contexto profissional foi também um pequeno desafio, dado que o estagiário não tinha ainda experiência profissional neste tipo de projetos. Esta situação depressa foi ultrapassada graças ao ambiente de caráter quase familiar que se vivia na empresa.

Finalmente, outro desafio importante foi a gestão do trabalho tendo em conta a dependência dos fatores externos. Dado que o desenvolvimento do sistema gráfico foi contratado a uma empresa externa à AIRC e acabou por sofrer alguns atrasos, o estagiário teve que gerir o seu trabalho tendo em conta a disponibilidade do sistema mencionado.

6.4 Trabalho Futuro

Tal como descrito anteriormente neste documento, apesar da investigação e implementações realizadas, os desenvolvimentos para um sistema de síntese de voz acabaram por ficar suspensos devido à falta de soluções gratuitas de síntese de voz em Português. A Web Speech API conta também com capacidades para síntese de voz, mas apenas para Português do Brasil, existindo rumores que brevemente ficará disponível também a opção em Português de Portugal. Tendo isto em mente e dado que o STA já integra soluções provenientes da Web Speech API, decidiu-se optar por esperar até que esta última opção fique disponível na API. Posto isto, o trabalho futuro acabará por passar por esta integração.

Outra proposta para eventuais desenvolvimentos futuros consistirá na expansão do número de comandos por voz disponíveis no STA, de modo a incrementar a interação por voz com as várias funcionalidades do sistema.

Um outro aspeto interessante que valorizaria ainda mais o sistema, seria a integração de um módulo de autenticação por voz. Neste caso, o utilizador do STA não necessitaria de se autenticar pelos meios tradicionais (*username* e *password*), mas sim através da sua voz, utilizando os meios físicos (neste caso, o microfone) já necessários pelo sistema de reconhecimento de voz implementado. Contudo esta característica iria exigir nova investigação e procura de uma solução que a permita implementar já que, à data, a Web Speech API não oferece esta funcionalidade.

Tendo agora conhecimento na área do reconhecimento e voz, seria possível também expandir este tipo de tecnologia a outro tipo de serviços ou ofertas. A título de exemplo, poder-se-ia implementar um sistema de atendimento automático ao cidadão, cujo funcionamento consistiria no reconhecimento de uma pergunta efetuada pelo mesmo através do telefone ou computador, análise dos padrões de palavras da pergunta e, posteriormente apresentação de uma lista de respostas possíveis baseadas nos padrões reconhecidos pelo sistema.

Referências

[1] P. Christensson, “Speech Recognition Definition,” Janeiro 2014. [Online]. Disponível: http://techterms.com/definition/speech_recognition. [Acedido em Dezembro 2015].

[2] M. Pinola, “History of voice recognition: from Audrey to Siri,” Novembro 2011. [Online]. Disponível: http://www.itbusiness.ca/news/history-of-voice-recognition-from-audrey-to-siri/15008. [Acedido em Dezembro 2015].

[3] IBM, “IBM Shoebox,” 2015. [Online]. Disponível: https://www-03.ibm.com/ibm/history/exhibits/specialprod1/specialprod1_7.html. [Acedido em Dezembro 2015].

[4] B. J. &. L. R. Rabiner, “Automatic Speech Recognition – A Brief History of the Technology,” Agosto 2004. [Online]. Disponível: http://www.idi.ntnu.no/~gamback/teaching/TDT4275/literature/juang_rabiner04.pdf. [Acedido em Dezembro 2015].

[5] A. Ossola, “Ever Wondered: How does speech-to-text software work?,” Agosto 2014. [Online]. Disponível: http://scienceline.org/2014/08/ever-wondered-how-does-speech-to-text-software-work/. [Acedido em Dezembro 2015].

[6] M. Pinola, “Speech Recognition Through the Decades: How We Ended Up With Siri,” Novembro 2011. [Online]. Disponível: http://www.pcworld.com/article/243060/speech_recognition_through_the_decades_how_we_ended_up_with_siri.html?page=2. [Acedido em Dezembro 2015].

[7] C. Woodford, “Voice recognition software,” Junho 2015. [Online]. Disponível: http://www.explainthatstuff.com/voicerecognition.html. [Acedido em Dezembro 2015].

[8] M. Collins, “Language Modeling,” 2013. [Online]. Disponível: http://www.cs.columbia.edu/~mcollins/lm-spring2013.pdf. [Acedido em Dezembro 2015].

[9] P. Blunsom, “Hidden Markov Models,” Agosto 2004. [Online]. Disponível: http://digital.cs.usu.edu/~cyan/CS7960/hmm-tutorial.pdf. [Acedido em Dezembro 2015].

[10] V. Ala-Keturi, "Speech Recognition Based on Artificial Neural Networks," 2004. [Online]. Disponível: http://www.cis.hut.fi/Opinnot/T-61.6040/pellom-2004/project-reports/project_07.pdf. [Acedido em Dezembro 2015].

[11] C. Woodford, "Neural Networks," Março 2016. [Online]. Disponível: http://www.explainthatstuff.com/introduction-to-neural-networks.html. [Acedido em Julho 2016].

[12] E. Grabianowski, "How Speech Recognition Works," Novembro 2006. [Online]. Disponível: http://electronics.howstuffworks.com/gadgets/high-tech-gadgets/speech-recognition4.htm. [Acedido em Dezembro 2015].

[13] C. G. Netto, "Sistema facilita a comunicação com surdo e deficiente auditivo," Novembro 2014. [Online]. Disponível: http://www.unicamp.br/unicamp/sites/default/files/jornal/paginas/ju_614_paginacor_09_web.pdf. [Acedido em Dezembro 2015].

[14] J. Schaeffer, "Voice Command Technology Enters the OR," Outubro 2014. [Online]. Disponível: http://www.fortherecordmag.com/archives/1014p24.shtml. [Acedido em Dezembro 2015].

[15] TechTarget, "Leveraging speech recognition technology in call centers," Maio 2009. [Online]. Disponível: http://searchcrm.techtarget.com/report/Leveraging-speech-recognition-technology-in-call-centers. [Acedido em Dezembro 2015].

[16] H. Wright, "Phonetic audio mining, audio searching, speech analytics," Março 2013. [Online]. Disponível: http://www.hakwright.co.uk/audio-mining.html. [Acedido em Dezembro 2015].

[17] Nuance Communications, "Dragon Financial Services Solutions," 2015. [Online]. Disponível: http://www.nuance.com/for-business/by-industry/financial-services/financial-services/index.htm. [Acedido em Dezembro 2015].

[18] AUTHENTIFY, "Voice Biometric Authentication," 2015. [Online]. Disponível: http://authentify.com/solutions/authentication-concepts/voice-biometric-authentication/. [Acedido em Dezembro 2015].

