Universidade de Aveiro 2009

Departamento de Electrónica, Telecomunicações e Informática

Pedro Manuel Pinho Amorim

Síntese de Nomes em Português

Universidade de Aveiro 2009

Departamento de Electrónica, Telecomunicações e Informática

Pedro Manuel Pinho Amorim

Síntese de Nomes em Português

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Electrónica e Telecomunicações, realizada sob a orientação científica do Doutor António Teixeira, Professor Auxiliar do Departamento de Electrónica e Telecomunicações e Informática da Universidade de Aveiro.

Dedico este trabalho aos meus pais e a um grupo de amigos que sempre me acompanhou na minha vida universitária, “os mitras”.

O júri

Presidente	Doutora Ana Maria Perfeito Tomé Professora Associada da Universidade de Aveiro
Vogais	Doutor Carlos Jorge da Conceição Teixeira Professor Auxiliar da Faculdade de Ciências da Universidade de Lisboa
	Doutor António Joaquim da Silva Teixeira Professor Auxiliar da Universidade de Aveiro (orientador)

Agradecimentos

Gostaria de expressar toda a minha gratidão para todos os meus familiares e amigos mais próximos, que sempre me motivaram e acompanharam.

Dirijo o meu sincero reconhecimento e agradecimento ao Professor Doutor António Teixeira pela orientação, apoio e disponibilidade, que sempre teve para comigo, tornando possível o desenvolvimento de todo este trabalho.

Palavras-chave

Síntese de voz, text-to-speech (TTS), nomes próprios, conversão grafema-fonema, difones, MBROLA (Multi Band Resynthesis Overlap Add), SPICE (Speech Processing Interactive Creation and Evaluation), Festival, prompt.

Resumo

Pretendeu-se com o trabalho realizado no âmbito desta dissertação desenvolver um sistema capaz de sintetizar nomes em português de forma inteligível.

Em termos metodológicos, a opção passou pela utilização de ferramentas de apoio ao desenvolvimento de novas vozes para sistemas de síntese – concretamente o sistema SPICE – e adopção do sistema de síntese Festival.

Depois de apresentadas informações de base da área da síntese de voz, assim como informações sobre as funcionalidades dos programas usados neste trabalho (SPICE, MBROLA e Festival), na segunda parte da dissertação, descreveu-se todo o processo prático da criação da voz, fazendo uso do SPICE e MBROLA.

O sistema desenvolvido foi avaliado em termos da sua capacidade de efectuar correctamente a conversão grafema-fone e da inteligibilidade dos nomes sintetizados com resultados favoráveis para uma eventual aplicação prática.

Keywords

Speech synthesis, text-to-speech (TTS), proper names, grapheme-to-phoneme conversion, diphones, MBROLA (Multi Band Resynthesis Overlap Add), SPICE (Speech Processing Interactive Creation and Evaluation), Festival, prompts.

Abstract

The major goal of the work presented in this dissertation is to develop a system capable of synthesizing Portuguese names in an intelligible form.
In methodological terms the option was to use tools to support the development of new voices to synthesis systems - specifically the SPICE system - and adoption of the synthesis system Festival.
After presenting information on the area of speech synthesis as well as information on programs' features used in this work (SPICE, MBROLA and Festival), the second part of the thesis, describes the practical process of creating a voice using SPICE and MBROLA.
The developed system was evaluated in terms of their ability to perform properly the grapheme-phone as well as intelligibility of synthesized names with favorable results for a possible practical application.

Índice

Índice de Tabelas

Índice de Figuras/Gráficos

Lista de Abreviaturas

TTS – *Text-to-speech*

SPICE – *Speech Processing Interactive Creation and Evaluation*

PSOLA – *Pitch Synchronous Overlap and Add*

MBR-PSOLA – *Multi-Band Resynthesis Pitch Synchronous Overlap and Add*

TD-PSOLA – *Time-Domain Pitch Synchronous Overlap and Add*

MBROLA – *Multi Band Resynthesis Overlap Add*

CART – *Classification and Regression Trees*

IPA – *International Phonetic Alphabet*

SAMPA – *Speech Assessment Methods Phonetic Alphabet*

LTS – *Letter-to-Sound*

G2P – *Grapheme-to-Phoneme*

ASR – *Automatic Speech Recognition*

TCTS – *Théorie des Circuits et Traitement du Signal*

Capítulo 1 - Introdução

1.1. Motivação

No mundo actual, a rede que suporta a transmissão de informação falada está a crescer, pelo que a representação eficiente dos sinais de fala tem cada vez maior importância. Deste modo, o número de interfaces com fala tem aumentado, permitindo ao homem sistemas mais cómodos. O homem passa, assim, a não ser o único gerador ou destinatário da fala, podendo parte desta cadeia de comunicação ser implementada por sistemas automáticos [30].

"*A fala é preferível em situações de ocupação dos olhos e/ou mãos, sempre que seja impossível utilizar teclados, ratos ou ecrãs e também em casos de deficiência (sobretudo visual e auditiva). Muitas vezes, porém, a utilização da fala não é imperativa, mas sim vantajosa*" [31].

Focando-nos no caso particular da síntese de fala, podemos encontrar um enorme número de domínios de aplicação, como por exemplo: em serviços de telecomunicações, aplicações educativas, aplicações militares, em controlo industrial, em alarme para situações de risco, aplicações em terapias (nomeadamente da fala), para a ajuda de pessoas com algum tipo de deficiência (em especial visual e auditiva), controlo de listas de espera em hospitais, sinalização de paragens em transportes públicos, sistemas de navegação GPS, *smartphones*, etc. O grau de aceitação destas aplicações, por parte do público, depende essencialmente da inteligibilidade e naturalidade da pronunciação.

No caso desta dissertação, focamo-nos na síntese de voz, mais concretamente, de nomes próprios.

A síntese de nomes revela-se um grande desafio, pois um nome pode ter diversas origens etimológicas, "viajar" de língua para língua, mostrando diferentes graus de ajustamento à língua de chegada. Por vezes, o nome é acolhido em línguas onde as regras de conversão grafema-fonema diferem da sua língua de origem, o que proporciona enormes dificuldades na criação de pronúncias. Como consequência, muitos nomes estrangeiros não podem ser identificados pela sua ortografia, pois estão em conformidade

com a língua nativa, podendo haver casos de nomes, que são inseridos numa língua sem sofrer os seus processos linguísticos. Deste modo, quando se transcreve um nome, não há, por vezes a possibilidade de identificar a sua origem, logo o processamento de regras pode ser inadequado.

No caso da síntese com textos arbitrários, é normal haver um grupo de fonemas de grande frequência, capazes de fazer a cobertura da maioria das palavras, o que não se verifica na síntese de nomes próprios. Este tipo de síntese necessita de bases de dados muito maiores, devido ao enorme número de nomes próprios a serem adquiridos, para se obter uma boa cobertura.

A complexidade de um sistema de síntese de fala pode variar significativamente conforme o domínio da aplicação. Se se tratar de um sistema capaz de apenas sintetizar um conjunto de mensagens ou palavras previamente gravadas, a complexidade é mínima. No caso de sistemas capazes de sintetizar fala a partir de qualquer texto, a complexidade é máxima.

Apesar de tudo, os sistemas actuais já têm elevada inteligibilidade e alguma naturalidade em domínios restritos, no entanto, em domínios mais abrangentes, há necessidade de melhoria a todos os níveis, mas em especial na prosódia. Desta forma, e pensando no futuro, pretende-se que os sintetizadores de voz tenham as seguintes características: inteligibilidade elevada, mesmo com ruído; timbre natural; sistemas multi-língua e multi-dialectais, importantes para quebrar barreiras de comunicação entre pessoas que falam línguas diferentes; sínteses com ritmo e acentuação, orientadas para aplicações; capacidade de exprimir emoção. Deste modo, concluímos aqui a ideia de que a conversão texto/fala é um problema ainda por resolver. Trata-se de uma área de trabalho bastante complexa e, sobretudo, fortemente interdisciplinar. Envolve conhecimentos de Engenharia (processamento de sinais, aprendizagem automática, técnicas de busca, etc.) e de Linguística (Fonética, Fonologia, Prosódia, Morfologia, Sintaxe), mas não só.

Apesar de alguns problemas descritos, já é possível uma síntese de nomes de alta qualidade, em especial para inglês. Para outras línguas menos faladas, esta síntese é grosseira, daí a necessidade da implementação e desenvolvimento de sistemas de síntese de nomes em português.

Uma síntese de nomes inteligível e com voz natural revela-se assim um desafio.

1.2. Objectivos

Esta dissertação tem como objectivo principal a criação de um sistema capaz de sintetizar nomes próprios em português, tendo como entrada texto (nomes).

Inicialmente, pretendia-se que a criação da voz fosse realizada com o mínimo de recursos e com o mínimo de tempo, através de processos simples, quase todos eles baseados em algoritmos já programados e automatizados, sendo apenas necessária a recolha de dados (em formato texto e formato áudio). Todo este processo seria baseado num único *software on-line*, o SPICE.

Devido a limitações do SPICE na recolha de dados áudio, houve necessidade de alterar o procedimento de criação da voz. Assim, pretende-se que esta se desenvolva segundo um processo híbrido, usando-se toda a estrutura da voz criada pelo sistema SPICE e um sintetizador externo, o MBROLA, para a geração das ondas sonoras. Visto os modelos de prosódia gerados pelo SPICE não serem os melhores, devido à falta de dados áudio para treino, pretende-se também aplicar novos modelos baseados em vozes já existentes.

Para além de todos estes objectivos práticos, é esperado com a realização deste trabalho, a aprendizagem de todos os processos de um sintetizador TTS, do funcionamento de todos os seus componentes, de vários métodos de síntese de voz, de vários modelos capazes de gerar prosódia e da estrutura e funcionamento de todos os *softwares* usados no trabalho (SPICE, Festival e MBROLA).

Outro dos objectivos deste trabalho consiste na avaliação das potencialidades e limitações do SPICE na criação de uma voz para o português.

O sistema deve ser criado tendo em vista um cenário de aplicação real (e.g. a automatização da chamada pelo nome em filas de espera num hospital ou a automatização da informação sobre estações no metro, entre outros).

1.3. Estrutura da Dissertação

Para a realização desta dissertação, foram desenvolvidas diversas tarefas, que, grosso modo, correspondem aos vários capítulos deste trabalho.

Assim, neste primeiro capítulo, a introdução, são apresentadas todas as motivações para a realização do trabalho, os seus objectivos e a estrutura da dissertação.

No capítulo 2, são expostos os fundamentos teóricos essenciais à criação de um sistema TTS. São explicados, inicialmente, todos os elementos constituintes do sistema, nomeadamente o bloco responsável pelo tratamento linguístico e o bloco responsável pela produção de sinais sonoros. No primeiro bloco, são especificadas todas as suas partes integrantes – analisador de texto, conversor de texto para fonemas e o gerador de prosódia – assim como todos os processos que decorrem durante o processamento de cada uma destas. Após a descrição dos processos linguísticos, são abordados alguns dos métodos mais populares de síntese de voz, tais como a síntese articulatória, a síntese de formantes e a síntese por concatenação.

No que diz respeito à síntese por concatenação, foi incluído também uma pequena secção sobre a escolha do melhor segmento para a síntese e a descrição da síntese com cada um dos segmentos de uso mais frequente. Foram apresentadas ainda, as técnicas TD-PSOLA e MBR-PSOLA.

Neste capítulo, é também apresentada a síntese por unidades de selecção e a síntese em domínio restrito.

Por fim, são referidos alguns problemas, específicos da síntese de nomes próprios.

No capítulo 3, são descritas todas as ferramentas usadas nesta dissertação. Em primeiro lugar, é descrito o Festival e toda a sua estrutura, assim como todos os passos e procedimentos necessários para a criação de uma voz em Festival. Neste capítulo, está também incluída a descrição do sistema MBROLA, assim como a criação de uma base de dados de difones usada neste *software*. Por fim, é apresentado o SPICE e todos os passos necessários para a criação de uma voz em SPICE.

No capítulo 4, são explicados todos os procedimentos práticos efectuados, no sentido da criação de uma voz capaz de sintetizar nomes próprios em português. Ainda neste capítulo, é feita a avaliação da voz criada, com um teste de avaliação para a conversão grafema-fonema e um outro para a identificação de nomes.

No capítulo 5, o capítulo das conclusões, é feito um pequeno balanço do trabalho efectuado, são referidos os principais resultados práticos alcançados e respectivas conclusões e, finalmente, são dadas algumas sugestões de trabalho futuro.

Este último capítulo é seguido da bibliografia consultada e de alguns endereços electrónicos relevantes.

Capítulo 2 - Síntese de Voz

2.1. Sistemas TTS

Um sistema TTS é um sistema capaz de produzir voz de forma artificial, efectuando a conversão de um texto de entrada em fala.

Apesar da sua diversidade, a maioria dos TTS apresenta uma estrutura comum. Podemos assim distinguir dois blocos principais: o de Processamento de Linguagem Natural e o de Processamento de Sinal (ondas sonoras). A seguinte figura mostra o diagrama de um sistema TTS [2].

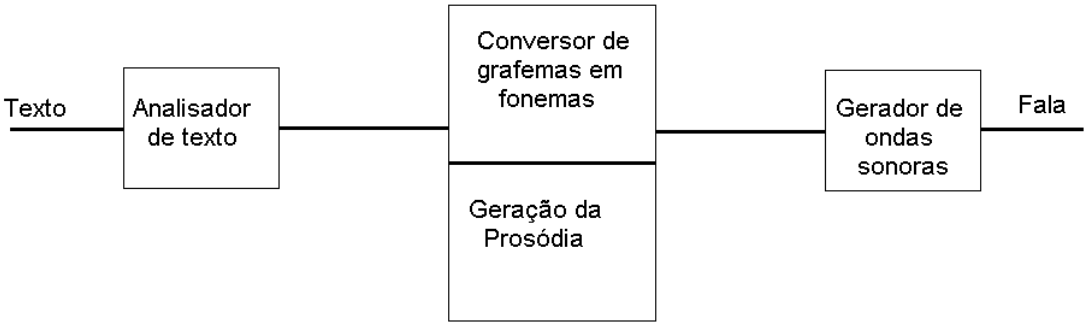

Figura 1 – Arquitectura geral de um sistema TTS

O primeiro bloco é responsável por todo o tratamento de texto e de etiquetagens necessárias para o processamento do segundo bloco (o sintetizador propriamente dito). O primeiro bloco é passível de ser subdividido em três partes: Analisador de Texto; Conversor de Texto para Fonemas; Gerador de Prosódia [2].

No Analisador de Texto é feita a separação de palavras e frases e a normalização do texto; no Conversor de Texto para Fonemas é feita a desambiguação de homógrafos, a análise morfossintatica, a conversão grafema-fonema e a divisão silábica; por fim, no Gerador de Prosódia são tratadas as pausas, a entoação, a duração, o ritmo e a acentuação das palavras [6].

A prosódia é muito importante, pois é em boa parte responsável pela naturalidade da voz gerada, requisito essencial para que um sintetizador seja aceite pelos utilizadores.

2.1.1. Análise de Texto

O Analisador de Texto tem como principal função minimizar a variabilidade do texto de entrada, para isso é essencial uma análise de texto, onde se fazem as seguintes operações [6].

Separação de frases: o método mais simples baseia-se em encontrar a pontuação de fim de frase, fazendo-se a divisão a partir desse lugar (salvo os casos onde os pontos estão associados a números ou abreviaturas).

Separação de palavras: ocorre sempre que se encontram espaços em branco, no caso de palavras com hífen, a palavra composta é tratada como duas palavras autónomas.

Normalização do texto: é a parte do sistema onde se faz a conversão de toda a espécie de símbolos, abreviaturas, siglas, acrónimos, números, para sequências ortográficas adequadas para a subsequente transcrição fonética, no idioma em que se pretende fazer a síntese.

2.1.2. Análise Fonética

O Conversor de Texto para Fonemas tem a função de converter um texto numa sequência de fonemas, daí a necessidade de uma análise fonética, através das seguintes operações [1].

Análise semântica: é uma análise pouco desenvolvida nos sistemas de síntese, que se ocupa da significação das palavras e da evolução do seu sentido, importante em caso de palavras homógrafas.

Análise morfológica: estuda a estrutura interna das palavras, bem como o processo de formação e variação destas.

Conversor grafema-fonema: faz a conversão dos grafemas nos respectivos fonemas. Contudo, esta tarefa é de extrema dificuldade, pois não existe uma correspondência directa entre grafia e forma de pronunciação.

Há vários tipos de abordagens, tais como: o uso de um dicionário; o uso de regras com base em teorias da fonologia, difíceis e demoradas de desenvolver; usar primeiro um

dicionário, caso este falhe, usar regras; ou usar regras obtidas por aprendizagem automática.

Divisor silábico: faz a identificação da unidade silábica, quer para a implementação de algumas regras do conversor grafema-fonema, quer para a modelização da prosódia, ao nível da duração e entoação.

2.1.3. Geração Prosódica

O Gerador de Prosódia procura controlar algumas características da prosódia, temporais e tonais, para que a voz seja produzida da forma mais natural possível [1], neste sentido é necessária a construção de vários modelos, para a geração de pausas, durações, acentuação e entoação.

Pausas: uma das maneiras mais seguras para a geração de pausas consiste em usar a pontuação, considerando sempre a sua ocorrência e duração.

Duração: Em geral costuma-se assumir uma duração para cada fonema, podendo esta depois variar por vários motivos, tais como: inserção de pausas devido a pontuação; aumento da velocidade no início de palavras ou em frases longas; aumento da velocidade nos clusters de consoantes; pausas antes do "e", quando se juntam orações.

Variação da entoação (*pitch*): a entoação varia essencialmente consoante o tipo de frase (declarativa, interrogativa, exclamativa, etc).

Acentuação: a acentuação de certas sílabas é importante, pois permite-nos distinguir palavras que nos poderiam parecer iguais (por exemplo, algumas homógrafas) e também para realçar as palavras mais importantes de uma frase.

Ritmo: o ritmo é marcado, de certo modo, pelas sílabas acentuadas que tendem a ocorrer em intervalos de tempo regulares. Já as sílabas não acentuadas são comprimidas nos tempos entre acentos.

O segundo bloco do TTS recebe como entrada *strings* de fonemas, com anotações de prosódia, processando-as de modo a gerar uma voz falada. Existem inúmeras técnicas para a síntese das ondas sonoras, que serão abordadas na próxima secção.

2.2. Técnicas de Síntese

Nesta secção, são apresentados três tipos de técnicas de síntese distintos: a síntese articulatória, a síntese de formantes e a síntese por concatenação, dentro desta última técnica abordarei a concatenação com vários tipos de segmentos (fonemas, difones, sílabas, palavras) e abordarei as técnicas TD-PSOLA e MBR-PSOLA, usadas para suavizar os pontos de concatenação.

Todas as subsecções da secção 2.2 foram baseadas no capítulo 16 do livro de Paul Taylor [1] e na tese de Sami Lemmetty [9], sendo todas as outras referências adicionais apresentadas em cada uma das subsecções.