[19] R. Gallagher, "Watch Your Tongue: Law Enforcement Speech Recognition System Stores Millions of Voices," Setembro 2012. [Online]. Disponível: http://www.slate.com/blogs/future_tense/2012/09/20/speechpro_voicegrid_nation_voice

_recognition_software_for_use_by_law_enforcement_.html. [Acedido em Dezembro 2015].

[20] D. Froomkin, “THE COMPUTERS ARE LISTENING: Speech Recognition is NSA’s Best-Kept Open Secret,” Maio 2015. [Online]. Disponível: https://theintercept.com/2015/05/11/speech-recognition-nsa-best-kept-secret/. [Acedido em Dezembro 2015].

[21] SRI International, “Speech@SRI: Industries Served,” 2005. [Online]. Disponível: http://www.speechatsri.com/products/served.shtml. [Acedido em Dezembro 2015].

[22] I. M. &. H. R. Sharifzadeh, “Speech Recognition for Smart Homes,” 2008. [Online]. Disponível: https://kar.kent.ac.uk/48867/1/IntechChapter.pdf. [Acedido em Julho 2015].

[23] SAMSUNG, “SAMSUNG Smart TV: Voice Control,” 2013. [Online]. Disponível: http://www.samsung.com/ph/smarttv/voice_control.html. [Acedido em Dezembro 2015].

[24] Nuance Communications, “Dragon NaturallySpeaking Home Edition,” 2015. [Online]. Disponível: http://www.nuance.com/for-individuals/by-product/dragon-for-pc/home-version/index.htm. [Acedido em Dezembro 2015].

[25] TranscribeMe Inc., “Why 100% Accuracy is Not Available With Speech Recognition Software Alone,” Agosto 2013. [Online]. Disponível: http://transcribeme.com/blog-en/why-100-accuracy-is-not-available-with-speech-recognition-software-alone/. [Acedido em Dezembro 2015].

[26] C. Funk, “Understanding Speech Recognition Limitations,” Novembro 2014. [Online]. Disponível: http://www.speechtechmag.com/Articles/Editorial/Sponsored-Guest-Commentary/Understanding-Speech-Recognition-Limitations-100586.aspx. [Acedido em Dezembro 2016].

[27] Philips Speech Processing, “Philips FreeSpeech 2000: Cutting-Edge Speech Recognition Software for PC Users At Home, in the Office, on the Move,” Abril 1999. [Online]. Disponível: http://www.prnewswire.com/news-releases/philips-freespeech-2000-cutting-edge-speech-recognition-software-for-pc-users-at-home-in-the-office-on-the-move-74220697.html. [Acedido em Dezembro 2015].

[28] A. McEvoy, “FreeSpeech 2000: well worth the cost,” Fevereiro 2006. [Online]. Disponível: http://www.pcworld.co.nz/article/487825/freespeech_2000_well_worth_cost/. [Acedido em Dezembro 2015].

[29] C. Null, "Dragon NaturallySpeaking 13 review: Better than ever at letting you speak freely," Julho 2014. [Online]. Disponível: http://www.pcworld.com/article/2458363/dragon-naturallyspeaking-13-review-better-than-ever-at-letting-you-speak-freely.html. [Acedido em Dezembro 2015].

[30] Nuance Communications, "Dragon NaturallySpeaking Home Edition," 2015. [Online]. Disponível: http://www.nuance.com/for-individuals/by-product/dragon-for-pc/home-version/index.htm. [Acedido em Dezembro 2015].

[31] Nuance Communications, "Dragon NaturallySpeaking Premium Edition," 2015. [Online]. Disponível: Nuance Communications. [Acedido em Dezembro 2015].

[32] Nuance Communications, "Dragon Professional Individual," 2015. [Online]. Disponível: http://www.nuance.com/for-business/by-product/dragon/dragon-for-the-pc/dragon-professional-individual/index.htm. [Acedido em Dezembro 2015].

[33] Nuance Mobile, "Dragon Search," 2015. [Online]. Disponível: https://itunes.apple.com/pt/app/dragon-search/id341452950?mt=8. [Acedido em Dezembro 2015].

[34] Nuance Mobile, "Dragon Dictation," 2015. [Online]. Disponível: https://itunes.apple.com/pt/app/dragon-dictation/id341446764?mt=8. [Acedido em Dezembro 2015].

[35] Voice Tech Group, "Tazti: Products," 2015. [Online]. Disponível: https://www.tazti.com/products.html. [Acedido em Dezembro 2015].

[36] Brainasoft, "ARTIFICIAL INTELLIGENCE (AI) VIRTUAL ASSISTANT SOFTWARE," 2015. [Online]. Disponível: https://www.brainasoft.com/braina/#features. [Acedido em Dezembro 2015].

[37] Brainasoft, "ARTIFICIAL BRAIN," 2015. [Online]. Disponível: https://www.brainasoft.com/braina/artificial-brain.html. [Acedido em Dezembro 2015].

[38] Brainasoft, "Remote PC Voice Control," 2015. [Online]. Disponível: https://play.google.com/store/apps/details?id=com.brainasoft.braina. [Acedido em Dezembro 2015].

[39] J. McGregor, "Siri Vs. Cortana Vs. Google Now: The Future of Mobile," Julho 2015. [Online]. Disponível: http://www.forbes.com/sites/jaymcgregor/2015/07/06/siri-cortana-google-now-are-the-future-of-mobile/. [Acedido em Dezembro 2015].

[40] V. M., “Siri é melhor que Cortana e que Google Now, diz comparativo,” Novembro 2015. [Online]. Disponível: http://pplware.sapo.pt/apple/siri-e-melhor-que-cortana-e-que-google-now-diz-comparativo/. [Acedido em Dezembro 2015].

[41] Google, “Google Now,” 2015. [Online]. Disponível: https://www.google.com/landing/now/. [Acedido em Dezembro 2015].

[42] D. Beres, “Microsoft’s Cortana Is Like Siri With A Human Personality,” Julho 2015. [Online]. Disponível: http://www.huffingtonpost.com/entry/microsofts-cortana-is-like-siri-with-a-human-personality_55b7be94e4b0a13f9d1a685a. [Acedido em Dezembro 2015].

[43] P. Simões, “A Cortana vai chegar em breve a novos idiomas, pelo Windows 10,” Julho 2015. [Online]. Disponível: http://pplware.sapo.pt/microsoft/a-cortana-vai-chegar-em-breve-a-novos-idiomas-pelo-windows-10/. [Acedido em Dezembro 2015].

[44] Microsoft, “Configurar o Reconhecimento de Voz,” 2015. [Online]. Disponível: http://windows.microsoft.com/pt-pt/windows/set-speech-recognition#1TC=windows-7. [Acedido em Dezembro 2015].

[45] Microsoft, “O que posso fazer com o Reconhecimento de Voz?,” 2015. [Online]. Disponível: http://windows.microsoft.com/pt-pt/windows/what-can-do-speech-recognition#1TC=windows-7. [Acedido em Dezembro 2015].