2.3. Síntese Articulatória

Através deste método, procura-se recriar um modelo capaz de simular de forma fiel os processos naturais da fala, de modo a obter um sinal de fala realista.

À partida, esta técnica de síntese parece ser a mais lógica e eficaz mas, na verdade, a sua implementação tem um grau de complexidade muito grande.

Normalmente, uma síntese deste tipo engloba dois subsistemas: um modelo anotomo-fisiológico e um modelo de produção do som nas estruturas envolvidas na fala.

O modelo anatomo-fisiológico consiste em transformar as variações das posições de todos os articuladores ao longo da produção de fala em áreas transversais do tracto vocal, tendo-se em consideração parâmetros como: a abertura dos lábios, a protrusão dos lábios, a altura da língua, a posição da ponta da língua, entre outros.

O segundo modelo baseia-se num conjunto de equações que descrevem as propriedades acústicas do tracto vocal. Apesar destes modelos descritos, existem outros adicionais, essenciais para uma melhor modulação da fala, por exemplo modelos para a parte fonatória, onde parâmetros como a abertura da glote, tensão das cordas vocais e a pressão pulmonar são essenciais.

Comparativamente a outros sistemas de síntese, a síntese articulatória tem recebido pouca atenção. Tal facto deve-se a todo um conjunto de dificuldades na obtenção de

informação morfológico-dimensional sobre o tracto e as cordas vocais durante a produção da fala, na obtenção de informação sobre a dinâmica dos articuladores e também devido à morosidade e complexidade dos cálculos necessários [18], o que torna quase impossível uma modelação precisa do tracto vocal. Desta forma, os processos de modelação são baseados em informações referentes a configurações quase sempre estáticas.

A síntese articulatória é pouco usada nos sistemas actuais, no entanto, devido ao desenvolvimento de métodos e de recursos computacionais e clínicos, a utilização de síntese articulatória revela um enorme potencial para o futuro [3].

2.4. Síntese de Formantes

Antes de apresentar a síntese de formantes, importa explicar alguns conceitos essenciais para que se possa perceber o método em questão.

Formante: Um formante é o pico de intensidade no espectro de um som, ou seja a concentração de energia (amplitude das ondas), que ocorre numa certa frequência. A vibração das pregas vocais produz ondas sonoras com um espectro de frequência bastante distribuído. Estas são filtrados pelo trato vocal, sendo algumas frequências atenuadas e outras reforçadas. As frequências que são fortemente reforçadas são precisamente os principais formantes das emissões sonoras [21].

Cada som da fala humana tem os seus formantes característicos, ou seja, uma diferente distribuição de energia sonora entre os formantes.

Modelo fonte-filtro: Como mostra a figura abaixo, há uma fonte que gera um espectro, que altera a sua forma de acordo com o filtro. Assim, o espectro de saída tem características tanto da fonte quanto do filtro [26].

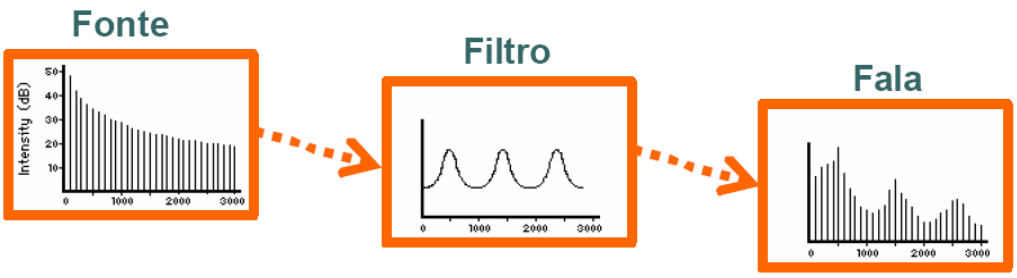

Figura 2 – Esquema de um modelo fonte-filtro

A produção da fala pode ser dividida em duas partes independentes:

• Fontes de som, que podem ser:

Fontes sonoras –> Resultante da vibração periódica das cordas vocais.

Fontes de ruído –> Resultante da passagem rápida do ar por uma constrição.

• Filtros que modificam o sinal (sistemas) – simulam as ressonâncias do tubo acústico, formado pela faringe, cavidade oral e lábios [26]. A função transferência do filtro pode ser modelada por um conjunto de pólos, onde cada pólo produz um pico no espectro, o formante. Há a necessidade de se introduzir zeros no modelo para simular consoantes nasais, vogais nasais e fricativas.

Síntese de formantes

É um método que se baseia em modelos fonte-filtro, onde a fonte gera uma forma de onda excitada, que passa num conjunto de filtros. Os filtros consistem em ressonâncias e anti-ressonâncias, que modelam a resposta em frequência do tracto vocal.

Para produzir uma fala inteligível, são necessários pelo menos 3 formantes, e até 5 formantes para que o discurso seja de elevada qualidade. Cada formante é normalmente modelado por um ressoador de dois pólos.

De acordo com a disposição dos filtros, é possível ter três tipos de configurações neste tipo de síntese: em cascata, em paralelo ou uma combinação destes dois formatos.

Num sintetizador de formantes em cascata, o conjunto de ressoadores é ligado em série e a saída de cada um destes é aplicada à entrada do seguinte. A estrutura em cascata necessita unicamente, para controlo, de informação da frequência de cada formante. A principal vantagem desta estrutura deve-se ao facto de as amplitudes relativas dos

formantes não precisarem de controlo individual, pois há um só ganho para todos os formantes.

A estrutura em cascata tem bons resultados para todos os sons vozeados e para sons não nasais, pois necessita de menos informação de controlo, do que, por exemplo uma estrutura em paralelo, e é de mais fácil implementação. Apesar de tudo, apresenta lacunas, não sendo capaz de produzir sons fricativos e plosivos.

Todas estas informações podem ser observadas através da seguinte figura, onde todos os "F" se referem à frequência de cada formante.

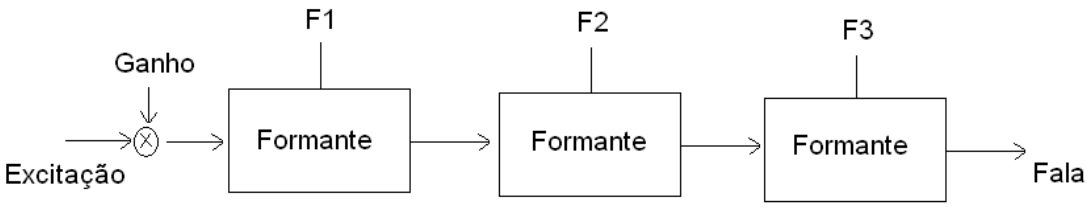

Figura 3 – Configuração em série

Num sintetizador de formantes em paralelo, o conjunto de ressoadores está dispostos em paralelo mas, por vezes, há necessidade de recorrer a ressonadores extra para a reprodução de sons nasais. O sinal de excitação é aplicado à entrada de todos os ressonadores em simultâneo e as saídas são todas somadas. A adjacência das saídas dos formantes deve ser somada em oposição de fase para evitar zeros indesejáveis ou anti-ressonâncias na resposta à frequência.

A estrutura em paralelo permite o controlo da largura de banda e do ganho individual de cada formante, pelo que necessita de maior informação de controlo. O uso da estrutura em paralelo é melhor para sons nasais, fricativos e pausas, contudo, há certas vogais que não podem ser moduladas por configurações em paralelo, assim como em cascata.

A seguinte figura mostra um formato em paralelo, onde os "BW" representam larguras de banda, os "F "as frequências dos formantes e os "A" os ganhos antes de cada ressonador.

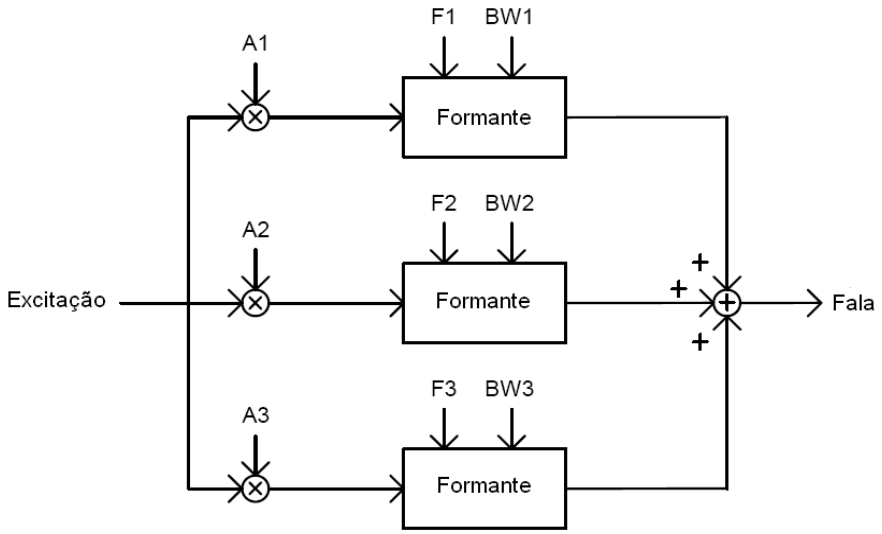

Figura 4 – Configuração em paralelo

Na tentativa de melhorar a qualidade da síntese surgiram novas configurações, que resultam da combinação das estruturas em paralelo e em cascata.

Em 1980, Dennis Klatt propôs um sintetizador de formantes que incorporava estas duas configurações, era de grande complexidade, tinha ressonâncias adicionais e anti-ressonâncias para os sons nasais, um sexto formante para os ruídos de alta-frequência e radiação característica. O sistema utilizava um modelo de excitação bastante complexo, controlado por 39 parâmetros com *updated* a cada 5 ms. Mais informação sobre este sintetizador de Klatt pode ser encontrada em [1].

2.5. Síntese por Concatenação

Na síntese por concatenação a voz sintetizada é criada concatenando-se pedaços de fala pré-gravada, armazenada em bases de dados.

Nesta secção, serão apresentados vários tipos de concatenação, com segmentos de diferentes tamanhos e serão apresentados também algoritmos capazes de atenuar as descontinuidades nos pontos de concatenação (zona de junção de dois segmentos).

2.5.1. Escolha da Melhor Unidade

Na síntese por concatenação, é possíveis optar por vários segmentos, tais como: fonemas, difones, trifones, sílabas, palavras e frases.

Um dos aspectos mais importantes na síntese por concatenação é encontrar o correcto comprimento da unidade, de acordo com as nossas necessidades.

A selecção é geralmente um compromisso entre as unidades mais curtas e as unidades mais longas. Com unidades longas, o resultado da síntese tem um elevado grau de naturalidade, menos pontos de concatenação, havendo um bom controlo na co-articulação. Por sua vez, tem a desvantagem de necessitar de bases de dados enormes, se pretendermos cobrir um número considerável de palavras. Com unidades mais pequenas, há necessidade de menor quantidade de memória, mas a recolha de amostras, a sua etiquetagem e contextualização, são um processo mais complexo e produzem um discurso menos natural.

Para ter alta qualidade na síntese, aquando da escolha da melhor unidade é preciso ter em atenção todo um conjunto de factores, a saber:

- As unidades escolhidas devem conduzir o sistema a uma baixa distorção nos pontos de concatenação. Deste modo, quanto menos pontos de concatenação tiver, melhor serão os resultados, daí que o uso de palavras e frases seja mais vantajoso.

  Já que alguns pontos de articulação são inevitáveis, devemos proporcionar o mínimo de distorção nesses pontos.

- As unidades devem conduzir a uma baixa distorção prosódica, devendo-se evitar juntar um *picth* crescente com um decrescente. Tal facto pode ser evitado através de técnicas de alteração de *pitch* e da duração das unidades, com o auxílio da adição de outras distorções.

- Em caso de discurso aleatório, irrestrito, devem-se usar unidades gerais (e.g. difones ou fonemas), para que o sistema adquira um carácter flexível.

- Como a aquisição de dados acústicos é limitada, as unidades devem ter um tamanho, que permita que todos os segmentos possíveis (para a unidade em questão), sejam contempladas numa base de dados. Como diz o termo inglês, todos os segmentos da base de dados devem ser "*trainables*".

2.5.2. Síntese por Concatenação de Fonemas

Os fonemas são a unidade de síntese mais pequena. A cada fonema está associado um som (que o representa), que é independente do contexto fonético de todos os fonemas vizinhos. Com o uso de fonemas é possível gerar qualquer palavra, pois a utilização desta unidade dá um carácter generalista à síntese. O número total de unidades (fonemas) é comparativamente menor do que qualquer outro tipo de unidades. No entanto, alguns fonemas não têm um estado estacionário, como por exemplo as consoantes oclusivas, facto que torna a síntese bastante complicada.

Com uma síntese por concatenação de fonemas, há necessariamente um grande número de pontos de articulação, zonas susceptíveis a descontinuidade espectral, o que pode provocar distorções no discurso sintetizado.

2.5.3. Síntese por Concatenação de Difones

Unidades como os difones preservam as transições entre fonemas, que seriam difíceis de ser produzidas, estendendo-se da parte central de um fonema (zona estacionária) para a parte central do fonema seguinte. Desta forma, os limites entre difones durante a síntese ocorrem no meio dos fonemas. Isto tende a resultar em descontinuidades de concatenação relativamente pequenas, porque o meio dos fonemas é, normalmente, a sua região espectral mais estável, sendo ainda, em geral, espectralmente consistente entre contextos fonéticos.

Outra vantagem do uso de difones deve-se ao facto de estes não necessitarem de regras para efeitos de co-articulação.

Geralmente, o número de difones é igual ao quadrado do número de fonemas de uma linguagem, se bem, que na prática, o número de difones necessário seja bem menor, pois nem todas as combinações de fonemas são possíveis dentro de uma palavra.

Os difones são unidades de pequena dimensão, não necessitando de grandes requisitos de memória.

Apesar dos difones conterem a transição entre fonemas, pode haver distorções devido à diferença de espectros entre as partes estacionárias das duas unidades (difones), caso estas estejam em diferentes contextos. Por esta razão, muitos dos sistemas que usam síntese por difones não são puramente baseados em difones, pois casos em que se pretende "ligar" duas fricativas, ou fricativas com paragens, não são contemplados. Nestes casos, os sistemas fornecem unidades mais longas, com maior qualidade de co-articulação. A prosódia pode ser imposta nos difones através de algoritmos que modelam a fala por meio de técnicas de processamento digital de sinal, como por exemplo, o PSOLA, um método usado para atenuar as descontinuidades de entoação entre dois difones, apresentado mais à frente.

2.5.4. Síntese por Concatenação de Sílabas

O uso de sílabas como unidade de concatenação implica bases de dados muito grandes.

A descontinuidade que existe entre sílabas é muito mais notada do que as descontinuidades dentro destas, logo as sílabas apresentam-se como uma unidade natural para a síntese, em relação a unidades mais pequenas.

Deve existir mais do que um "segmento de dicção", por sílaba, para que diferentes contextos acústicos e padrões prosódicos sejam garantidos, sobretudo se não se usar algoritmos de modificação da forma de onda, para cada sílaba.

2.5.5. Síntese por Concatenação de Palavras

O uso de palavras como unidades de concatenação aumenta a naturalidade dos sistemas, mas estes perdem por sua vez o carácter generalista e flexível, característico de sistemas que usam unidades mais pequenas.

Concatenar palavras é relativamente fácil, comparando com unidades de síntese menores, porque a co-articulação entre palavras é normalmente mais fraca do que dentro delas.

A simples concatenação de formas de onda que representam palavras é muito difícil de estender. A naturalidade das palavras produzidas isoladamente é diferente de um discurso formado pela combinação de algumas destas. Isto deve-se, principalmente, às descontinuidades do *pitch* e dos formantes nos limites das palavras, o que implica que estas transições sejam processadas. Para além disso, as palavras pronunciadas isoladamente têm maior duração do que as palavras em contexto de frase. A realização acústica e fonética das palavras varia também de acordo com o contexto.

Para se alcançar níveis razoáveis de naturalidade, é necessário registar várias versões de cada palavra, pronunciada em contextos diferentes ou usar processamento para a modificação do *pitch* e das durações, para a resolução dos problemas de descontinuidades dos formantes entre palavras, no sentido de se produzir um discurso natural.

Se o vocabulário a cobrir pelo sintetizador não for restrito, tendo em conta as palavras existentes numa língua, acrescentando a estas, as suas variações de acordo com um contexto de prosódia diferente, as bases de dados a serem gravadas podem tomar proporções dimensões gigantescas. No caso de domínios restritos sabendo á partida as palavras a concatenar, as bases de dados são possíveis de realizar e o discurso produzido é natural.

Outra vantagem do uso de palavras, deve-se ao facto de estas não dependerem foneticamente das transcrições de um dicionário. Em sistemas de concatenação de unidades menores, é possível que uma *string* de fonemas, que se associe a uma palavra através de um dicionário, não seja a correcta ou suficientemente fluente, problema que não acontece num sistema de concatenação de palavras.

2.5.6. Síntese PSOLA

Um algoritmo muito utilizado na concatenação de unidades é o PSOLA [1], desenvolvido em França, em 1986, pela *France Telecom*. Uma das características do algoritmo PSOLA consiste na manipulação síncrona dos períodos de *pitch* do sinal. Por isso, a qualidade do sinal gerado é intimamente dependente da existência de um algoritmo de marcação de *pitch* eficiente.

O sinal de voz a ser processado deve ser submetido a um algoritmo de marcação de *pitch*. As marcas são posicionadas nos picos do sinal, nas porções sonoras, e espaçadas de um valor fixo, nas porções não sonoras.

O PSOLA suaviza os segmentos concatenados, permitindo ainda que o *pitch* e a duração dos segmentos seja alterada. A versão mais simples do algoritmo é conhecida como TD-PSOLA [10]. A sua grande popularidade provém do facto de se tratar de um algoritmo extremamente simples e de custo computacional bastante baixo, capaz de realizar o processo de síntese praticamente em tempo real, gerando um sinal de alta qualidade. A razão principal por de trás da sua simplicidade computacional reside no facto de o algoritmo não exigir qualquer tipo de análise espectral do sinal, trabalhando directamente sobre a forma de onda, no domínio do tempo.

A primeira etapa do algoritmo TD-PSOLA consiste em dividir o sinal a ser modificado (chamado sinal de análise) numa sequência de sinais menores, denominados sinais elementares. Esta divisão é feita de modo a que a soma dos sinais elementares corresponda ao sinal original. Para alcançar esse resultado submete-se o sinal original a uma sequência de janelas, onde a frequência de análise é síncrona com o período de *pitch* do sinal [19]. Note-se que as janelas adjacentes deverão ser sobrepostas como mostra a figura a baixo.