[46] SpeechTexter, “TYPE WITH YOUR VOICE,” 2016. [Online]. Disponível: https://www.speechtexter.com/. [Acedido em Julho 2016].

[47] A. Agarwal, “ONLINE DICTATION,” 2016. [Online]. Disponível: https://dictation.io/. [Acedido em Julho 2016].

[48] TalkTyper, “TalkTyper,” 2015. [Online]. Disponível: https://talktyper.com/pt/index.html. [Acedido em Julho 2016].

[49] Speechpad, “Speech Pad - a new voice notebook,” 2016. [Online]. Disponível: https://speechpad.pw/. [Acedido em Julho 2016].

[50] J. M. P. &. W. B. Croft, “A Language Modeling Approach to Information Retrieval,” 1998. [Online]. Disponível: http://www.computing.dcu.ie/~gjones/Teaching/CA437/p275.pdf. [Acedido em Julho 2016].

[51] Computaris, "Deciphering Voice of Customer through Speech Analytics," 2015. [Online]. Disponível: http://pt.slideshare.net/Rsystemsgroup/deciphering-voice-of-customer-through-speech-analytics. [Acedido em Julho 2016].

[52] X. H. &. L. Deng, "An Overview of Modern Speech Recognition," Setembro 2009. [Online]. Disponível: https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/Book-Chap-HuangDeng2010.pdf. [Acedido em Julho 2016].

[53] Wikipedia, "List of speech recognition software," Junho 2016. [Online]. Disponível: https://en.wikipedia.org/wiki/List_of_speech_recognition_software. [Acedido em Julho 2016].

[54] R. M. Z., E.-B. Hazem M., I. Islam R e M. Nikos, "An Overview of Text-To-Speech Synthesis," 2010. [Online]. Disponível: http://www.wseas.us/e-library/conferences/2010/Corfu/CIT/CIT-14.pdf. [Acedido em Julho 2016].

[55] C. Woodford, "Speech Synthesizers," Janeiro 2016. [Online]. Disponível: http://www.explainthatstuff.com/how-speech-synthesis-works.html. [Acedido em Julho 2016].

[56] J. J. Ohala, "CHRISTIAN GOTTLIEB KRATZENSTEIN:PIONEERINSPEECH SYNTHESIS," Agosto 2011. [Online]. Disponível: https://www.researchgate.net/publication/267831415_CHRISTIAN_GOTTLIEB_KRATZENSTEIN_PIONEER_IN_SPEECH_SYNTHESIS. [Acedido em Julho 2016].

[57] G. Dalakov, "Wolfgang von Kempelen," Julho 2016. [Online]. Disponível: http://history-computer.com/Dreamers/Kempelen.html. [Acedido em Julho 2016].

[58] Haskins Laboratories, "Talking Heads: Simulacra," 2008. [Online]. Disponível: http://www.haskins.yale.edu/featured/heads/SIMULACRA/wheatstone.html. [Acedido em Julho 2016].

[59] K. Panos, "RETROTECHTACULAR: THE VODER FROM BELL LABS," Agosto 2014. [Online]. Disponível: http://hackaday.com/2014/08/12/retrotechtacular-the-voder-from-bell-labs/. [Acedido em Julho 2016].

[60] R. Mannell, "A Brief Historical Introduction to Speech Synthesis: A Macquarie Perspective," Janeiro 2010. [Online]. Disponível: http://clas.mq.edu.au/speech/synthesis/history_synthesis/. [Acedido em Julho 2016].

[61] H. Traunmüller, “Wolfgang von Kempelen's speaking machine and its successors,” Setembro 2000. [Online]. Disponível: http://www2.ling.su.se/staff/hartmut/kemplne.htm. [Acedido em Julho 2016].

[62] Instituto Camões, “A Pronúncia do Português Europeu,” 2008. [Online]. Disponível: http://cvc.instituto-camoes.pt/cpp/acessibilidade/capitulo2_1.html. [Acedido em Julho 2016].

[63] M. R. W. A. H. Daniel C. Burnett, “Speech Synthesis Markup Language (SSML) Version 1.0,” Setembro 2004. [Online]. Disponível: https://www.w3.org/TR/speech-synthesis/#S1. [Acedido em Julho 2016].

[64] Voice RSS, “Text-to-speech (TTS) Overview,” 2014. [Online]. Disponível: http://www.voicerss.org/tts/. [Acedido em Julho 2016].

[65] P. Birkholz, “About Articulatory Speech Synthesis,” 2015. [Online]. Disponível: http://www.vocaltractlab.de/index.php?page=background-articulatory-synthesis. [Acedido em Julho 2016].

[66] S. Lemmetty, “Applications of Synthetic Speech,” Junho 1999. [Online]. Disponível: http://research.spa.aalto.fi/publications/theses/lemmetty_mst/chap6.html. [Acedido em Julho 2016].

[67] J. Sass, “A Digital Reader That Says It All: The Intel Reader,” Maio 2011. [Online]. Disponível: http://scoop.intel.com/a-digital-reader-that-says-it-all-the-intel-reader/. [Acedido em Julho 2016].

[68] D. TAKAHASHI, “Intel introduces a digital book reader that reads aloud to the blind,” Novembro 2009. [Online]. Disponível: https://venturebeat.com/2009/11/09/intel-introduces-a-digital-book-reader-for-the-blind/. [Acedido em Setembro 2018].

[69] J. Oldman, “10 Free Screen Readers For Blind Or Visually Impaired Users,” Julho 2012. [Online]. Disponível: http://usabilitygeek.com/10-free-screen-reader-blind-visually-impaired-users/. [Acedido em Julho 2016].

[70] Nuance Communications, Inc., “Accessibility Solutions for Individuals,” 2016. [Online]. Disponível: http://www.nuance.com/for-individuals/by-solution/accessibility/index.htm. [Acedido em Julho 2016].

[71] S. McInnes, “Sweden’s railway stations get new text-to-speech technology for public announcements,” Novembro 2010. [Online]. Disponível:

http://nationalpainreport.com/swedens-railway-stations-get-new-text-to-speech-technology-for-public-announcements-885121.html. [Acedido em Agosto 2016].

[72] M. Tabini, "Vox technica: How Siri gets its voice," Novembro 2013. [Online]. Disponível: http://www.macworld.com/article/2056718/vox-technica-how-siri-gets-its-voice.html. [Acedido em Agosto 2016].

[73] ReadSpeaker Holding B.V., "The Benefits of Text to Speech," 2016. [Online]. Disponível: http://www.readspeaker.com/benefits-of-text-to-speech/. [Acedido em Setembro 2016].

[74] S. Lemmetty, "Problems in Speech Synthesis," Junho 1999. [Online]. Disponível: http://research.spa.aalto.fi/publications/theses/lemmetty_mst/index.html. [Acedido em Setembro 2016].