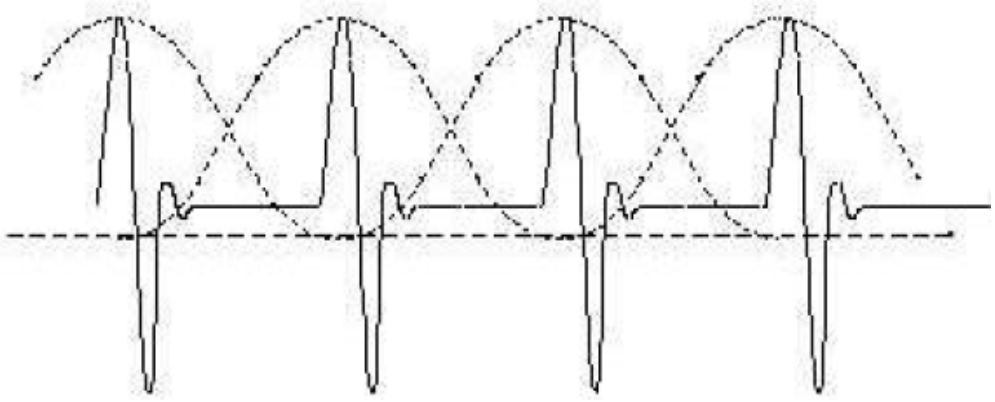

Figura 5 – Janelas aplicadas a um sinal

Para a geração de uma nova sequência de sinais (sequência de síntese) os sinais elementares são manipulados, de forma a alterar os parâmetros prosódicos do sinal

original. Há dois tipos de manipulações que podem ser efectuadas no sinal original, a alteração da duração e da frequência fundamental.

O procedimento básico para alterar a duração do sinal consiste em retirar ou repetir alguns dos seus sinais elementares. Ao retirar é provocada a diminuição da duração e ao adicionar é aumentada a duração do sinal. Em ambos os casos, o número de sinais elementares retirados ou adicionados determina a nova duração do sinal. O processo de alteração da duração pode ser feito tanto com as marcas sonoras como também com as marcas não sonoras.

Para alterarmos a frequência fundamental do sinal original devemos modificar o intervalo de tempo entre 2 sinais elementares. Ao aumentarmos o intervalo de tempo, diminuímos a frequência, e vice-versa. Ao contrário do que ocorre no caso da alteração da duração, a alteração da frequência é feita apenas com as marcas sonoras do sinal de análise.

A última etapa do algoritmo TD-PSOLA consiste simplesmente em somar os sinais elementares que irão compor a sequência de síntese.

Apesar de todas as vantagens, o algoritmo TD-PSOLA apresenta também algumas limitações.

Uma das limitações deve-se ao facto deste algoritmo não permitir a multiplicação, à duração do sinal original, de qualquer valor para a sua expansão ou compressão. O valor global da duração tem que ser múltiplo do período de *pitch*. Por sua vez, se se tentar efectuar um aumento de duração em segmentos não sonoros (surdos), há a adição de ruído, percepcionada pela audição de um som metálico. Isto acontece, pois, ao replicar um sinal não sonoro, estamos a dar-lhe um carácter periódico (sendo que os sinais não sonoros são aperiódico) [8]. Outro problema dá-se sempre que se pretende efectuar uma alteração na frequência fundamental do sinal, como consequência o espaçamento entre as janelas é modificado e, com isso, a duração do sinal também é afectada de maneira indesejada. Portanto, modificações da frequência fundamental devem vir sempre acompanhadas de correcções do factor de duração, a fim de compensar essa distorção [19].

Por fim, é importante referir os problemas deste algoritmo relativamente às descontinuidades nas zonas de junção dos segmentos. Existem 3 tipos de descontinuidades, de fase, de *pitch* e na envolvente espectral [15].

A descontinuidade de fase ocorre, porque as janelas de análise nem sempre estão posicionadas no mesmo ponto em relação ao período no sinal. A descontinuidade de *pitch* ocorre quando o *pitch* final do primeiro segmento é diferente do *pitch* inicial do segmento seguinte. Por fim, a descontinuidade na envolvente espectral ocorre, pois o TD-PSOLA não possui nenhuma estratégia de manipulação espectral dos segmentos. Assim, se os espectros não forem idênticos, há descontinuidade.

2.5.7. Síntese MBR-PSOLA

No sentido de eliminar algumas das limitações do método TD-PSOLA, surgiram novos algoritmos, com especial destaque para o MBR-PSOLA, muito simples e de baixo processamento computacional. O algoritmo MBR-PSOLA [15], tem como objectivo solucionar os problemas de descontinuidade de fase, saltos de *pith* e a descontinuidade da envolvente espectral.

A ideia por trás do MBR-PSOLA é a de efectuar um processo de análise/re-síntese em todas as *frames* pertencentes aos segmentos da base de dados. Baseado no modelo MBE (*multi-band excited*), esse processo procura efectuar uma normalização, fazendo ajuste de fase e atribuindo um valor de F0 constante para todos os segmentos da base de dados. O processo de análise/re-síntese causa pouca degradação no sinal original e elimina os problemas de descontinuidade de *pitch* e de fase que costumam afectar o resultado da concatenação dos segmentos. Além disso, a re-síntese permite lidar com o problema da descontinuidade espectral de maneira simples, pois ela faz com que a interpolação da envolvente espectral equivalha a uma interpolação simples no domínio do tempo (obedecidas as condições de igualdade de fase e de F0). Portanto, ao introduzir um algoritmo de interpolação temporal que actue nos períodos próximos à região de junção, consegue-se uma suavização espectral, impossível de se obter pela técnica TD-PSOLA tradicional. Trata-se de uma operação simples realizada sempre no domínio do tempo, evitando-se a adição de uma complexidade extra, devida à manipulação directa do espectro do sinal [16].

O custo computacional introduzido é pequeno, uma vez que o processo de análise/re-síntese é realizado uma única vez para todos os segmentos sonoros [8].

Este algoritmo, no entanto apresenta uma desvantagem, pois ao alterar as fases para uma fase fixa é introduzido ruído.

2.6. Síntese em Domínio Restrito

Na síntese em domínios restritos, usam-se muitas vezes palavras e frases pré-gravadas para criar a completa articulação do discurso.

Este tipo de síntese em sistemas TTS só faz sentido para sínteses em domínios específicos (restritos), sendo que, para estes casos, a qualidade do discurso sintetizado é bastante boa.

A naturalidade destes sistemas é muito alta, porque o vocabulário que se pretende reproduzir é limitado, sendo o discurso produzido muito próximo da prosódia e entoação das gravações originais. Estes sistemas só podem sintetizar combinações de frases e palavras que estão dentro de um contexto à partida programado [1].

A mistura das palavras pode causar problemas na produção de um discurso natural, a não ser que muitas variações das mesmas sejam contempladas ou que se usem métodos que permitam o "alisamento" do discurso. Deste modo, não se pode fazer uma simples concatenação de palavras, são necessários processos adicionais para tratar a complexidade de contextos [29].

2.7. Síntese por Selecção de Unidades

Existem sistemas que usam a combinação de unidades, tendo as suas bases de dados um conjunto de palavras e frases mais frequentes e também um conjunto de unidades mais pequenas, de modo, a dar ao sistema um carácter generalista, capaz de produzir qualquer trecho de discurso.

Uma síntese por selecção de unidades usa bases de dados de grandes dimensões. Durante a criação destas bases de dados, os segmentos gravados são segmentados a vários níveis (fonemas, difones, sílabas, palavras, etc) [19].

A selecção de unidades permite uma maior naturalidade, pois aplica pequenas quantidades de processamento de sinal à fala gravada. As técnicas de processamento de sinal digital produzem geralmente discursos gravados de menor naturalidade, embora se use em alguns sistemas em pontos de concatenação para minimizar os efeitos de distorção (como exemplo, o PSOLA) [28].

Nos sistemas em que se usa este tipo de concatenação as saídas são, muitas vezes, indistinguíveis das vozes humanas, especialmente quando a síntese é feita em contextos para os quais o TTS foi sintonizado.

2.8. Síntese de Nomes Próprios e sua Problemática

A construção de um sistema para síntese de nomes próprios com uma pronúncia inteligível está longe de ser um luxo para sistemas TTS. A maioria destes sistemas tem aplicações alvo como a reprodução de nomes de pessoas, endereços, cidades, etc. No entanto, este tipo de síntese tem problemas específicos e necessita de uma atenção especial.

Os nomes podem ter diversas origens etimológicas. O nome é uma palavra que pode viajar de língua para língua, mostrando diferentes graus de ajustamento à língua de chegada, podendo estabelecer-se, por vezes, em línguas com regras de conversão grafema-fonema diferentes da sua língua de origem, o que traz enormes problemas na criação da pronúncia. Como consequência, muitos nomes estrangeiros não podem ser identificados pela sua ortografia, pois estão em conformidade com a língua nativa [12]. Pode também acontecer, em último caso, que um nome seja "introduzido" numa língua sem sofrer os seus processos linguísticos. A etimologia do léxico do nome não é de conhecimento geral, a tarefa de "adivinhação" é deixada ao transcritor. Por vezes, mesmo havendo uma boa "adivinhação", outros factores se colocam, como o conhecimento das regras de pronúncia da língua estrangeira, o conhecimento do verdadeiro local da pronúncia e a capacidade de pronúncia da língua estrangeira [11].

A conversão grafema-fonema para nomes é normalmente pior do que a conversão geral, do léxico comum, pois, na maioria dos idiomas, os nomes são processados com regras diferentes das palavras vulgares.

No que diz respeito à síntese a partir de textos arbitrários, é normal haver um grupo de fonemas de grande frequência, capaz de fazer a cobertura da maioria das palavras, o que não se verifica na síntese de nomes próprios. Esta síntese necessita de bases de dados maiores do que a síntese de palavras vulgares. Há um número enorme de nomes, havendo, por vezes, a necessidade de criar uma lista exaustiva destes, para se obter uma boa cobertura [12].

Capítulo 3 - Criação de Vozes para o Festival

3.1. O Festival

O Festival é um sistema usado para desenvolver processos relacionados com a síntese de fala. A sua utilização permite a realização de síntese de fala de alta qualidade a partir de textos arbitrários, de forma rápida e eficiente, o desenvolvimento de sistemas de linguagem, que tenham como saída síntese de voz, e também o desenvolvimento e teste de novos métodos e técnicas de síntese de voz [23].

O sistema é compatível com o Windows e com o Linux, sendo bastante popular devido ao seu carácter aberto, com um acesso e utilização gratuita, permitindo a alteração do código fonte, podendo este ser personalizado de acordo com as necessidades do utilizador, desde que se respeitem algumas regras de utilização pré-definidas.

Na sua estrutura interna, conta também com um suporte adequado a sistemas multi-linguísticos e à realização de sínteses por concatenação, com a disponibilização de scripts e manuais detalhados [2].

Este sistema tem a arquitectura de um sistema TTS em geral, incorporando todos os seus blocos.

A seguinte imagem mostra a arquitectura do Festival.

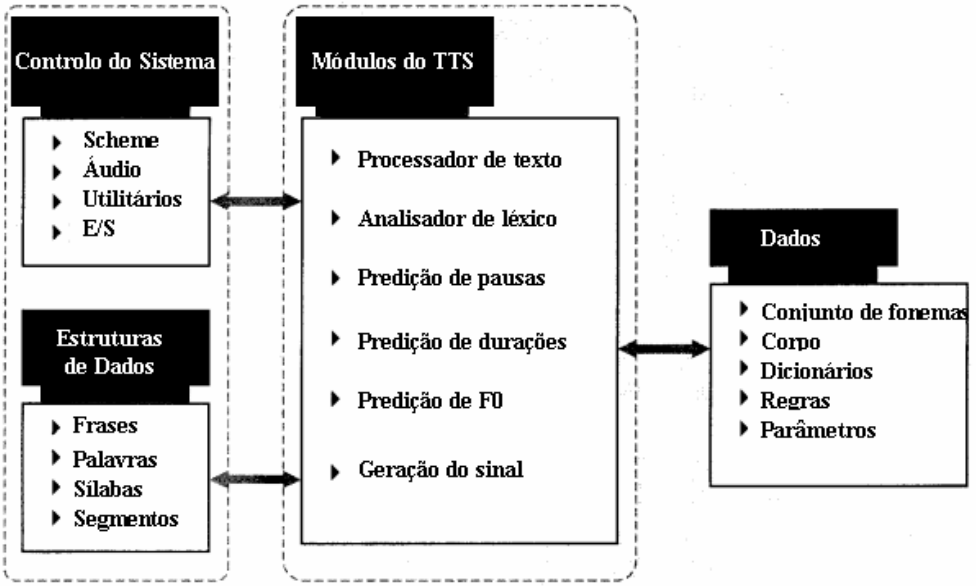

Figura 6 – Estrutura do Festival

Estes blocos são susceptíveis a mudanças, consoante as necessidades do sistema a implementar, sendo assim possível desenvolver novas aplicações e mesmo elaborar testes para a criação de novas técnicas de síntese de voz.

Todo o controlo do Festival é feito por um conjunto de programas em C++ e C, programas de baixo nível, mas de execução rápida e capazes de monitorizar todas as funções de entrada e saída através de linha de comandos. Esta interacção com o utilizador, por via de linha de comandos, permite ao Festival agir com uma "caixa negra" à qual se fornece dados de entrada para se receber um sinal de saída. Por exemplo, se se introduzir o comando (SayText "Pedro Manuel Amorim") o sistema produz um sinal sonoro que reproduz o nome ([32] e [23]).

A seguinte imagem mostra alguns comandos Festival executados através do Cygwin.

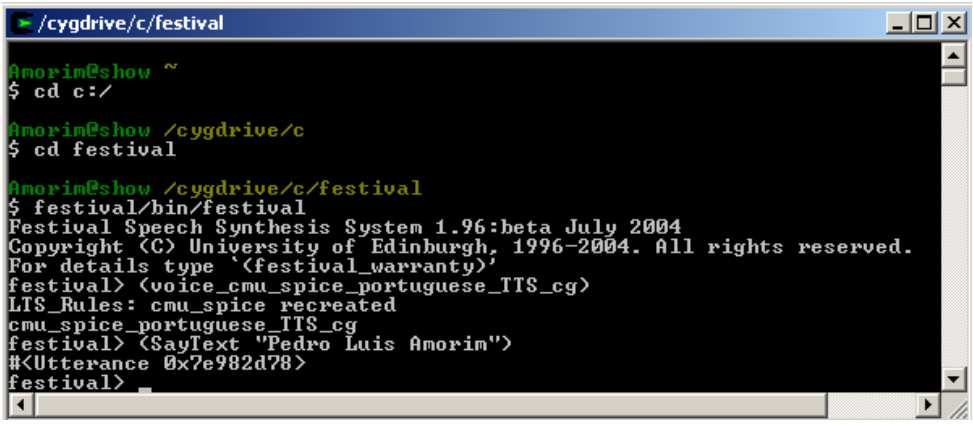

Figura 7 – Festival executado a partir do Cygwin

No processo de conversão do texto em fala, é necessária a criação de uma estrutura de dados que permita a formação de uma *utterance* (pronunciação). A *utterance* representa, assim, a quantidade de texto que é processado como discurso. Em geral, o processo de conversão do texto em fala consiste em tornar a *utterance*, inicialmente uma sequência de caracteres, num conjunto de informações, adquiridas passo a passo ao longo dos vários módulos constituintes de um sistema TTS, até que se gere uma onda sonora capaz de reproduzir a sequência de caracteres inicialmente adquirida [32].

O Festival disponibiliza todo um conjunto de módulos de um sistema TTS [7], contendo na sua estrutura um processador de texto, um analisador de léxico, modelos para a predição de pausas entre frases, modelos para a predição de durações, entoação e frequência fundamental e também métodos para a geração de sinais sonoros, baseados em concatenação de difones. Todos estes módulos serão descritos de forma mais pormenorizada mais à frente.

O desenvolvimento de um sistema de síntese desde o início é um processo trabalhoso e demorado, que não faz sentido se pretender desenvolver ou testar uma pequena parte do sistema. Daqui surge uma vantagem do Festival em relação a outros programas, pois este permite aos utilizadores focarem-se apenas nas partes do sistema que pretendem desenvolver (com vista a melhorar a qualidade de todo o sistema), sem terem de construir um sistema de início, nem adaptar um sistema já existente, poupando-se tempo e trabalho. O Festival é um *software* projectado com a finalidade de permitir a adição de novos modelos, de forma fácil e eficiente, evitando-se tempo e trabalho necessário nos casos de reimplementações [23].

Este *software*, para além de fácil e eficaz em sistemas de síntese de voz, tem um ambiente de teste bastante vantajoso, que permite ao utilizador beneficiar directamente de todas as experiências e investigações desenvolvidas no programa, úteis para o desenvolvimento de sintetizadores. Permite também assegurar se um determinado projecto é ou não funcional, possibilitando a visualização dos problemas existentes e a sua localização.

Toda a base do Festival está baseada nas bibliotecas da *Edimburgh Speech Tools*, usando parte das suas potencialidades. Muitos dos desenvolvimentos destas bibliotecas são dirigidos às necessidades do Festival, de modo a torná-lo mais usável. Destaca-se ainda no sistema o projecto Festvox, que torna o sistema acessível à construção de novas vozes [2].

O Festival é um sistema em constante evolução e desenvolvimento. Este vem sendo aperfeiçoado ao longo do tempo. A nível da arquitectura poucas mudanças se perspectivam, estando a evolução dos próximos tempos virada para questões de prosódia, criação de dialectos independentes da necessidade de um léxico específico, implementação de outros tipos de síntese por concatenação, como a por selecção de unidades, introdução de novas técnicas de entoação, novos modelos para a síntese de ondas sonoras, etc. Com isto, espera-se que no futuro o sistema consiga elaborar sínteses cada vez mais naturais e inteligíveis [23].

3.2. Criação de uma Voz no Festival

Para a criação de uma voz no Festival é necessário a criação de um certo número de parâmetros. Estes parâmetros podem ser obtidos de diversas maneiras, usando as ferramentas e exemplos disponíveis no Festival e no FextVox, através de aplicações externas, como o SPICE, ou mesmo de forma manual.

Passo agora a enumerar todas etapas fundamentais à criação de uma voz [23]:

- A criação do conjunto de fonemas.
- A criação de um processador de texto
- A criação de um analisador de léxico.
- A criação de um conjunto de regras para a conversão grafema-fone.

- A criação de um modelo para as pausas entre frases.
- A criação de um modelo para a entoação.
- A criação de um modelo para as durações dos segmentos.
- A escolha de um sintetizador de ondas sonoras.

O Festival tem todo um conjunto de ferramentas disponíveis para a criação de uma voz, mas há sempre a possibilidade de estas serem alteradas e melhoradas com a adição de novas vozes, no sentido de se criarem vozes com uma melhor qualidade.

Passo agora a descrever todos os passos essenciais à criação de uma voz para o Festival, referenciando sempre os exemplos e ferramentas que o Festival tem disponível na sua estrutura.

3.2.1. Criação do Conjunto de Fonemas

A definição do conjunto de fonemas é a base para a criação de uma voz, pois muitos dos processos necessários à construção de vozes dependem directamente dos fonemas. A definição do léxico e das regras para a conversão grafema-fonema, a criação das formas de onda dos sintetizadores, assim como outras etapas, só são possíveis após esta definição.

Para a definição de todo o conjunto de fonemas, é necessário enumerar todos aqueles que fazem parte da língua a criar. Aos fonemas, enumerados um por linha, devem ser associados um conjunto de propriedades que os caracterizam, nomeadamente: se o fonema é vogal ou consoante; caso seja vogal, o seu comprimento (curta, longa, ditongo ou schwa), a altura (alta, media ou baixa), a zona de articulação (alta, média ou baixa) e se há arredondamento dos lábios; caso seja consoante, o seu tipo (oclusiva, fricativa, africada, nasal, lateral ou aproximaste), o ponto de articulação (labial, alveolar, palatal, labio-dental, dental, velar ou uvular) e, por fim, se a consoante é ou não vozeada.

É importante definir também um ou mais fonemas que representem os silêncios (por exemplo, para as respirações, para o silêncio inicial, etc).

Um exemplo de uma definição de um conjunto de fonemas pode se encontrado no Festival em "*lib/mrpa_phones.scm*". Informações mais detalhadas acerca da criação do conjunto de fonemas podem ser encontradas no manual do Festival [23].

3.2.2. Processador de Texto

O Festival tem disponível um processador de texto. O objectivo desta etapa consiste na normalização do texto de entrada. Assim, todos os caracteres especiais, numerais, abreviaturas, acrónimos são tratados de modo a serem símbolos válidos e pertencentes ao idioma em estudo, passando o texto a ser constituído por "*tokens*". Um exemplo de um conjunto de funções capazes desta normalização podem ser encontrados no Festival em "*lib/token.scm*". Mais informações acerca da criação do processador de texto podem ser encontradas no manual do Festival em [23].

3.2.3. Analisador de Léxico

O analisador do Festival é um subsistema que fornece a previsão da pronúncia das palavras. Este processo pode ser elaborado de três modos diferentes: através de um dicionário local, normalmente escrito à mão, um dicionário de léxico compilado, com um enorme volume de palavras ou através de um método capaz de predizer a pronúncia das palavras que não se encontrem em nenhum destes dicionários, geralmente através de regras.

Como exemplo de uma entrada (através do comando "*lex.add.entry*") da palavra "*present*", para um dicionário do Festival (exemplo retirado do manual do Festival [23]) temos:

(lex.add.entry'("present" nil (p r e) 0) ((z @ n t) 1))))

O primeiro campo ("*present*") contém a palavra que se pretende transcrever. O espaço "*nil*" é reservado para informações de discurso. Deve ser "*nil*" quando não há

especificações. Do lado direito, a palavra é enunciada como uma *string* de fonemas, a pronúncia propriamente dita.

Todos os formatos dos dicionários e de entradas de novas palavras, assim como outras especificações e informações, podem ser encontradas no manual do Festival [23].

3.2.4. Regras de Conversão Grafema-fone

Quando uma palavra não está contida no dicionário local, nem no dicionário léxico compilado, a opção tomada no Festival consiste em usar uma função que manipule um conjunto de regras capazes de garantir a transcrição grafema-fonema, obtendo-se assim, uma pronúncia para a palavra desconhecida.

Abaixo, é representado um exemplo de uma possível regra:

(a [s] e = z), esta regra significa que o caracter [s], com um contexto à esquerda [a] e um contexto à direita [e], é transcrito pelo fonema [z].

Exemplos de funções capazes de chamar e processar este tipo de regras, assim como exemplos das mesmas, podem ser encontrados no manual do Festival [23] e no Festival em "*lib/lts.scm*"e "*lib/lts_build.scm*".

3.2.5. Pausas entre Frases

O Festival tem disponíveis dois métodos para a predição de pausas entre as frases, um deles simples e outro mais sofisticado.

O primeiro método usa uma árvore CART. A árvore é aplicada a cada palavra de modo a decidir a inserção de pausas. Assim, fica-se a saber se após uma palavra, existe uma pausa, uma grande pausa ou nenhuma pausa. Esta previsão é feita com base nos sinais de pontuação.

Um exemplo desta árvore pode ser encontrado no Festival em "*lib/phrase.scm*".

O segundo método é baseado em modelos probabilísticos (baseados em modelos *n-gram*), onde a probabilidade de uma pausa depois de uma palavra, depende do contexto e das palavras anteriores.

Todas as especificações, exemplos e códigos destes modelos podem ser encontrados no manual do Festival [23].

3.2.6. Entoação

O Festival disponibiliza vários modelos de entoação com diferentes níveis de controlo.

Em geral, a entoação é gerada em dois passos: a predição do tipo de acento para cada uma das sílabas e a definição do contorno de F0 usando pontos de entoação. Como reflexo desta divisão, podemos destacar dois sub-módulos para a construção da entoação. Exemplos destes modelos estão disponíveis no Festival em *"lib/intonation.scm"*.

Existem vários modelos de entoação disponíveis no Festival tais como: o *Default intonation*, o *Simple intonation*, o *Tree intonation* e o *General intonation*. Passamos agora a descrever, de forma resumida, a ideia principal de cada um destes métodos.

O ***Default intonation*** é a maneira mais simples de se criar uma entoação. Neste modelo, não há a predição de acentos para as sílabas, apenas se atribui um ponto de entoação ao início e ao fim da pronúncia, para definir o contorno de F0.

O ***Simple intonation*** usa uma árvore CART para predizer a acentuação das sílabas. Se na árvore for devolvido o valor NENHUMA, significa que a sílaba não é acentuada, logo não é gerado nenhum acento, se outro valor for devolvido é colocado um acento na sílaba. Pode ser consultado um exemplo padrão desta árvore no Festival em "*lib/intonation.scm*".

Para a definição do contorno de *F0*, usa-se o seguinte raciocínio: em cada frase é gerada uma pronúncia, em que o valor de *F0* inicial é dado pela expressão: "*f0_code + (f0_std*0,6)*". O valor de *F0* decai f0*_std* Hz (valor de desvio padrão) ao longo do comprimento da frase até à última sílaba, tendo neste ponto o valor de "*f0_code – f0_std*". Assim, o contorno de *F0* é estabelecido por uma linha imaginária que vai do princípio ao

fim da frase, denominada de *baseline* (linha que liga o valor estabelecido para a primeira sílaba ao da última sílaba).

Por sua vez, para cada sílaba acentuada são gerados três pontos de entoação, um no início, um no meio e outro no fim. Os pontos de entoação do início e fim da sílaba tem o valor de *baseline* Hz, o ponto de entoação do meio da sílaba tem um valor *fo_std* Hz acima da *baseline*.

O ***tree intonation*** é um modelo mais flexível, podendo usar-se duas árvores CART diferentes para predizer a acentuação das sílabas. Em relação à determinação dos pontos de entoação, este método é mais sofisticado que o usado no *Simple intonation*, usando um modelo de regressão linear para predizer os pontos de entoação no início, fim e meio da sílaba acentuada. Estes modelos de regressão linear estão descritos no manual do Festival, na secção 25.5 (em festvox.org/docs/manual-1.4.3/festival_toc.html). Existem outros modelos como o *General intonation* com um maior grau de complexidade.

Mais informações e exemplos destes modelos podem ser consultados e abordados de forma mais pormenorizada no manual do Festival [23].

3.2.7. Duração dos Segmentos

O Festival suporta vários métodos de predição da duração de segmentos, normalmente fonemas.

Os métodos que o Festival tem disponível para este fim são: o *Default*, o *Average*, o *Klatt* e outros baseados em CARTs.

No ***método Default***, assume-se que todos os fonemas têm a mesma duração, usando-se assim valores fixos. Como padrão no Festival é usado o valor de 100 milissegundos. Este valor pode ser mudado com a alteração do valor do parâmetro "*Duration_strerch*".

No ***método Average***, a duração dos segmentos é baseada nas durações médias dos seus fonemas. Assim a variável "*phoneme_durations*" deve ser uma lista de fonemas com as suas durações médias associadas, em segundos. Estas médias são calculadas a partir um qualquer corpus de voz. Se for encontrado um segmento não representado na lista, é assumida uma duração de 0,1 segundos e gera-se uma mensagem de erro.

O ***método Klatt*** assume que a cada fonema está associado uma duração inerente e uma duração mínima. Esta lista de valores de durações esta definida numa variável "*duration_klatt_params*".

Por fim, surgem os métodos mais sofisticados, que usam árvores de decisão. Um dos métodos é relativamente simples: a árvore prediz a duração de cada segmento directamente. O segundo método, que parece dar melhores resultados, prediz um factor a ser multiplicado pela duração média dos segmentos, em vez de predizer a duração dos segmentos directamente.

Exemplos de todos estes métodos, assim como códigos e informações adicionais, associadas a cada um destes métodos podem ser consultados no manual do Festival [23] e nos exemplos disponíveis no directório "*lib*" do Festival.

3.2.8. O Sintetizador Propriamente Dito

O Festival suporta o uso de diversos sintetizadores baseados na concatenação de difones. Para usar um destes sintetizadores, deverão ser especificados dois parâmetros: um método de síntese, que pode ser por exemplo, o Diphone, OGlresLPC ou também o MBROLA e a base de difones, que consiste na colecção de arquivos áudio onde são gravados os difones e tem associada uma lista que estabelece a correspondência dos difones do corpus com a sua posição dentro do arquivo áudio. A criação destas bases de difones é um processo bastante complexo.

Mais informações sobre a escolha dos sintetizadores podem ser consultadas em [23].

3.3. MBROLA

Como já referido anteriormente, o Festival suporta o uso de diversos sintetizadores baseados na concatenação de difones. Neste ponto, será abordado o projecto MBROLA, assim como, os principais passos para a criação de uma base de dados de difones.

O MBROLA é um *software* desenvolvido num projecto [22] do laboratório TCTS da *Faculté Polytechnique de Mons*, na Bélgica. Com este projecto pretende-se o desenvolvimento de sintetizadores de voz, para um número crescente de línguas. Este sistema está disponível de forma gratuita para aplicações não comerciais. Actualmente, o sistema disponibiliza um total de 26 bases de dados de difones, incluindo o português europeu.

O sintetizador MBROLA faz a concatenação de difones, através da técnica MBR-PSOLA, já descrita no capítulo 2.

É importante referir que o sintetizador MBROLA não é um sistema TTS completo, mas sim um gerador de ondas sonoras, recebendo como entrada uma sequência de fonemas e informações acerca das suas durações e da descrição dos contornos de F0 [8].

O MBROLA está suportado no Festival, no ficheiro "*lib/mbrla.scm*". A função MBROLA_Synth, disponível neste ficheiro, é chamada sempre que se pretende usar a síntese MBROLA. Esta função guarda os segmentos necessários à síntese (com todas as informações de prosódia) e chama o programa externo, MBROLA (com a sua base de dados de difones já seleccionada), para a geração de ondas sonoras. Após a criação destas ondas, estas são carregadas de volta para o Festival, através desta mesma função [23].

3.3.1. Base de Dados de Difones para o MBROLA

Para se iniciar uma síntese com o MBROLA, primeiro é necessário a inserção de uma base de dados de difones no sistema. Esta base de dados pode ser adquirida de duas maneiras: através do seu *download*, disponível na página do projecto MBROLA [22], ou através da sua criação.

Na página, está disponível um conjunto de bases de dados de difones num grande número de línguas, incluindo o português.

Caso se opte pela criação de uma base de dados partindo do zero, é necessário ter em conta que o processo é moroso e trabalhoso.

Para a criação de uma base de dados partindo do zero são necessários 3 passos fundamentais: a criação de um corpus de texto, a gravação desse corpus e a segmentação do corpus falado.

Usando as informações do site do projecto [22], temos:

Criação de um corpus de texto:

O primeiro passo para a criação da base de dados é definir o conjunto de fones de uma língua. É necessário contemplar todos os fones de fonemas que proporcionam alofonia. Após ter uma lista com todos os fones contemplados, é gerada uma lista de difones e faz-se a recolha de uma lista de palavras, cuidadosamente escolhida de modo a que cada difone seja contemplado pelo menos uma vez. É necessário ter em conta que posições desfavoráveis, como no interior de sílabas tónicas e em zonas mais co-articuladas, devem ser excluídas. Deve-se ter em conta que existem difones que só aparecem por associação de palavras e, que grande número de difones nem sequer existe na prática. Conclui-se assim, que é difícil a criação de um corpus de texto que contemple todos os casos possíveis.

Gravação do corpus:

O corpus deve ser gravado por um informante profissional (se possível), de forma digital, sendo os dados gerados também armazenado em formato digital (formato recomendado Fs= 16Khz, 16_bit, Mono). De modo a obter melhores resultados, o corpus deve ser lido com uma entoação o mais monótona possível. Até final da leitura das palavras, a frequência fundamental deve ser mantida constante, o mais possível. O orador deve treinar antes de começar a sessão de gravação.

Para que as gravações sejam geradas da melhor maneira, é necessário o uso de equipamento de alta qualidade (microfones, amplificadores, conversor A/D, etc).

De modo a evitar ruídos, é necessário verificar a qualidade da placa de som. Outros ruídos a evitar provêm do próprio ambiente da gravação, do efeito de reverberação e de ruídos de baixas frequências, deste modo é aconselhável o uso de um estúdio de gravação blindado ao som.

Segmentação do corpus:

Após a segmentação do corpus é esperado que aos arquivos de áudio onde se gravaram os difones tenham anexado uma de lista que associa os difonemas do corpus com a sua posição dentro dos arquivos de áudio.

Depois da gravação do corpus, todos os difonemas devem ser localizados. Este processo pode ser feito de forma manual, com a ajuda de ferramentas de visualização de sinal ou com algoritmos de segmentação, onde as decisões são verificadas e corrigidas de forma interactiva. Um exemplo de um *software* de segmentação é o Diphone Studio, referenciado na página do projecto MBROLA [22]. Com o uso deste *software*, poupa-se tempo na árdua tarefa de segmentação.

A base de dados é então criada, sendo cada resultado apresentado como um conjunto de parâmetros onde a cada difone está associado o seu nome, as formas de onda, a sua duração e as subdivisões internas.

É conveniente armazenar a fronteira entre os fones, pois permite a modificação da duração de meio fone sem afectar o comprimento do outro. Para optimização dos resultados com o uso do MBROLA, deve-se manter o contexto à direita e à esquerda de cada difone (em geral 50ms), pois a síntese MBROLA inclui algumas análises de entoação, proporcionando-se assim resultados mais precisos.

3.4. SPICE

Actualmente, com o desenvolvimento tecnológico e informático no processamento de voz, há uma tendência para quebrar as lacunas derivadas das diferenças de língua. Apesar de tudo, este é um processo bastante custoso, uma vez que o número de línguas em todo o mundo ascende a 6900. Houve assim, a necessidade de criar uma alternativa para o desenvolvimento do processamento destas línguas. Surgiu assim o SPICE [24], um *software* gratuito de fácil manuseamento ao alcance de qualquer pessoa especialista ou não em processamento de fala [25].

O SPICE é um sistema dinâmico. Foi criado de maneira a aproximar os peritos de engenharia dos peritos de linguística, partilhando-se assim, os conhecimentos de duas áreas distintas, no sentido da evolução do processamento de fala.

Esta ferramenta permite a construção de reconhecedores de voz, de síntese TTS e de modelos para tradução automática. É um sistema inteligente que está em constante evolução, desenvolvendo-se com o trabalho de cada utilizador. Cada utilizador é obrigado a criar uma conta para iniciar um projecto (numa determinada língua). Com a criação destas contas é possível iniciar ou continuar todos os projectos, já iniciados pelos utilizadores [17]. A aprendizagem do sistema é assim feita de forma iterativa com o desenvolvimento dos projectos de cada utilizador, aperfeiçoando-se as ferramentas do sistema (SPICE), com os projectos desenvolvidos. Há a possibilidade de fazer o *upload* de modelos já criados anteriormente, poupando-se tempo na execução de tarefas ou até mesmo a omissão do desenvolvimento de algumas destas.

Após este breve resumo do que é o SPICE, explica-se, em seguida, o conjunto de tarefas necessárias para a construção de um sistema TTS.

3.4.1. Passos Principais

No ambiente gráfico da ferramenta, destaca-se uma coluna do lado esquerdo com nove tarefas distintas, que fazem parte dos procedimentos essenciais à criação de uma voz, ou de um reconhecedor. O SPICE tem conhecimento sobre todas estas tarefas, só permitindo a activação destas, apenas quando todos os pré-requisitos necessários a cada uma estiverem cumpridos. Todas estas componentes têm um *help* associado e estão activas quando se encontram a verde [17].

A seguinte figura mostra o ambiente gráfico base do SPICE.

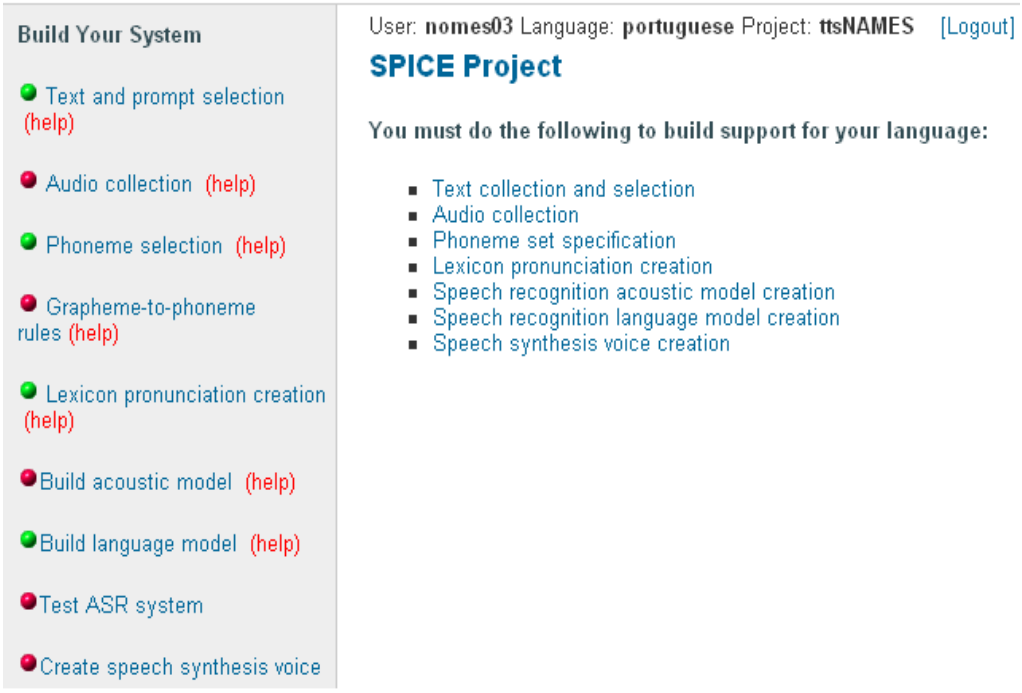

Figura 8 – Ambiente gráfico do SPICE

Após a descrição gráfica do SPICE, passa-se a explicar todas as etapas do sistema, necessárias para a criação de uma voz. O esquema da figura abaixo representa a interdependência das várias etapas (*text collection, prompt extration, speech collection, phoneme selectionio and dictionary building*) para sistemas ASR, TTS e de tradução [13].