[75] P. Simões, "Skype Translator está já disponível para todos usarem," 2015. [Online]. Disponível: https://pplware.sapo.pt/microsoft/skype-translator-esta-ja-disponivelpara-todos-testarem/. [Acedido em Março 2018].

[76] Skype, "How do I set up and use Skype Translator?," 2018. [Online]. Disponível: https://support.skype.com/en/faq/FA34542/how-do-i-set-up-and-use-skype-translator. [Acedido em Março 2018].

[77] Google, "Translate," 2018. [Online]. Disponível: https://translate.google.com/intl/en/about/languages/. [Acedido em Março 2018].

[78] Nuance, "A more life-like automated voice for your brand," 2018. [Online]. Disponível: https://www.nuance.com/omni-channel-customer-engagement/voice-and-ivr/text-to-speech/vocalizer.html. [Acedido em Março 2018].

[79] Microsoft, "Apresentação do Narrador," Abril 2018. [Online]. Disponível: https://support.microsoft.com/pt-pt/help/22817/windows-10-narrator-introducing. [Acedido em Abril 2018].

[80] "Vozes de TTS," Abril 2018. [Online]. Disponível: https://support.microsoft.com/pt-pt/help/22797. [Acedido em Abril 2018].

[81] Natural Readers, "Natural Readers," 2018. [Online]. Disponível: https://www.naturalreaders.com/online/. [Acedido em Abril 2018].

[82] A. W. Black, "Festvox: An Online Speech Synthesizer," Dezembro 2017. [Online]. Disponível: http://www.festvox.org/voicedemos.html. [Acedido em Abril 2018].

[83] iSpeech, “iSpeech: Free Online Text to Speech,” 2018. [Online]. Disponível: https://www.ispeech.org/text.to.speech. [Acedido em Abril 2018].

[84] B. Jump, “THE DEFINITION AND HISTORY OF SOUND RECORDING,” Dezembro 2017. [Online]. Disponível: https://brianjump.net/2017/12/29/the-definition-and-history-of-sound-recording/. [Acedido em Abril 2018].

[85] Sound Recording History, “History of of Phonautograph,” 2018. [Online]. Disponível: http://www.soundrecordinghistory.net/history-of-sound-recording/phonautograph-history/. [Acedido em Maio 2018].

[86] Sound Record History, “History of Phonograph - First Phonograph,” 2018. [Online]. Disponível: http://www.soundrecordinghistory.net/history-of-sound-recording/phonograph-history/. [Acedido em Maio 2018].

[87] Art & Antiques, “Fonógrafo Edison Standard, 1899,” 2018. [Online]. Disponível: https://www.antiguedadestecnicas.com/productos/B-109.php. [Acedido em Setembro 2018].

[88] Sound Record History, “Gramophone History,” 2018. [Online]. Disponível: http://www.soundrecordinghistory.net/history-of-sound-recording/history-of-gramophone/. [Acedido em Maio 2018].

[89] M. Brain, “How Tape Recorders Work,” 2018. [Online]. Disponível: https://electronics.howstuffworks.com/gadgets/audio-music/cassette.htm. [Acedido em Maio 2018].

[90] S. E. Schoenherr, “Loudspeaker History,” 2001. [Online]. Disponível: http://history.sandiego.edu/gen/recording/loudspeaker.html. [Acedido em Maio 2018].

[91] Philips Historical Products, “Philips Compact Cassette,” 2013. [Online]. Disponível: http://www.philips-historische-producten.nl/cassette-uk.html. [Acedido em Maio 2018].

[92] Sound Recording History, “History of Digital Recording,” 2018. [Online]. Disponível: http://www.soundrecordinghistory.net/history-of-sound-recording/history-of-digital-recording/. [Acedido em Maio 2018].

[93] MUVI, “The Evolution of Online Audio Streaming,” Agosto 2017. [Online]. Disponível: https://www.muvi.com/blogs/evolution-online-audio-streaming.html. [Acedido em Maio 2018].

[94] J. Lee, "10 Common Audio Formats Compared: Which One Should You Use?," Maio 2016. [Online]. Disponível: https://www.makeuseof.com/tag/audio-file-format-right-needs/. [Acedido em Maio 2018].

[95] J. Ganz, "How Streaming Is Changing Music," Junho 2015. [Online]. Disponível: https://www.npr.org/sections/therecord/2015/06/01/411119372/how-streaming-is-changing-music. [Acedido em Maio 2018].

[96] S. Humphries, "Audacity 2.2.2 Review," Maio 2018. [Online]. Disponível: http://www.toptenreviews.com/software/multimedia/best-voice-recording-software/audacity-review/. [Acedido em Maio 2018].

[97] B. Bommer, "Adobe Audition Review," 2018. [Online]. Disponível: http://www.toptenreviews.com/software/multimedia/best-audio-editing-software/adobe-audition-review/. [Acedido em Junho 2018].

[98] S. Humphries, "WavePad Review," Maio 2018. [Online]. Disponível: http://www.toptenreviews.com/software/multimedia/best-voice-recording-software/wavepad-review/. [Acedido em junho 2018].

[99] Oracle Corporation, "An Introduction to Netbeans," 2016. [Online]. Disponível: https://netbeans.org/about/index.html. [Acedido em Julho 2016].

[100] Microsoft, "Visual Studio Code - Code Editing Redefined," 2016. [Online]. Disponível: https://code.visualstudio.com. [Acedido em Julho 2016].

[101] GitHub, "ATOM - A hackable text editor for the 21st Century," 2016. [Online]. Disponível: https://atom.io/. [Acedido em Julho 2016].

[102] Don Ho, "About Notepad++," 2016. [Online]. Disponível: https://notepad-plus-plus.org/. [Acedido em Julho 2016].

[103] Node.js, "About Node.js," 2016. [Online]. Disponível: https://nodejs.org/en/about/. [Acedido em Julho 2016].

[104] npm Inc., "About npm," 2016. [Online]. Disponível: https://www.npmjs.com/about. [Acedido em Julho 2016].

[105] HashiCorp, "About Vagrant," 2016. [Online]. Disponível: https://www.vagrantup.com/about.html. [Acedido em Julho 2016].

[106] Git, “About,” 2016. [Online]. Disponível: https://git-scm.com/about/. [Acedido em Julho 2016].

[107] Mozilla Developer Network, “Introduction to HTML,” Julho 2016. [Online]. Disponível: https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/Introduction. [Acedido em Julho 2016].

[108] Mozilla Developer Network, “CSS,” Janeiro 2016. [Online]. Disponível: https://developer.mozilla.org/en-US/docs/Web/CSS. [Acedido em Julho 2016].

[109] Mozilla Developer Network, “What is JavaScript?,” Outubro 2015. [Online]. Disponível: https://developer.mozilla.org/en-US/docs/Web/JavaScript/About_JavaScript. [Acedido em Julho 2016].