Figura 9 – Interacção das várias etapas, para a criação de sistemas TTS, ASR e de tradução.

Passa-se agora à descrição de todos os passos para a criação de uma síntese no SPICE.

3.4.2. *Text and Prompt Selection*

O primeiro passo para o desenvolvimento da síntese de voz é a aquisição de dados de texto.

O volume de dados de entrada deve ser grande, mas não exageradamente grande, pois pretende-se um sistema com capacidade de aprendizagem. O texto adquirido deve estar de acordo com as nossas pretensões, ou seja se se pretende fazer uma síntese de nomes, o meu texto de entrada deverá ser nomes.

Esta aquisição de dados é importante para a obtenção de *prompts* e para a criação do *vocabulary*, indispensável no *audio collection*, essencial na síntese de voz.

No SPICE, há duas formas de obter dados de texto, por uma busca *web* ou por *upload* de um ficheiro de texto. Todas estas informações estão disponíveis nos *helps* do SPICE [24].

Para fazer o *crawl* de um texto de uma página *web*, é necessário especificar a página "mãe" e a profundidade dos links a aceder a partir desta. Após estas especificações, as ferramentas do SPICE acedem a todas estas páginas, seleccionando todos os textos relevantes e colocando-os no formato adequado. Não é aconselhado o acesso a uma profundidade elevada, pois implica muito tempo para a geração do texto na forma pretendida.

Outra forma de obter texto é através do *upload* de um ficheiro. Espera-se que este esteja na codificação apropriada (UTF-8) [13] e que em cada linha esteja uma única frase, tendo-se assim tantas frases como linhas.

No *text and prompt selection*, após a aquisição de texto, pode-se obter a informação sobre a frequência de cada palavra e de todos os caracteres, com a geração dos ficheiros, "*text/data.wc*"e "*text/data.lc"*.

Ainda no *text and prompt selection*, há a parte da geração de *prompts*. Estes podem ser gerados automaticamente pelo SPICE ou pelo *upload* de um ficheiro de *prompts*.

O SPICE disponibiliza um processo automático para a geração *prompts.* Sempre que se opte por esta opção, é processado um algoritmo que se encarrega de recolher as frases com as palavras de maior frequência, de modo a criar um léxico para a língua. Como padrão, o sistema usa as 5000 palavras mais frequentes. Após esta análise, o sistema escolhe todas as frases que lhe são "simpáticas", seleccionando aquelas que têm comprimento razoável, que contenham apenas palavras do *frequency lexicon*, que não tenham pontuação estranha, entre outros factores heurísticos, como descrito nos *helps* do

SPICE [24]. Um dos objectivos da geração dos *prompts* é obter a maior cobertura fonética possível para uma língua. Como esta geração se baseia em processos automáticos, há sempre a probabilidade de alguns dos *prompts* gerados não reunirem as características essenciais para serem bons *prompts*.

É importante ter em conta que os *prompts* devem ser foneticamente e prosodicamente equilibrados, orientados para o domínio em questão (para que o estilo de discurso seja igual à aplicação a desenvolver), fáceis de pronunciar e não propício a erros de leitura. Este conjunto de *prompts* não deve ser exagerado, para que um orador os consiga reproduzir sem perdas na precisão da pronúncia [24].

A outra forma de se obter *promts* é através do *upload* de um ficheiro. Este ficheiro deve estar codificado no formato UTF-8 [17]. O ficheiro deve apenas conter *prompts* segundo o formato que se segue no exemplo abaixo.

Pedro Amorim: (data_00001 "Pedro Amorim")

A seguinte imagem mostra o ambiente gráfico do *text and prompt selection.*

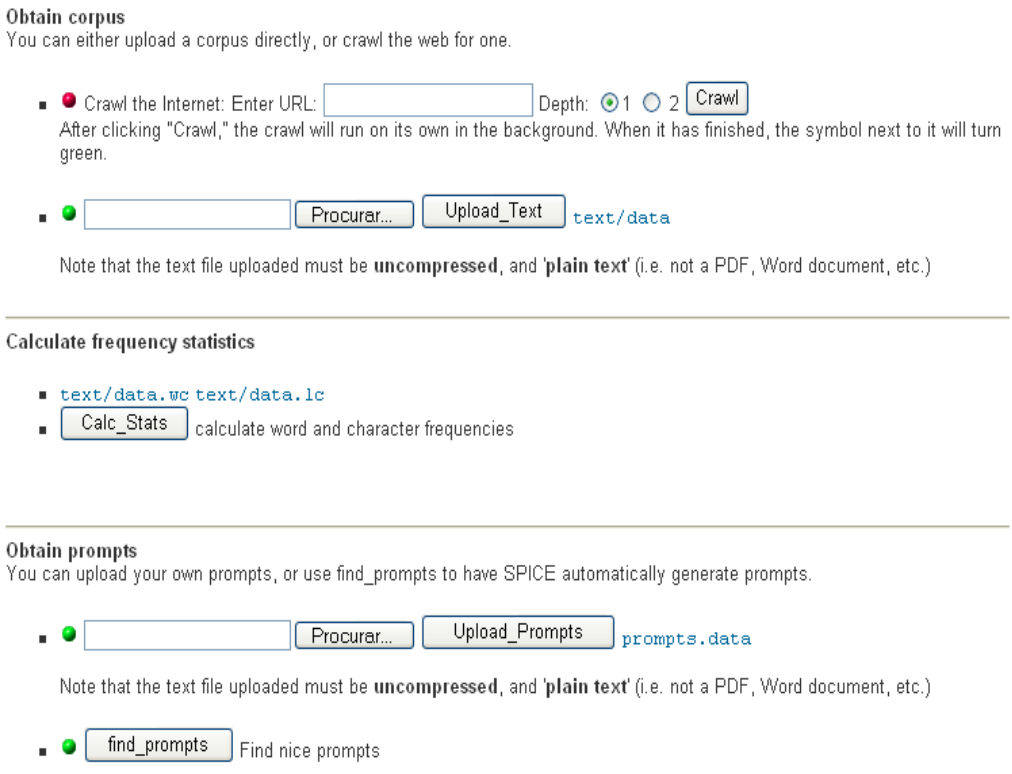

Figura 10 – Ambiente gráfico do *text and prompt selection*

3.4.3. *Phoneme Selection*

Nesta parte do sistema, são definidos os sons da língua (fonemas), essenciais para a construção da estrutura de uma língua.

Todas as línguas têm o seu conjunto de fonemas característico. A definição do "mapa" de fonemas é obrigatória na síntese de voz, pois é fundamental na construção de modelos de pronúncia e modelos acústicos [24].

A ferramenta *phoneme sellection* oferece duas possibilidades para a definição do conjunto de fonemas, o *upload* de um ficheiro com o nome "*settingFile*" pré-existente (descrito no exemplo abaixo) e o preenchimento online do mapa de fonemas, disponível no ambiente do SPICE [24].

Quando se faz *upload* de um ficheiro "*settingFile*", é esperado que cada linha do ficheiro tenha um fonema e a respectiva descrição.

Na tabela seguinte é apresentado o exemplo do conteúdo de um ficheiro"*settingsFile*".

PloBila1 p	FriLab2 v	OpenMidFront2 EE
PloBila2 b	FriAlv1 s	OpenMidBack2 OO
PloDAP1 t	FriAlv2 z	OpenHalfMidCent AX
PloDAP2 d	FriPos1 SS	OpenFront1 a
PloVel1 k	FriPos2 ZZ	ai AJ
PloUvu2 g	LatAppDAP l	au2 AW
NasBila m	LatAppPal LL	ei EJ
NasDAP n	CloFront2 i	eu EW
NasPal JJ	CloCent1 IX	iu IW
TriUvu RR	CloBack2 u	oi OJ
TapDAP r	CloMidFront2 e	ou OW
FriLab1 f	CloMidBack2 o	oci OOJ

Tabela 1 – Conteúdo de um ficheiro "*settingsFile*"

No caso das consoantes, a cada símbolo representativo do fonema está associada uma referência ao ponto de articulação (bilabial, lábio-dental, dental, alveolar, pré-palatal, palatal, velar ou uvular) e ao modo de articulação (oclusivas, nasais, fricativas, laterais ou vibrantes).

No caso das vogais, a cada símbolo representativo do fonema está associado uma referência ao grau de abertura (fechadas, semi-fechadas, semi-abertas ou abertas) e à zona de articulação (anterior, média ou posterior).

No caso dos ditongos, à representação escolhida para cada ditongo (fonema) está associada uma representação segundo o IPA.

Após o *upload* deste ficheiro, o SPICE cria um outro automaticamente, o "*PhoneMapFile*". O ficheiro tem um conteúdo semelhante ao exemplo representado na tabela abaixo, onde, em cada linha, há uma correspondência de um fonema (new_phoneme) com um elemento do *GloblaPhone phone set*.

New_Phoneme	***GP_Phoneme***
d	M_d
k	M_k
g	M_g
m	M_m
n	M_n
J	M_nj

Tabela 2 – Exemplo do formato de um possível ficheiro "*PhoneMapFile*".

Caso se opte pelo preenchimento online, o SPICE, no seu ambiente gráfico, disponibiliza um conjunto de 4 tabelas, uma para as consoantes mais comuns, outra para as menos comuns, uma para vogais e uma quarta para ditongos. Cada tabela, tem um conjunto de fones representados segundo o IPA, com uma caixa associada, para o preenchimento do utilizador. É através do preenchimento destas caixas que o utilizador assume quais os fonemas que o seu sistema deve conter e escolhe uma representação para cada um deles.

Nestas tabelas, associado a cada fone, há também um ficheiro áudio, que permite a sua audição.

As seguintes imagens mostram o ambiente gráfico disponível no SPICE para a construção do mapa de fonemas.

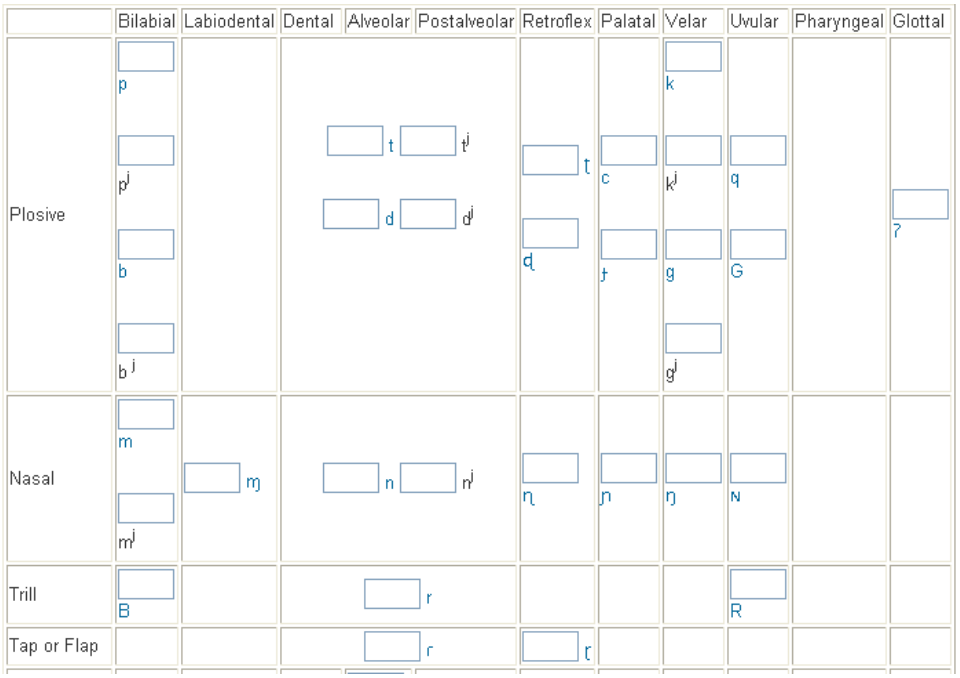

Figura 11 – Tabela das consoantes mais comuns

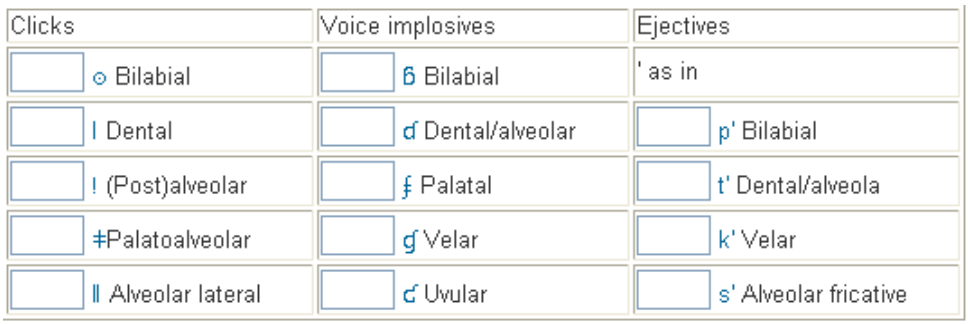

Figura 12 – Tabela das consoantes menos comuns

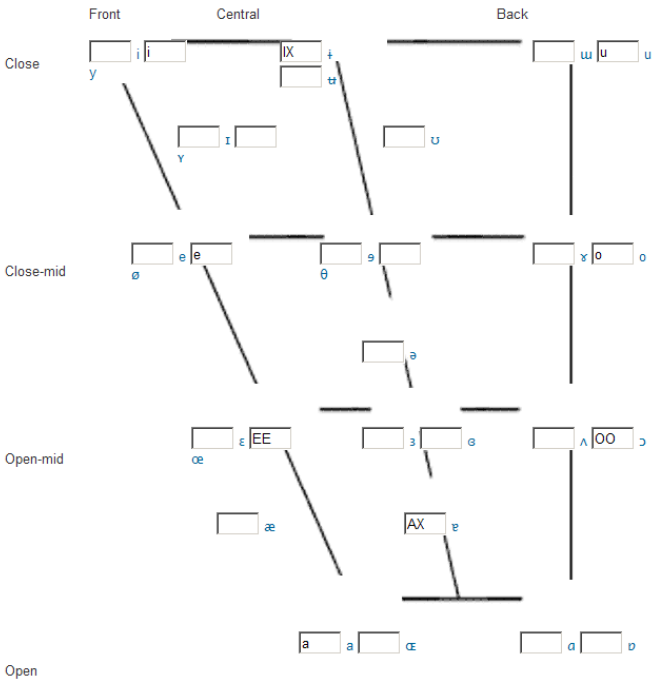

Figura 13 – Tabela disponível para as vogais

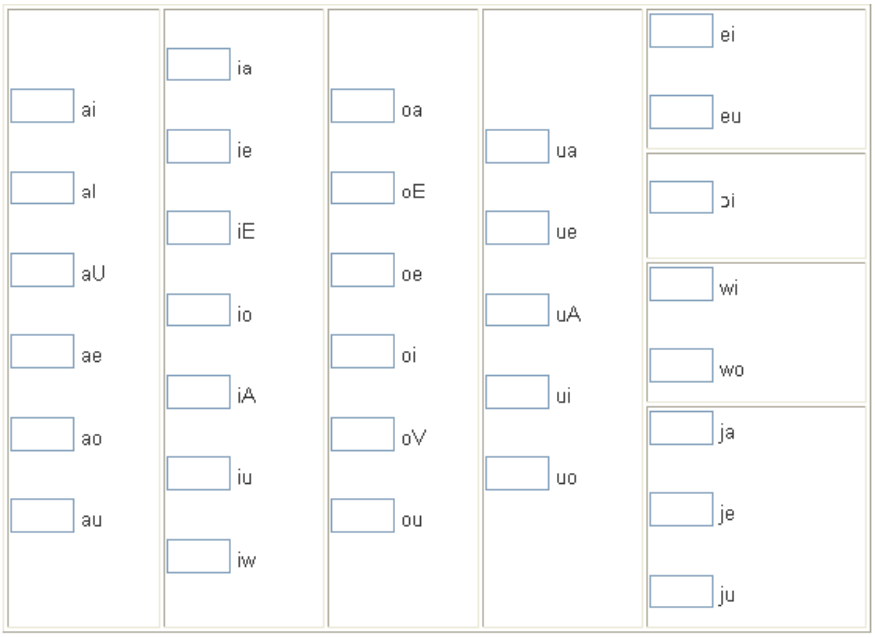

Figura 14 – Tabela dos ditongos

Após o preenchimento de todos os fonemas, o SPICE encarrega-se de gerar os ficheiros anteriormente descritos (“*PhoneMapFile”* e “*settings File”*). É de extrema

importância atribuir nomes diferentes a fonemas diferentes, de modo a evitar problemas de decisão ao sistema [24].

3.4.4. *Build Language Model*

Após a definição dos fonemas, o próximo passo é a construção de um modelo de linguagem.

Um modelo de linguagem representa a probabilidade de uma dada palavra ocorrer, dadas as palavras anteriores. Assim, considerando $W = w_1, \ldots, w_n$ como uma sequência de n palavras, a probabilidade dessa sequência, P (W) é dada por:

$$P(W) = \Pi_{t=1}^{n} P(w_t \mid w_1, \ldots, w_{t-1})$$

Assim, P ($w_t \mid w_1, \ldots, w_{t-1}$) é a probabilidade de ocorrer a palavra w_t dado que anteriormente ocorreu a sequência ($w_1, \ldots, w_{t-1}$), onde a sequência é conhecida como a *history* da palavra.

A primeira fase para a construção de um modelo de linguagem é definir um vocabulário, que deve ser constituído por um número finito de palavras.

Em segundo lugar, deve-se ter noção que não é viável calcular a probabilidade de uma palavra que tenha uma *history* muito longa, pois se tiver um vocabulário de tamanho V, existem V^{t-1} *histories* distintas e são precisos V^t valores para conseguir calcular P ($w_t \mid w_1, \ldots, w_{t-1}$), o que pode gerar um número astronómico de estimativas. Por este motivo, a *history* das palavras é dividida em classes equivalentes, onde cada *history* da palavra só pertence a uma classe equivalente ($(w_1, \ldots, w_{t-1}) \in K_t$, onde K_t representa uma classe equivalente).

O método mais comum para a divisão das *histories* das palavras é usar *N-grams*, onde, por exemplo, 3-*gram* significa a probabilidade de ocorrer uma palavra dadas as 2 palavras anteriores.

Os modelos de linguagem *N-gram* tentam ter atenção a restrições semânticas e sintáticas, para estimar a probabilidade de uma palavra numa frase precedida por N-1 palavras. As probabilidades dos *N-gram* podem ser estimadas durante o processo de

training, usando-se as frequências relativas de cada palavra. Mais especificações acerca de modelos de linguagem podem ser consultadas [5].