[110] ORACLE, “Learn About Java Technology,” 2016. [Online]. Disponível: https://www.java.com/en/about/. [Acedido em Julho 2016].

[111] ECMA International, “Introducing JSON,” 2013. [Online]. Disponível: http://www.json.org/. [Acedido em Julho 2016].

[112] Google, “What Is Angular?,” 2016. [Online]. Disponível: https://docs.angularjs.org/guide/introduction. [Acedido em Julho 2016].

[113] Microsoft, “TypeScript - JavaScript that scales,” 2016. [Online]. Disponível: https://www.typescriptlang.org/. [Acedido em Julho 2016].

[114] Mozilla Developer Network, “WebSockets,” Maio 2016. [Online]. Disponível: https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API. [Acedido em Julho 2016].

[115] Google, “Gson User Guide,” Junho 2016. [Online]. Disponível: https://github.com/google/gson/blob/master/UserGuide.md. [Acedido em Julho 2016].

[116] Mozilla Developer Network, “Web Speech Concepts and Usage,” Junho 2016. [Online]. Disponível: https://developer.mozilla.org/en-US/docs/Web/API/Web_Speech_API. [Acedido em Julho 2016].

[117] Voice RSS, “Voice RSS: Text-to-speech API Documentation,” 2014. [Online]. Disponível: http://www.voicerss.org/api/documentation.aspx. [Acedido em Julho 2016].

[118] Sauron Software, “JAVE,” 2012. [Online]. Disponível: http://www.sauronsoftware.it/projects/jave/index.php. [Acedido em Julho 2016].

[119] MindMaster - Educação Profissional, "Scrum: A Metodologia Ágil Explicada de forma Definitiva," Junho 2016. [Online]. Disponível: http://www.mindmaster.com.br/scrum/. [Acedido em Novembro 2018].

[120] IBM, "IBM Shoebox," 2018. [Online]. Disponível: https://www-03.ibm.com/ibm/history/exhibits/specialprod1/specialprod1_7.html. [Acedido em Setembro 2018].

[121] DIGSSTV, "O programa experimental DIGSSTV de VK4AES," Dezembro 2002. [Online]. Disponível: https://www.qsl.net/py4zbz/hdsstv/digsstv.html. [Acedido em Setembro 2018].

[122] Brainasoft, "Braina Pro Speech Recognition Software for PC," Setembro 2015. [Online]. Disponível: https://www.youtube.com/watch?v=WkWCi72BlbY. [Acedido em Setembro 2018].

[123] Haskins Laboratories, "Talking Heads: Simulacra," 2008. [Online]. Disponível: http://www.haskins.yale.edu/featured/heads/SIMULACRA/kratzenstein.html. [Acedido em Setembro 2018].

[124] reaktorplayer, "The Birth Of The Vocoder And It's Use In Modern Music (With Audio Examples)," Fevereiro 2015. [Online]. Disponível: https://reaktorplayer.wordpress.com/2009/11/20/the-birth-of-the-vocoder-and-its-use-in-modern-music/. [Acedido em Setembro 2018].