Para a criação de um modelo de linguagem, o SPICE desenvolve um algoritmo que se processa da seguinte maneira: em primeiro lugar, todos os textos carregados anteriormente no SPICE são copiados para a formação de um corpus; é calculada a lista de frequências de todos os caracteres e também a frequência de todas palavras; de seguida, é gerado o vocabulário, a partir do ficheiro das frequências das palavras, usando-se apenas as palavras que ocorrem mais que uma vez; após a construção do vocabulário, este é analisado e passa-se à construção dos modelos *N-gram*; por fim, há um teste ao corpus e são calculadas as probabilidades.

Para poder aceder aos "passos", *Grapheme-to-phoneme rules* e ao *Lexicon pronunciation creatio*, é necessário primeiro construir um modelo de linguagem.

3.4.5. *Grapheme Definition*

Nesta parte do sistema, é exigida a introdução de um conjunto de regras, num ficheiro "*G2P"*, necessárias para que o sistema consiga adivinhar a pronúncia correcta das palavras.

Outro ficheiro é também necessário nesta parte do sistema, o "*char.info*", que mostra a especificação de cada caracter, ou seja se é um caracter maiúsculo, minúsculo, número, pontuação, etc.

O ficheiro "*G2P*" e "*char.info*" podem ser inseridos no sistema pelo seu *upload* ou através do preenchimento *online* das caixas disponibilizadas no ambiente gráfico do SPICE [24].

Figura 15 – Ambiente gráfico para o preenchimento *on-line* do ficheiro *"G2P"*

Como mostra a figura, cada caixa deve ser preenchida com uma representação de um fonema, segundo o que foi definido no *Phoneme selection*, importante para a geração do ficheiro "*G2P"*. A escolha das especificações de cada caracter é necessária para a formação do ficheiro "*char.info"*.

Caso faça o *upload* destes ficheiros, é esperado que estes tenham os formatos dos exemplos apresentados abaixo.

Exemplo do conteúdo de um ficheiro "*G2P"*:

Caracter	**Representação do fonema correspondente**
Y	I
z	Z
Á	A
Â	AX
É	EE
Ó	OO

Tabela 3 – Exemplo do conteúdo de um ficheiro *"G2P".*

O ficheiro é constituído por duas colunas, uma à esquerda com os caracteres e outra à direita com os fonemas correspondentes a cada caracter, segundo as representações definidas no *Phoneme sellection*.

Exemplo do conteúdo de um ficheiro "*char.info"*.

Caracter	Especificação do caracter
'	other
-	other
.	punctuation
A	uppercase
B	uppercase
C	uppercase

Tabela 4 – Exemplo do conteúdo de um ficheiro *"char.info".*

O ficheiro é constituído por duas colunas, uma à esquerda com os caracteres e outra à direita com as especificações para cada caracter.

3.4.6. *Lexicon Pronunciation Creation*

O principal objectivo na aquisição de texto é mostrar ao sistema como uma língua é escrita. Por sua vez, a aquisição de dados áudio mostra como uma língua é falada. Da colaboração dos modelos de língua e dos modelos acústicos surgem modelos de pronúncia.

Para se construir um modelo de pronúncia, são necessários certos requisitos, tais como: a definição dos sons da língua (fonemas), a criação de um dicionário que descreve como as palavras de um idioma são pronunciadas e um conjunto de regras pós-lexicais, que alteram a pronúncia das palavras com o seu contexto.

Nesta parte do sistema, podemos usar a funcionalidade *Learn Rules*, para criar os dicionários de pronúncias (dicionário Festival e Janus) ou fazer o *upload* destes dicionários.

Caso se faça o *upload* de dicionários, é importante respeitar certos formatos e normas.

Existem dois tipos de dicionários: um segundo o formato Festival e outro segundo o modelo Janus [24].

Como criar os dicionários:

Segue-se agora um exemplo de como deve ser o formato do dicionário Festival (exemplo para a palavra *blue* (exemplo retirado do *help* do SPICE)):

("blue" nil (b j uw1)) -> todas as palavras do dicionário devem seguir este formato.

O espaço "*nil*" é reservado para informações de discurso.

Do lado direito, a palavra é enunciada como uma *string* de fonemas.

Caso exista já um dicionário criado, é possível então fazer o seu *upload*. No caso do dicionário no formato Festival, é obrigatório que este ficheiro tenha o nome de "*dictionary-festival*" [24].

Ao fazer o *upload* do dicionário Festival, o dicionário Janus é criado automaticamente pelo SPICE. Só se deve fazer *upload* de um dicionário no formato Janus se se pretender ter dicionários diferentes para o ASR e para o TTS.

A outra maneira de se criarem os dicionários de pronúncia, é através do próprio SPICE, que permite gerar os dicionários e um conjunto de regras para a conversão grafema-fonema, de forma iterativa com o utilizador. O SPICE selecciona palavras do léxico e apresenta sugestões de pronúncia. Há a possibilidade de aceitar, modificar, ignorar ou remover estas sugestões (sequência de fonemas), criadas para cada palavra. A figura que se segue mostra o ambiente gráfico descrito.

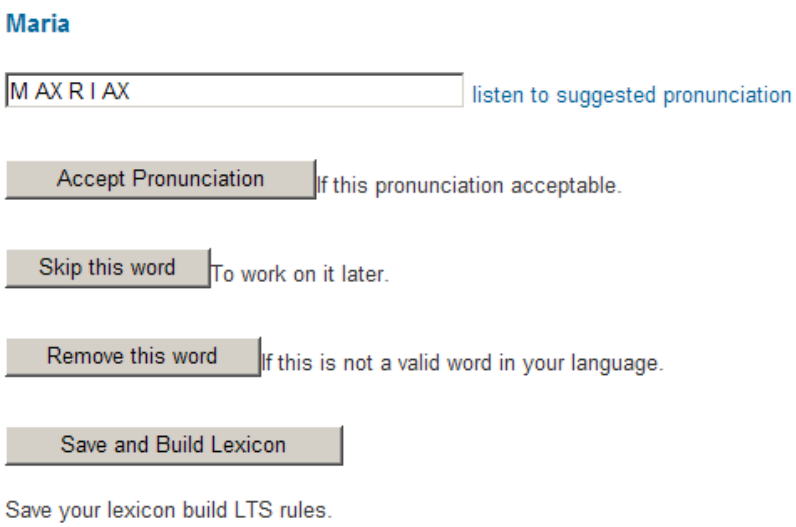

Figura 16 – Ambiente gráfico para a predição de uma pronunciação.

O seguinte exemplo mostra uma possível regra gerada.

Exemplo: (C [a] s = A)

Isto significa que o caracter [a], precedido de [C] e seguido de [s] é lido como [A] ([A] é uma possível representação para o fonema, que corresponde ao caracter [a]).

3.4.7. Gravação (*Audio Collection*)

O TTS exige a geração de fala humana, juntamente com as suas transcrições. Os dados áudio são importantes para a formação e treino de modelos acústicos, essenciais na síntese de voz. Estes modelos acústicos têm um impacto positivo nos modelos de pronúncia, pois tornam possível ouvir um *feedback*.

O SPICE tem disponível, no seu ambiente, um pequeno *script* em Java capaz de fazer gravações.

Para a aquisição de dados áudio é necessário cumprir vários procedimentos: em primeiro lugar, deve-se verificar a lista de *prompts* e se esta se encontra no formato adequado, UTF-8; em segundo, deve-se definir o caminho no disco local onde os dados

gravados irão ser armazenados; em terceiro, deve-se iniciar uma nova secção de gravação, para um determinado ID [24].

Os *prompts* são apresentados e lidos um a um. Sempre que um registo é concluído, é possível o *upload* da gravação para o servidor, refazer a gravação, passar para o seguinte *prompt* ou até saltar *prompts*.

Quando a secção é encerrada, toda a informação adquirida, aquando das gravações, é copiada para o disco local e enviada mais tarde para o servidor do SPICE.

A seguinte imagem mostra o processo de uma gravação no SPICE.

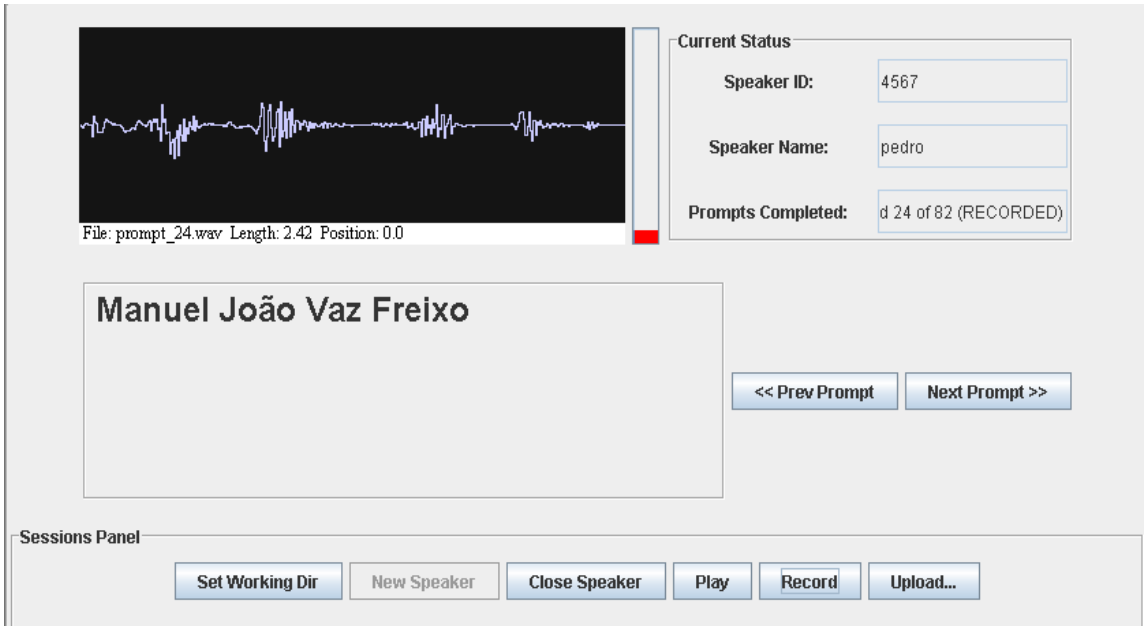

Figura 17 – Ambiente gráfico para a gravação de *prompts*

Há também a possibilidade de fazer o *upload* de ficheiros áudio (.wav) para os servidores do SPICE.

3.4.8. *Creat speech synthesis voice*

Este é o último passo para a criação de uma voz. Após todas as etapas anteriores estarem concluídas, o SPICE fica preparado para a construção da voz.

Este processo da criação de voz recolhe de forma sequencial todo um conjunto de informações geradas anteriormente e cria um conjunto de modelos e ficheiros essenciais à construção da voz.

A seguinte imagem mostra o ambiente gráfico do "*creat speech synthesis voice*" antes de se proceder à geração da voz.

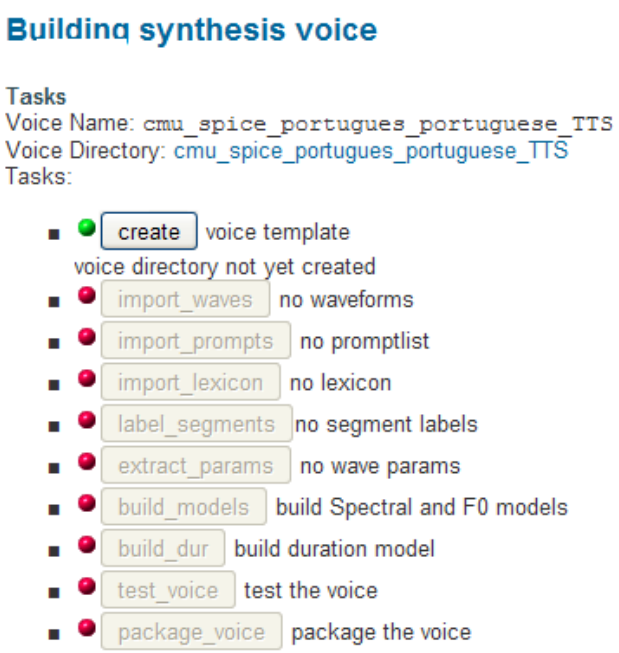

Figura 18 – Ambiente gráfico do creat speech synthesis voice antes de se proceder à construção da voz.

O primeiro passo para a geração da voz é a criação de um *template*, segundo as exigências do Festival, gerando-se um directório com o nome "*cmu_spice_NAME*" (sendo *NAME* o nome escolhido para o projecto). Neste directório, são gerados mais dois directórios, um chamado "festival" e outro chamado "festvox", onde serão armazenadas todas as informações de modelos, ficheiros e funções, essenciais à criação da voz. Todo este conjunto de funções e ficheiros são desenvolvidos ao longo deste processo da geração da voz.

Para que a voz seja obtida é necessário então: importar os *prompts* (em formato áudio) gravados anteriormente, de extrema importância para o treino dos modelos que irão ser criados; importar os *promts* (em formato de texto), importantes para a criação do

corpus e do léxico; importar o léxico gerado, assim como todas as regras de conversão gafema-fonema.

Após a captação destes dados, são criados ficheiros com extensão .lab e .sl , para cada *prompt*. Estes ficheiros são relevantes para o treino dos modelos de duração. São criados também os ficheiros essenciais para a técnica de síntese a usar (c*lustergen*) e as árvores, essenciais à criação dos modelos de F0, de entoação e duração.

Com toda esta informação disponível, o sistema fica capaz de construir os modelos e funções necessárias para que a síntese seja possível. Todos os dados e funções geradas são armazenados no template, gerado anteriormente, obtendo-se assim uma voz pronta a ser usada no Festival.

A seguinte imagem mostra o "*Creat speech synthesis voice*", após todo o processo estar concluído.

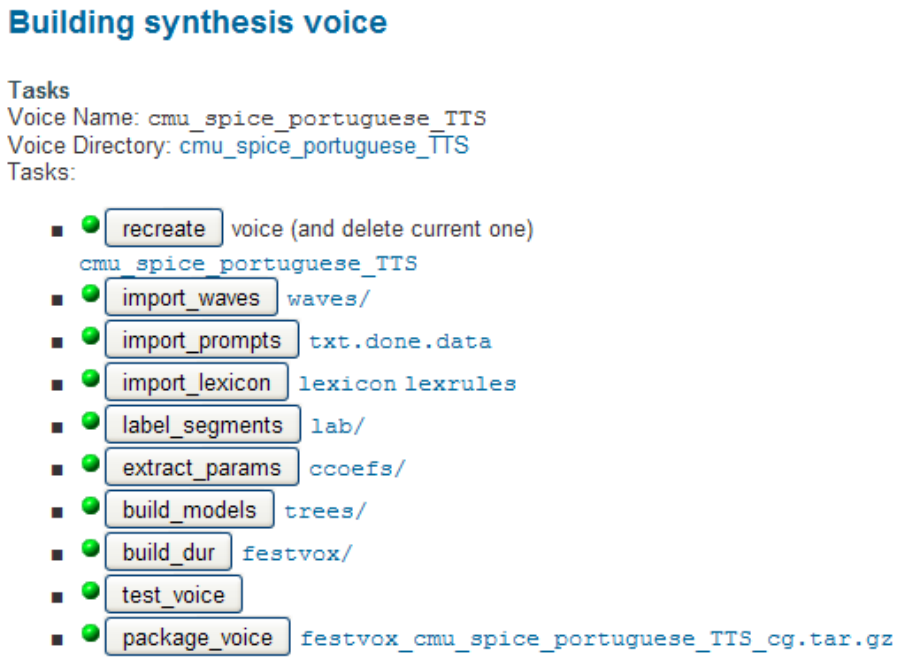

Figura 19 – Ambiente gráfico do *creat speech synthesis voice* depois de se proceder à construção da voz

Dentro do directório "festvox", ficam guardados todos os ficheiros necessários à construção da voz, como descrito na secção 3.2 (ficheiros para a criação do conjunto de fonemas, para o processador de texto, para o analisador de Léxico, para as regras de

conversão grafema-fonema, para os modelos de predição de pausas entre frases, entoação e durações e sobre o tipo de síntese a usar).

As vozes cridas, pelo SPICE usam a técnica de síntese *Clustergen* [14]. Para a criação das durações é usado o método *Tree zscore* (método que usa uma árvore para predizer um factor a ser multiplicado pela duração média dos segmentos, em vez de predizer a duração dos segmentos directamente [23]).

Para a geração da entoação é usado o método *General intonation* (já descrito na secção 3.2), onde a previsão dos acentos é feita tal como no *Simple intonation*, através de uma árvore de decisão. Por sua vez, os contornos de F0 são determinados através de uma função que retorna uma lista de pontos de entoação para as sílabas. Mais detalhes podem ser consultados no manual do Festival [23].

Capítulo 4 - Desenvolvimento de um Sistema de Síntese de Nomes em Português, para o Festival

Neste capítulo, serão apresentados e explicados todos os processos práticos desenvolvidos para a construção de uma voz capaz de sintetizar nomes em português.

Inicialmente, pensou-se em desenvolver todo o processo da criação de voz com o uso exclusivo do SPICE, mas tal facto não se revelou possível devido a algumas limitações deste sistema. Como tal, optou-se por uma solução híbrida, em que toda a parte da geração de regras de conversão grafema-fonema, definição dos fonemas, normalização de texto e predição de pausas, entoação, durações fosse feita com o SPICE e a parte de geração de sinal sonoro fosse feita através de um sintetizador externo, o MBROLA.

É importante referir que devido à incapacidade do SPICE em adquirir dados áudio, todos os modelos de duração não foram treinados suficientemente para proporcionarem boa qualidade, optando-se então em adaptar a estes modelos, criados no SPICE, outros provindos de outras vozes já construídas como: a voz criada em [2] e uma voz do português brasileiro, do *CSLU toolkit* [33].

De seguida, serão descritas todas as etapas desenvolvidas no processo prático da criação da voz.

4.1. Aquisição de Texto para o SPICE

O primeiro passo para o desenvolvimento do sistema é a aquisição de texto, como já referido. Este passo é de extrema importância para a criação do vocabulário e para o conhecimento da língua escrita, por parte do sistema. A recolha de texto deve ser contextualizada de acordo com as necessidades da aplicação da voz a sintetizar, sendo neste caso nomes próprios. Neste trabalho, optou-se por fazer o *upload* de um ficheiro de texto em detrimento de uma busca na *web*.

Para obter um ficheiro de texto composto única e exclusivamente por nomes completos, foi necessária a pesquisa de uma base de dados com um enorme volume de nomes. Foram consultados os sites do instituto dos registos e do notário (que só continham

nomes próprios, mas não completos), das páginas amarelas (que para além dos nomes continha alguma informação desnecessária) e a lista de professores universitários em Portugal do final de 2007 (com informações desnecessárias sobre a qualificação e o curso dos professores).

Optou-se por usar como base a lista de nomes de professores universitários, pois continha um número de nomes mais do que suficiente para os fins a alcançar. Para a limpeza desta lista, criou-se um programa em Perl que encontrava as palavras mais frequentes antes de um nome e apagava-as. O resto da limpeza do ficheiro foi feita manualmente.