Anexo A – Mockups da Aplicação para Gravação Sonora das Reuniões

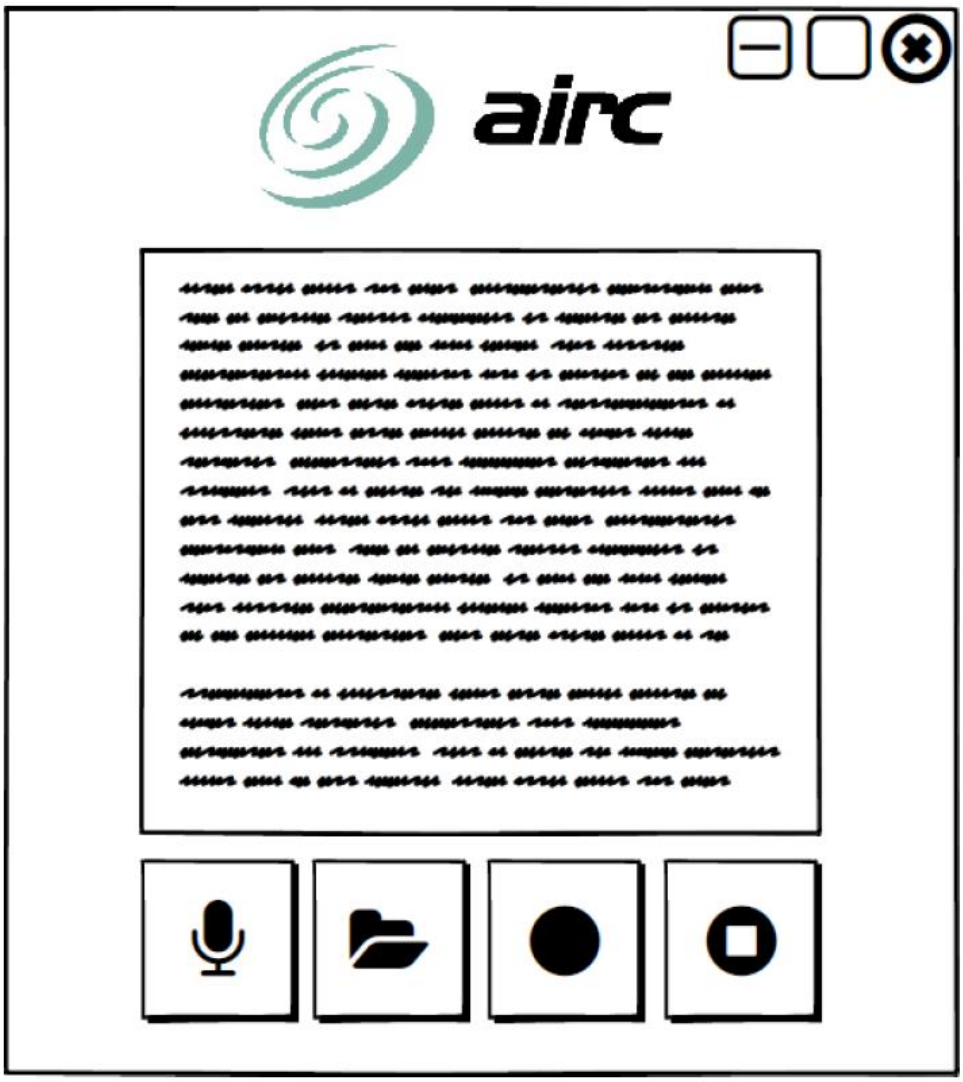


Figura 28 – Ecrã Inicial da Aplicação

Figura 29 – Ecrã da Aplicação após início da gravação

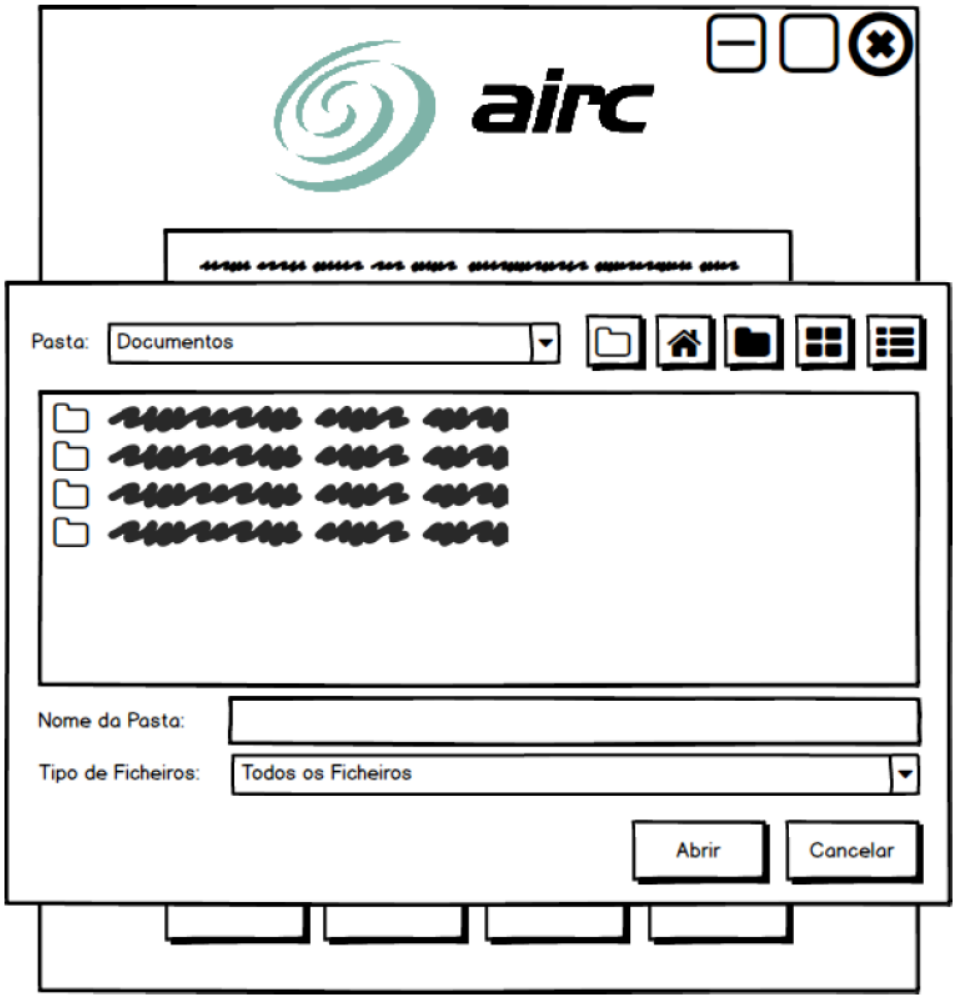


Figura 30 – Seleção da pasta destino para os ficheiros gerados durante a gravação