O ficheiro de texto obtido foi codificado no formato UTF-8, contendo um nome completo por linha. Após o *upload* deste ficheiro, segue-se o próximo passo, a geração de *prompts*.

4.2. Geração de *Prompts* para o SPICE

A geração de *prompts* pode ser feita automaticamente pelo SPICE ou pelo *upload* de um ficheiro já existente.

No caso deste projecto, não foi possível a geração automática dos *prompts*, pois apesar da enorme recolha de nomes completos, o número de palavras diferentes era inferior a 5000, o que invalida o algoritmo de geração de *prompts* do SPICE (referido no capítulo 3).

Deste modo, optou-se por carregar o sistema com um ficheiro de *prompts*. Este ficheiro tem de estar codificado em UTF-8 e todos os *prompts* têm de respeitar o formato, como já descrito no capítulo 3.

Para a geração deste ficheiro, criou-se um programa em Perl. Este processava o ficheiro de nomes completos, referido anteriormente e de forma aleatória (controlada por um rand), escolhia todos os que estivessem acima de um dado valor (muito próximo da unidade, limite superior do rand), escrevendo-os num ficheiro de saída, segundo o formato já descrito e na codificação adequada.

O ficheiro criado tinha um total de 250 *prompts*. Este número de *prompts* tem a sua base de fundamento no artigo de John Kominek, Tanja e Alan Black [13], onde se refere, como boa prática, usar sempre um número superior a 200 *prompts* para gravação.

4.3. Conversão Grafema-fonema

O primeiro passo para a conversão de grafemas em fonemas é a definição dos sons da língua (fonemas).

Enumera-se, em seguida, os fonemas usados para o português, utilizando o alfabeto SAMPA e o IPA e também as representações usadas no sistema em construção.

A seguinte tabela mostra as representações das consoantes.

Classe	**Representação segundo o alfabeto IPA**	**Representação segundo o alfabeto SAMPA**	**Representação usada no sistema**	**Exemplo do fonema numa palavra**
Oclusivas	p	p	P	Pai
	t	t	T	Tia
	k	k	K	Casa
	b	b	B	Bar
	d	d	D	Data
	g	g	G	Gato
Fricativas	f	f	F	férias
	s	s	S	Selo
	ʃ	S	SS	chave
	v	v	V	Vaca
	z	z	Z	Azul
	ʒ	Z	ZZ	Jacto
Nasais	m	m	M	Meta
	n	n	N	Neta
	ɲ	J	JJ	Senha
Laterais	l	l	L	Lado
	ʎ	L	LL	Folha
Vibrante simples (Tap)	ɾ	r	R	Caro
Vibrante múltipla (Trill)	r	R	RR	Carro

Tabela 5 – Representações das consoantes.

A seguinte tabela mostra as representações das vogais orais.

Classe	Representação segundo o alfabeto IPA	Representação segundo o alfabeto SAMPA	Representação usada para o sistema em construção	Exemplo do fonema numa palavra
Vogais	ɐ	6	AX	C̲am̲a̲
	a	a	A	C̲a̲ra
	e	e	E	P̲ê̲ra
	ɛ	E	EE	S̲e̲te
	ɨ	@	IX	Pot̲e̲
	i	i	I	F̲i̲ta
	o	o	O	Topo
	ɔ	O	OO	P̲o̲te
	u	u	U	B̲u̲da

Tabela 6 – Representações das vogais orais.

Para os ditongos optou-se por transcrever apenas os decrescentes, sendo os ditongos crescentes lidos pelo sistema como uma sequência de duas vogais e não como um ditongo.

A seguinte tabela, mostra todos os ditongos orais decrescentes usados no sistema, de acordo com a lista apresentada em [4].

Representação usado no SPICE	Representação segundo o alfabeto SAMPA	Representação usada no sistema	Exemplo do fonema numa palavra
ai	aj	AJ	P̲a̲i̲
al	6j	AXJ	D̲e̲i̲
oi	oj	OJ	F̲o̲i̲
ei	Ej	EJ	Papéis
ui	uj	UJ	Azuis
ɔi	Oj	OOJ	Dói
	Ew	*	Céu
eu	ew	EW	M̲e̲u̲
iw	iw	IW	V̲i̲u̲
au	aw	AW	M̲a̲u̲

Tabela 7 – Representação dos ditongos orais decrescentes usados no sistema.

* O ditongo [Ew] (representação segundo o SAMPA) é lido pelo sistema como uma sequência de duas vogais e não como ditongo.

O SPICE apresenta limitações na representação de alguns fonemas do português, não contemplando um espaço para a representação de semivogais, vogais nasais e ditongos nasais.

Inicialmente, para resolver o problema da nasalidade das vogais e dos ditongos, optou-se por associar a cada fonema correspondente a uma vogal ou a um ditongo um fonema com características nasais, sendo a nova representação destas vogais e ditongos feita com a associação de dois fonemas. O fone usado para dar uma característica nasal aos ditongos e às vogais foi o [ɲ] (representação segundo o IPA), sendo representado no sistema por [NN].

As seguintes tabelas enumeram todos os ditongos e vogais nasais usados no sistema.

Visto o SPICE não contemplar um espaço para o mapeamento destes casos, estes são introduzidos no sistema apenas durante o processo de aprendizagem das regras de conversão grafema-fonema, necessárias para a formação dos dicionários de transcrições e para a criação de regas LTS.

A seguinte tabela enumera as representações das vogais nasais.

Representação segundo o alfabeto IPA	**Representação segundo o alfabeto SAMPA**	**Representação usada no sistema**	**Exemplo do fonema numa palavra**
ɐ̃	6~	AX NN	C<u>an</u>to
ẽ	e~	E NN	P<u>en</u>te
ĩ	i~	I NN	P<u>in</u>ta
õ	o~	O NN	P<u>on</u>to
ũ	u~	U NN	M<u>un</u>do

Tabela 8 – Representação das vogais nasais.

A tabela abaixo mostra a lista de ditongos nasais decrescentes, usada no sistema, de acordo com a tese de [4]. Os ditongos nasais crescentes são lidos pelo sistema como a sequência de uma vogal oral com outra nasal.

Representação segundo o alfabeto IPA	Representação segundo o alfabeto SAMPA	Representação usada no sistema	Exemplo do fonema numa palavra
ɐ̃w	6~w~	AW NN	C<u>ão</u>
ɐ̃j	6~j~	AJ NN	M<u>ãe</u>
õj	o~j~	OJ NN	P<u>õe</u>
ũj	u~j~	UJ NN	M<u>ui</u>to

Tabela 9 – Representação dos ditongos nasais decrescentes.

Durante o preenchimento das tabelas do *Phoneme selection*, surgiram problemas com a representação de alguns dos fonemas. O SPICE não reconhece representações com letras minúsculas, convertendo-as em maiúsculas. Como inicialmente se optou por representar alguns dos fonemas com letras maiúsculas e outros com minúsculas, havia casos em que diferentes fonemas tinham a mesma representação. Este problema foi resolvido, optando-se por representar os fonemas "maiúsculos" com duas letras maiúsculas iguais.

Por fim, é importante referir que o sistema não contempla o fonema [l~], usado em palavras como nat<u>al</u> e pap<u>el</u>, sendo este assumido pelo fonema [l].

Todos os problemas com as representações dos fonemas e com as limitações do SPICE estavam então resolvidos, mas a partir do momento em que houve necessidade do uso do MBROLA, com a base de dados de difones pt1 (disponível em [22]), tiveram que se fazer alguns ajustes na criação do conjunto de fonemas.

Esta base de dados de difones (pt1) é do português europeu, sendo constituída por 1369 difones (com uma voz feminina). Do conjunto dos fonemas contemplados por estes difones, fazem parte as semi-vogais [w] e [j]e as vogais nasais [i~], [e~], [6~], [o~] e [u~], não contempladas pelo SPICE.

Com o uso do MBROLA, optou-se então por não se carregar o sistema (SPICE) com os símbolos representativos dos ditongos decrescentes e "forçou-se" a representação das vogais nasais e das semi-vogais, como demonstrado na figura a baixo.

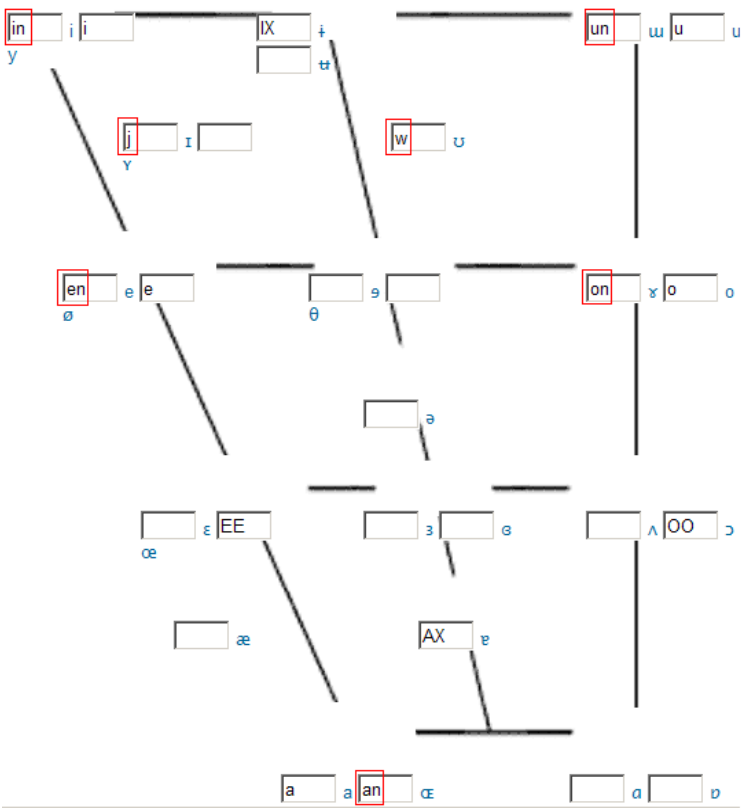

Figura 20 – Representação das vogais nasais e das semi-vogais

Nota: ([an] corresponde a [6~], [en] a [e~], [in] a [i~], [on] a [o~], [un] a [u~] , [w] a [w] e [j] a [j]).

Os ditongos orais e nasais decrescentes passam assim a ser lidos como a sequência de uma vogal com uma semi-vogal.

Os fones que representam as vogais nasais e as semi-vogais (no SPICE) não correspondem, na verdade, ao som das vogais nasais e das semi-vogais. Tal facto, torna-se pouco relevante, pois a partir do momento que se utiliza o MBROLA, pretende-se apenas obter as regras LTS e os dicionários gerados pelo SPICE, que, por sua vez, têm de estar de acordo com o *fone set* da base de dados de difones usada no MBROLA.

Aquando da síntese de voz, é no MBROLA que ocorre a correcta correspondência dos fones aos símbolos das vogais nasais e semivogais. Esta estratégia permitiu, de algum modo, colmatar as limitações do SPICE na representação de vogais nasais e semi-vogais.

Após a definição de todos os fonemas, passámos à construção do modelo de linguagem, disponível no *Build language model*. Este modelo é construído através de um processo automático, como já referido anteriormente no capítulo 3.

Só após a criação do modelo, é possível aceder às restantes etapas necessárias para a conversão dos grafemas em fonemas.

Após a construção do modelo de linguagem, é necessário definir um conjunto de regras. Estas regras são introduzidas num ficheiro "*G2P"* e é através destas que o sistema consegue gerar uma possível transcrição para a pronúncia das palavras. Este processo de transcrição baseado em regras permite ao sistema atribuir a cada caracter de texto (proveniente dos textos carregados anteriormente) o fonema correspondente.

Para além do ficheiro "*G2P*", há um outro associado, o "*char.info*", que contém as especificações de cada fonema, como já referido no capítulo3.

A seguinte tabela mostra o conteúdo do ficheiro "*G2P"* criado para o sistema.

A A	P O	c K	q K	Ó OO	ñ JJ
B B	Q K	d D	r RR	Ô O	ó OO
C K	R RR	e EE	s S	à A	ô O
D D	S S	f F	t T	á A	õ O NN
E EE	T T	g G	u U	â AX	ú U
G G	U U	i I	v V	ã AX NN	ü U
I I	V V	j ZZ	w U	ä AX	
J ZZ	W U	k K	x SS	ç S	
K K	X SS	l L	y I	é EE	
L L	Y I	m M	z Z	ê E	
M M	Z Z	n N	Á A	ë E	
N N	a A	o OO	Â AX	í I	
O OO	b B	p P	É EE	î I	

Tabela 10 – Conteúdo do ficheiro *"G2P"* criado para o sistema.

Com o uso do MBROLA, os casos do "ã AX NN" e do "õ O NN", passam a ser "ã AN" e "õ ON".

A seguinte tabela mostra o conteúdo do ficheiro "*char.info"* gerado para o sistema.

. punctuation	O uppercase	d lowercase	s lowercase	â lowercase
A uppercase	P uppercase	e lowercase	t lowercase	ã lowercase
B uppercase	Q uppercase	f lowercase	u lowercase	ä lowercase
C uppercase	R uppercase	g lowercase	v lowercase	ç lowercase
D uppercase	S uppercase	h lowercase	w lowercase	é lowercase
E uppercase	T uppercase	i lowercase	x lowercase	ê lowercase
F uppercase	U uppercase	j lowercase	y lowercase	ë lowercase
G uppercase	V uppercase	k lowercase	z lowercase	í lowercase
H uppercase	W uppercase	l lowercase	Á uppercase	î lowercase
I uppercase	X uppercase	m lowercase	Â uppercase	ñ lowercase
J uppercase	Y uppercase	n lowercase	É uppercase	ó lowercase
K uppercase	Z uppercase	o lowercase	Ó uppercase	ô lowercase
L uppercase	a lowercase	p lowercase	Ô uppercase	õ lowercase
M uppercase	b lowercase	q lowercase	à lowercase	ú lowercase
N uppercase	c lowercase	r lowercase	á lowercase	ü lowercase

Tabela 11 – Conteúdo do ficheiro *"char.info"* gerado para o sistema.

Após a definição das regras, há a necessidade da criação de um dicionário de pronúncias e de um conjunto de regras LTS. Este processo é desenvolvido no *Pronunciation creation*. A criação de um dicionário, como já explicado anteriormente, tem a finalidade de descrever como as palavras de um idioma são pronunciadas.

Nesta parte do sistema, podemos usar a funcionalidade do próprio SPICE, para criar os dicionários de pronúncias (dicionário Festival e Janus) e as regras LTS. A outra forma do SPICE adquirir os dicionários é através do seu *upload*.

Para a obtenção dos dicionários, optou-se por usar a funcionalidade do *Learn Rules* (do SPICE), gerando-se um dicionário festival com 1544 palavras. Após o dicionário estar disponível, no SPICE, este foi copiado para um ficheiro de texto e verificado manualmente, corrigindo-se alguns erros de transcrição.

Para a geração das regras LTS, usou-se o dicionário criado anteriormente (de 1554 palavras), o conteúdo do ficheiro "*G2P"* e a funcionalidade do *Learn Rules*, onde o SPICE selecciona palavras do léxico e apresenta uma sugestão de pronúncia, para cada uma delas, cabendo ao utilizador a decisão de aceitar, modificar, ignorar ou remover esta sugestão. É através deste sistema de aprendizagem (proporcionado pelo *Learn Rules*), inicialmente

suportado pelo ficheiro "*G2P"* e pelos dicionários de pronúncias, que o sistema evolui e cria as regras essenciais à geração da pronúncia das palavras.

4.4. Criação da Voz

Para a criação da voz, usou-se como base a voz "*cmu_spice_portuguese_TTS_cg*", criada com um número insuficiente de dados áudio. Foi aproveitada toda a sua estrutura, com todos os ficheiros e funções, alterando-se alguns dos modelos para a prosódia pouco treinados. As regras LTS e o *fone set*, foram alterados de modo a que se respeitassem os formatos impostos pela base de dados do MBROLA. Foi também "forçado" o uso de um sintetizador externo, o MBROLA (com a base de dados de difones pt1).

Como já descrito no capítulo 3.2, uma voz é constituída por um conjunto de ficheiros e funções. Todas as alterações, para a construção da nova voz, foram feitas directamente nos ficheiros disponíveis no directório festvox, da voz "*cmu_spice_portuguese_TTS_cg*".

Um dos ficheiros alterados foi o "*cmu_spice_portuguese_TTS_phoneset*". Este ficheiro teve de ser ajustado, de modo a ser compatível com o uso do MBROLA. Para isso, foi necessário mudar a representação usada no SPICE, para cada fonema, para uma representação em SAMPA. Ainda neste ficheiro, foram eliminadas as representações dos ditongos, que não fazem parte do conjunto de fonemas da base de dados de difones pt1 e foram adicionadas as vogais nasais e as semi-vogais.

Como já descrito no capítulo 3.2, a cada fonema do *fone set* está associado um conjunto de características. A adição das vogais nasais e semi-vogais foi muito simples: no caso das vogais nasais copiou-se as características da vogal [6] para [6~], da [e] para [e~], da [i] para [i~], da [o] para [o~], da [u] para [u~] e acrescentou-se a característica nasal; Para o caso das semi-vogais, copiou-se as características de [i] para [j], de [u] para [w] e eliminou-se a característica vogal. Foram também adicionados ao *fone set* dois silêncios representados pelo símbolo "pau" e "_".

No ficheiro *cmu_spice_portuguese_TTS_lexrules.scm*, foi alterado e introduzido, de forma manual, um conjunto de novas regras LTS. Ainda neste ficheiro, todas as regras

LTS foram processadas por um programa em perl, de modo a garantir o formato imposto pelo MBROLA, segundo o SAMPA.

Para se forçar o uso do sintetizador MBROLA, dois ficheiros sofreram alterações: o "*clustergen.scm"* e o "*cmu_spice_portuguese_TTS_cg.scm"*. No ficheiro "*clustergen.scm"*, foram comentadas todas as linhas de código necessárias para a síntese *clustergen* e adicionado o código capaz de proporcionar a síntese MBROLA. No ficheiro "*cmu_spice_portuguese_TTS_cg"*, na função "*voice_cmu_spice_portuguese_TTS_cg*" (função responsável pela construção da voz, capaz de chamar todas as funções definidas nos outros ficheiros, onde estão definidos todos os parâmetros essenciais à criação da voz), foram comentadas as linhas de código responsáveis pela "chamada" dos parâmetros essenciais à síntese *clustergen*, com a excepção da linha que garante a execução do ficheiro "*clustergen.scm"*, pois é nesse ficheiro que está definida a utilização do sintetizador MBROLA, todas estas alterações foram baseadas numa voz do português brasileiro, do *CSLU toolkit* [33].

O método usado para as durações também sofreu alterações, passando-se a usar o método *Tree_zscore*, com uma lista de durações para cada fonema oriunda de outra voz. Para isto, foram adicionadas aos ficheiros "*cmu_spice_portuguese_TTS_durdata"* e "*cmu_spice_portuguese_TTS_durdata_cg"* as listas de durações médias para cada fonema, retiradas da voz criada em [2].