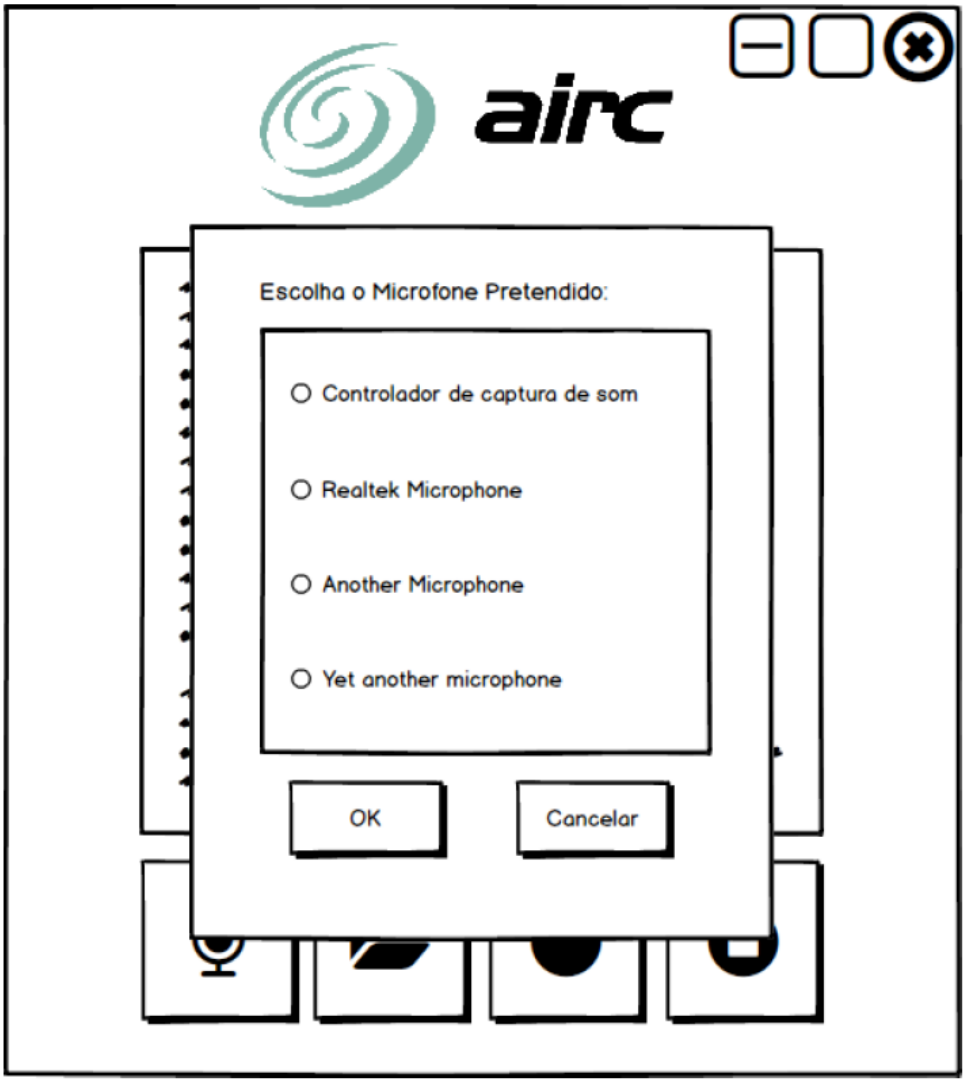


Figura 31 – Menu para seleção do Microfone

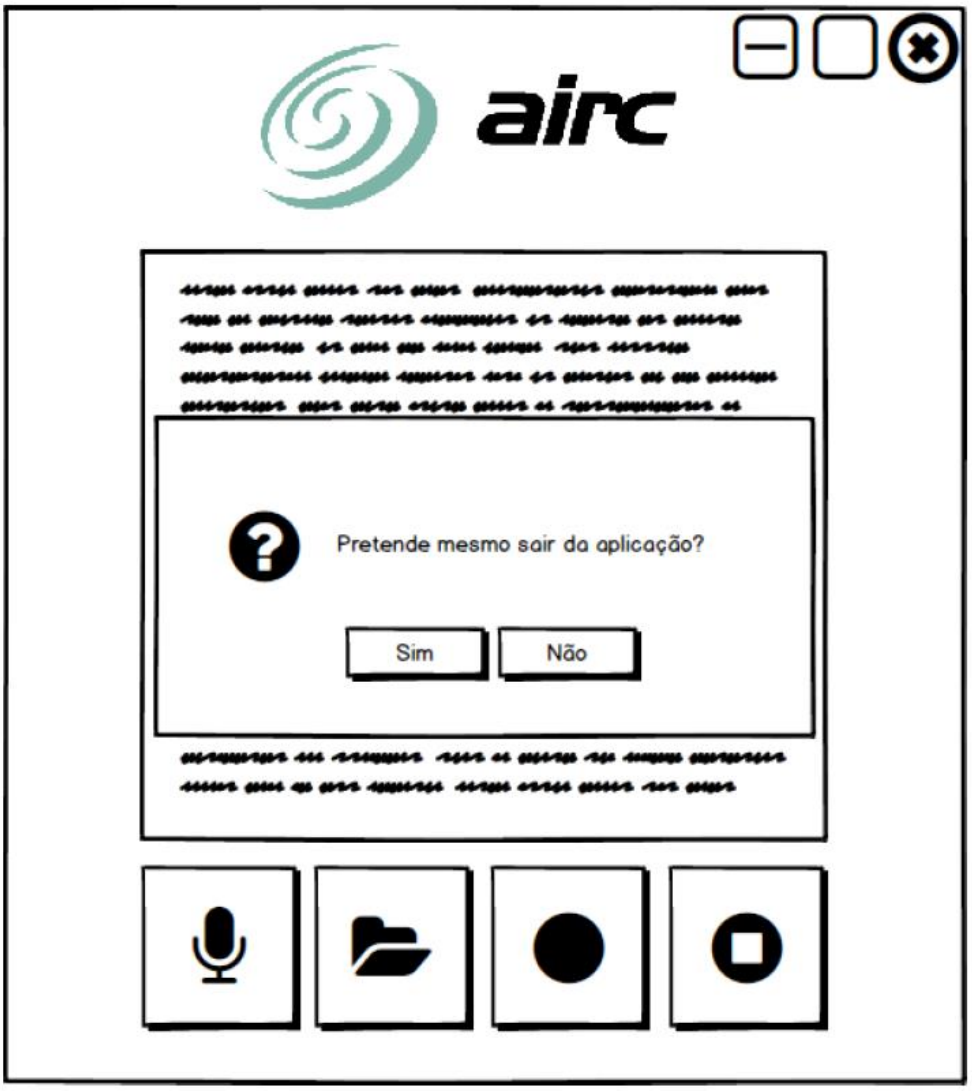


Figura 32 – Caixa de confirmação

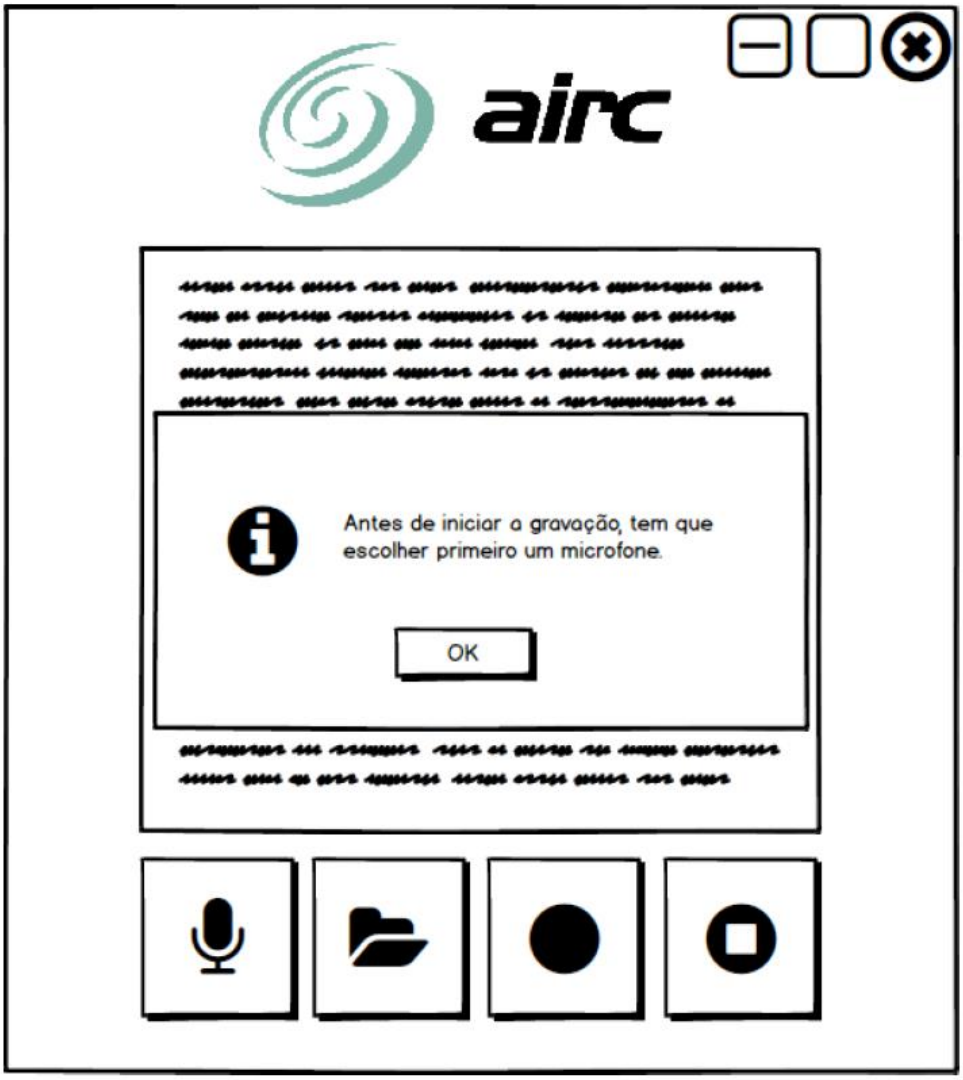


Figura 33 – Mensagem de alerta

Anexo B – Teste ao *Voice RSS*

Este teste foi realizado durante a investigação para a segunda prova de conceito. Pretendia-se escolher a configuração que melhor combinasse a qualidade sonora com o menor tamanho possível do ficheiro gerado.

Texto de teste a transcrever (parte de uma ata genérica):

"O requerimento de adjudicação dá lugar às notificações referidas no n.º 2 do art. 876.º do Cód. Proc. Civil e ainda a publicações que têm em vista a obtenção de outras propostas (cfr. n.º 1 deste normativo). Estas propostas são feitas em carta fechada, com sujeição ao regime da venda nesta modalidade, independentemente da natureza dos bens cuja adjudicação haja sido requerida. Este procedimento não se altera quando o agente de execução tenha optado pela venda por modalidade diversa da das propostas em carta fechada, maxime, por negociação particular. Caso a penhora recaia sobre bens móveis, o acto de abertura das propostas tem lugar perante o agente de execução e em data por este escolhido (art. 876.º, n.º 3, 2.ª parte). Assim, o SE deve proceder às notificações referidas no n.º 2 do art. 876.º do Cód. Proc. Civil e ainda à publicitação da adjudicação nos termos do art. 890.º, com a menção do preço oferecido pelo requerente da adjudicação, tendo em vista a obtenção de outras propostas, sendo aquela aceite apenas no caso de não aparecerem propostas em carta fechada ou, sendo apresentadas, estas não ofereçam preço superior."

Teste 1:

Duração do discurso: 1m:18s

Qualidade mínima – "8khz_8bit_mono":

- Tamanho do ficheiro gerado: 77Kb

- Qualidade: aceitável, mas com algum ruído

LINK: **http://tinyurl.com/8khz-8bit-mono**

Teste 2:

Qualidade mínima – "8khz_8bit_stereo":

- Tamanho do ficheiro gerado: 230Kb

- Qualidade: bastante aceitável, pouco ruído, discurso claro.

LINK: **http://tinyurl.com/8khz-8bit-stereo**

Teste 3:

Qualidade máxima – "48khz_16bit_stereo":

- Tamanho do ficheiro gerado: 1,19MB

- Qualidade: bastante boa, quase nenhum ruído, discurso bastante claro

LINK: **http://tinyurl.com/48khz-16bit-stereo**

Teste 4:

Qualidade máxima – "48khz_16bit_mono":

- Tamanho do ficheiro gerado: 613kb

- Qualidade: bastante boa, quase nenhum ruído, discurso bastante claro (a diferença em relação à anterior é mínima)

LINK: **http://tinyurl.com/48khz-16bit-mono**

Teste 5:

Qualidade média – "24khz_16bit_mono":

- Tamanho do ficheiro gerado: 307Kb

- Qualidade: bastante aceitável, pouco ruído, discurso claro.

LINK: **http://tinyurl.com/24khz-16bit-mono**

Teste 6:

Qualidade média – "24khz_16bit_stereo":

- Tamanho do ficheiro gerado: 613Kb

- Qualidade: bastante aceitável, pouco ruído, discurso claro (similar à qualidade mais alta).

LINK: **http://tinyurl.com/24khz-16bit-stereo**

Teste 7:

Qualidade baixa – "11khz_16bit_mono":

- Tamanho do ficheiro gerado: 153Kb

- Qualidade: bastante aceitável, algum ruido, discurso claro.

LINK: **http://tinyurl.com/11khz-16bit-mono**

Teste 8:

Qualidade baixa – "11khz_16bit_stereo":

- Tamanho do ficheiro gerado: 307Kb

- Qualidade: bastante aceitável, pouco ruído, discurso claro.

LINK: **http://tinyurl.com/11khz-16bit-stereo**

Algumas conclusões:

Se possível focar na qualidade mínima stereo para ter alguma qualidade em poucos Kb (230kb neste exemplo), ou na média mono que é um pouco melhor, mas com algum acréscimo de Kb (307 neste exemplo);

Ou:

Colocar uma opção que permita ao utilizador personalizar a qualidade do discurso, podendo ajustar a mesma conforme as possibilidades da velocidade do seu serviço de internet.

Anexo C - Proposta de Estágio

Âmbito

Para facilitar a realização e transcrição de reuniões de órgãos deliberativos pretende-se avaliar os atuais sistemas de reconhecimento de voz e dotar a aplicação de STA – Sistema de Tratamento de Atas da capacidade de integração deste tipo de sistemas. Dar-se-á preferência a sistemas de reconhecimento de fala natural e síntese de voz que sejam desenvolvidos em projetos *open source*. É objetivo deste estágio, desenvolver este módulo de forma a ser integrável em qualquer aplicação do ERP AIRC.

Objetivos

O presente projeto pretende atingir um conjunto de objetivos genéricos, que passam por diversos requisitos:

- Análise do estado da arte de *encoders* de reconhecimento de voz em Português.
- Seleção do *encoder* a adotar no projeto.
- Implementação de uma biblioteca de reconhecimento de voz.
- Possibilidade de transcrição de áudio para texto, inserindo essa informação em dados passíveis de usar na criação de documentos base da reunião.
- Armazenamento do ficheiro de áudio em base de dados.
- Integração na solução AIRC já existente - Sistema de Tratamento de Atas.

Programa de Trabalhos

O estágio consistirá nas seguintes atividades e respetivas tarefas:

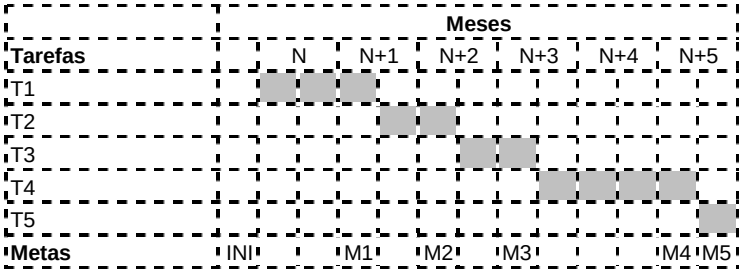
- **T1** – Formação e Análise – Aquisição de conhecimentos nas ferramentas de desenvolvimento identificadas para a elaboração do projeto e identificação das funcionalidades a implementar.
- **T2** – Desenho – Apresentação de soluções de acordo com as necessidades recolhidas.
- **T3** – Implementação – Construção dos âmbitos identificados, considerando a definição e criação das diferentes fases de produção.

- **T4** – Testes e Validação – Execução de testes para validação das tarefas desenvolvidas.
- **T5** – *Deployment* – Disponibilização dos resultados.

Calendarização das Tarefas

O plano de escalonamento dos trabalhos é apresentado em seguida:

Tabela 6 – Calendarização das tarefas a efetuar durante o estágio



INI		Início dos trabalhos
M1	(**INI** + 6 Semanas)	Tarefa T1 terminada
M2	(**INI** + 10 Semanas)	Tarefa T2 terminada
M3	(**INI** + 14 Semanas)	Tarefa T3 terminada
M4	(**INI** + 22 Semanas)	Tarefa T4 terminada
M5	(**INI** + 24 Semanas)	Tarefa T5 terminada