No caso da geração de entoação, houve apenas uma pequena alteração no ficheiro "*cum_spice_portuguese_TTS_intonation*", onde foram acrescentadas umas linhas de código na árvore de decisão para a acentuação das palavras monossilábicas. Deste modo, a árvore, para além de gerar um acento nas sílabas acentuadas das palavras polissilábicas, passa a gerar também um acento em todas as palavras constituídas por uma única sílaba.

Por fim, de modo a que o Festival conseguisse ler caracteres com acentos ou cedilhas, foi necessário eliminar o uso da função *utf8explode*, presente em muitos destes ficheiros.

4.5. Exemplos de Síntese

A título de exemplo do que é possível obter no sistema, apresenta-se, na seguinte figura, o sinal sonoro e o espectrograma relativo, correspondentes ao resultado da síntese do nome "Cláudia Margarida Correia Balula Chaves".

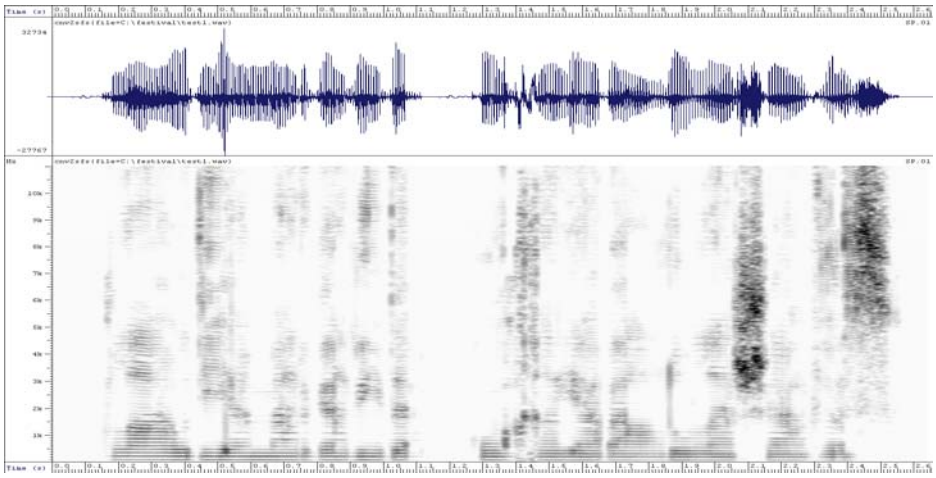

Figura 21 – Sinal sonoro e espectrograma

4.6. Avaliação do Sistema

Para fazer a avaliação da voz, foram feitos dois tipos de testes distintos, um para avaliar a conversão grafema-fonema, outro para avaliar a identificação dos nomes por um conjunto de ouvintes.

4.6.1. Teste da Conversão Grafema-fone

Neste teste, foram processados 50 nomes completos, retirados de forma aleatória do ficheiro de nomes de professores universitários de 2007.

A avaliação da conversão grafema-fonema foi feita através da comparação da conversão grafema-fonema dos nomes, realizada de forma manual por uma especialista em Linguística, com os outputs gerados pelo Festival.

4.6.1.1 Resultados

No total dos 50 nomes, existiam1341 fonemas a serem transcritos. Após a conversão grafema-fonema, 110 destes fonemas não foram convertidos correctamente. Este valor corresponde a uma taxa de erro de 8,2% (que corresponde a uma taxa de acerto de 91,8%), um resultado bastante aceitável.

O gráfico abaixo mostra todos os fones que foram erradamente transcritos pelo sistema, assim como o número de vezes que tal erro ocorreu.

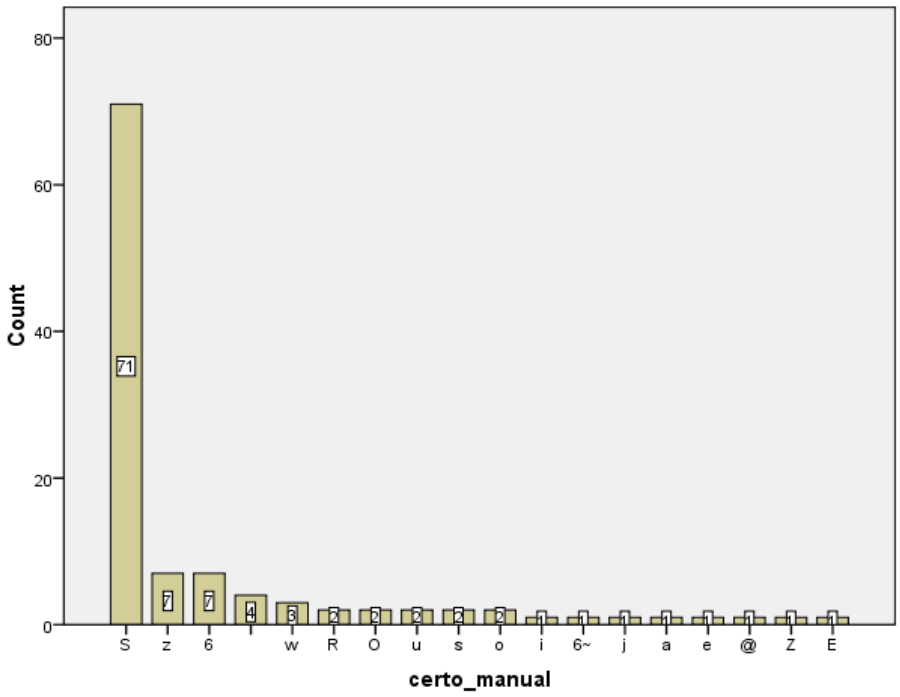

Figura 22 – Gráfico dos erros cometidos pelo sistema, ao nível dos fones.

Pode-se verificar que 64% dos erros de conversão grafema-fonema se devem à transcrição errada do fone [S], quase sempre mal convertido para o fone [s]. No caso do

fone [z], este é normalmente transcrito como [s] e o [6] é algumas vezes substituído pelo fone [a].

Estes erros devem-se ao uso de regras de conversão grafema-fonema indevidas ou à falta delas. Tal facto poderia ser minorado com a adição e correcção manual de regras de conversão grafema-fonema, por parte de um especialista.

É também importante referir que 10% dos erros provêm de palavras de origem estrangeira. Se considerarmos apenas palavras estrangeiras, cujo número de fonemas corresponde a 48, o número de fonemas erradamente transcritos é igual a 11, o que revela uma taxa de acerto 77,1%. Este valor mais baixo é normal, pois a voz foi criada para ler nomes próprios apenas em português.

Visto a voz estar programada só para a leitura de nomes em português, é relevante referir que, considerando apenas a conversão dos fonemas de palavras portuguesas (1293), a taxa de acerto aumenta para 92,3%.

4.6.2. Teste de Identificação

Neste tipo de avaliação, a capacidade de percepção de cada nome está intimamente ligada ao ouvinte.

Neste teste, foi proposto a um conjunto de 8 pessoas que ouvissem, individualmente, 40 nomes completos (um a um) e que identificassem cada um deles.

O perfil de cada um dos participantes no teste é apresentado na seguinte tabela:

	Idade	**Sexo**	**Ocupação**	**Localidade**
Ouvinte 1	26	Feminino	Estudante universitário	Santa Maria da Feira
Ouvinte 2	23	Masculino	Estudante universitário	Avanca
Ouvinte 3	23	Masculino	Estudante universitário	Estarreja
Ouvinte 4	23	Masculino	Estudante universitário	Santa Maria da Feira
Ouvinte 5	23	Masculino	Estudante universitário	Santa Maria da Feira
Ouvinte 6	55	Masculino	Empregado bancário	Santa Maria da Feira
Ouvinte 7	22	Masculino	Engenheiro Civil	Aveiro
Ouvinte 8	54	Feminino	Educadora de Infância	Vale de Cambra

Tabela 12 – Informações acerca de cada ouvinte.

Para a apresentação dos nomes aos ouvintes, foi criado uma aplicação de administração do teste que, entre outras coisas, apresentava cada nome escrito ao administrador do teste (o autor desta dissertação), durante a apresentação da voz sintetizada pelo sistema, de modo a facilitar o registo dos nomes identificados por parte dos ouvintes. É de realçar que os ouvintes não tinham acesso a esta representação escrita dos nomes, de modo a não influenciar as suas opções.

A seguinte imagem mostra o ambiente gráfico descrito anteriormente.

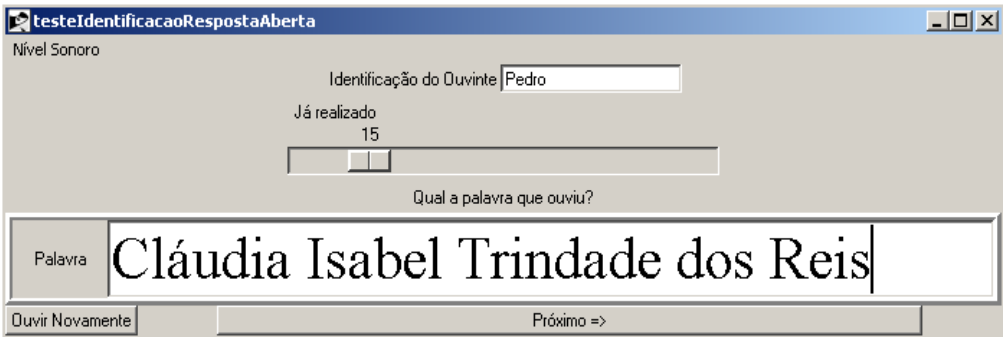

Figura 23 – Ambiente gráfico para o administrador no teste de identificação.

Os nomes foram apresentados aos ouvintes apenas uma vez, com a excepção de nomes completos com mais de 5 palavras, lidos duas vezes.

Os ouvintes ouviram os nomes através de *headphones*, num ambiente de baixo ruído sonoro.

Os quarenta nomes lidos foram escolhidos de forma aleatória, através de um programa em Perl, que processou o ficheiro de nomes de professores universitários, já referido anteriormente. O conjunto destes quarenta nomes completos é constituído por um total de 187 palavras, o que corresponde a um número médio de 4.67 palavras por nome.

O seguinte histograma mostra a relação do número de nomes completos com o número de palavras por nome (este número de palavras pode oscilar entre 3 e 8).

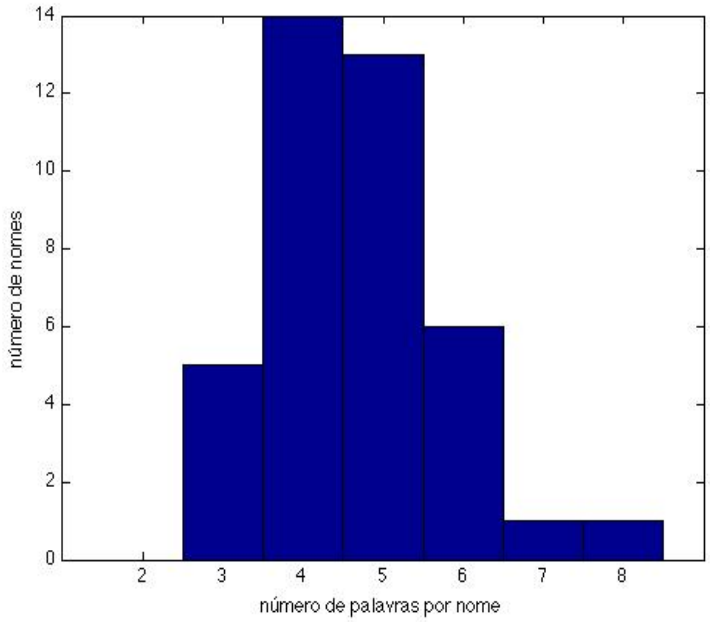

Figura 24 – Histograma de relação do número de nomes completos com o número de palavras por nome.

O seguinte gráfico mostra a frequência dos caracteres existentes em cada um dos quarenta nomes completos.

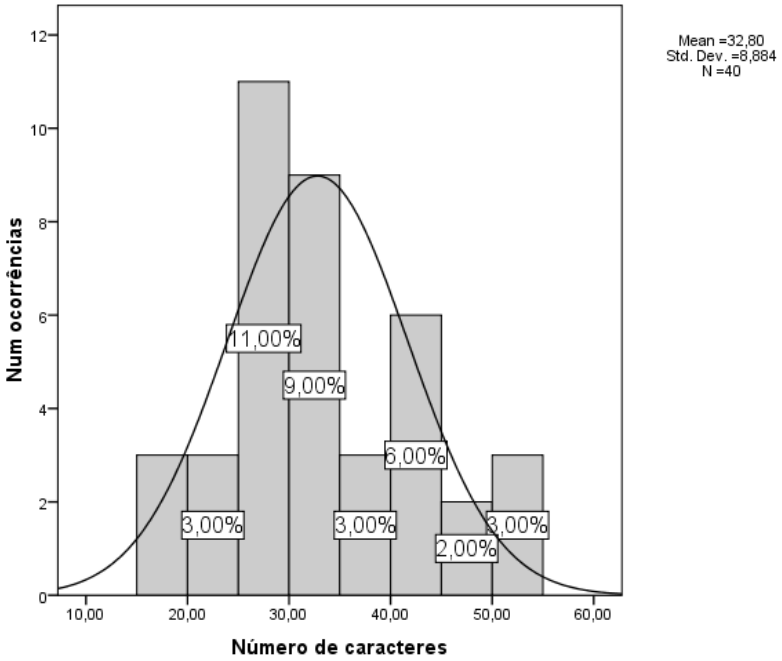

Figura 25 – Gráfico da frequência dos caracteres.

Cada nome completo tem um número médio de 32,78 caracteres.

4.6.2.1. Resultados

Após a conclusão do teste de identificação, foram calculados dois tipos de resultados para cada um dos ouvintes: a percentagem de nomes correctamente identificados, no total das 187 palavras que constituem os quarenta nomes completos e a percentagem de nomes completos, sem qualquer erro de identificação.

O seguinte gráfico representa o número de nomes correctamente identificados, do total das 187 palavras existentes.

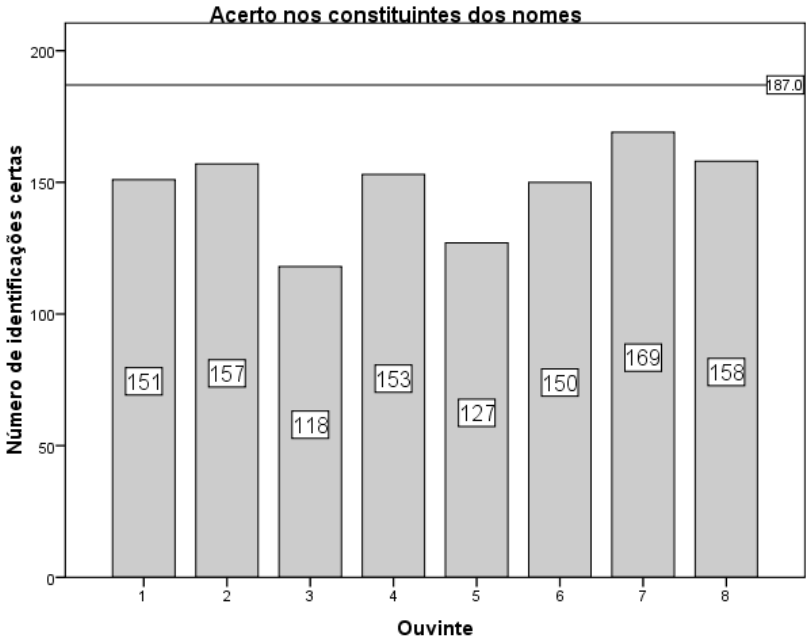

Figura 26 – Gráfico do número de nomes correctamente identificados.

Na seguinte tabela é apresentada a percentagem de nomes correctamente identificados por cada um dos ouvintes, do total das 187 palavras existentes.

	Ouvinte1	Ouvinte2	Ouvinte3	Ouvinte4	Ouvinte5	Ouvinte6	Ouvinte7	Ouvinte8
% De nomes correctamente identificados	80,7	84,0	63,1	81,8	67,9	80,2	90,2	84,5

Tabela 13 – Percentagens dos nomes correctamente identificados para cada um dos ouvintes.

Com estes resultados obteve-se um valor médio de 79,1% de nomes correctamente identificados.

Passámos agora a relatar os resultados relativos à percentagem de identificação de nomes completos sem qualquer erro de identificação.

O seguinte gráfico mostra o número de nomes completos totalmente identificados, por cada um dos ouvintes.

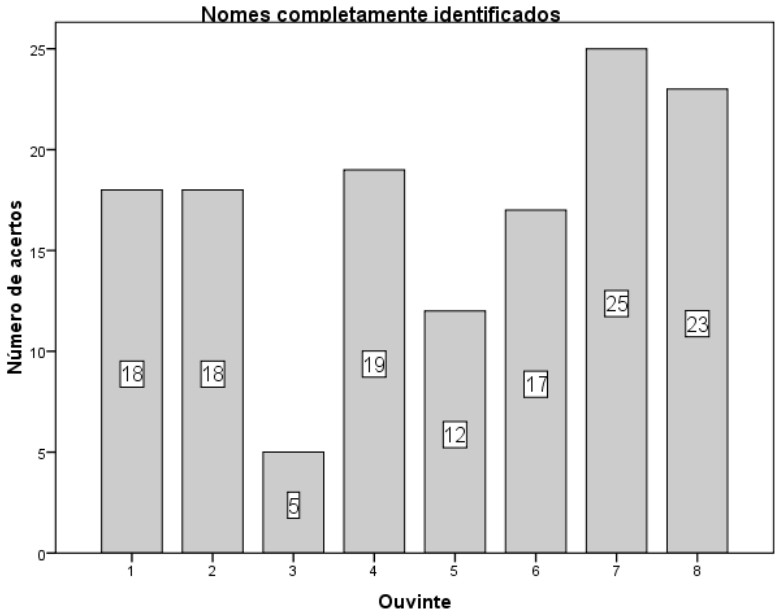

Figura 27 – Gráfico do número de nomes completos totalmente identificados.

Na tabela é apresentada a percentagem de nomes completos totalmente identificados por cada um dos ouvintes.

	Ouvinte1	Ouvinte2	Ouvinte3	Ouvinte4	Ouvinte5	Ouvinte6	Ouvinte7	Ouvinte8
% De nomes totalmente identificados	45,0	45,0	12,5	47,5	30	42,5	62,5	57,5

Tabela 14 – Percentagens dos nomes completos totalmente identificados para cada um dos ouvintes.

Passámos agora a discutir alguns dos factores que podem ter contribuído para estes resultados.

Na generalidade dos casos, todos os ouvintes falharam a identificação de nomes estrangeiros. O que, de algum modo faz sentido, já que a voz em questão foi programada apenas para a leitura de nomes em português.

A seguinte tabela apresenta todos os nomes estrangeiros existentes na lista dos quarenta nomes completos e o número de ouvintes que os identificaram correctamente.

Nomes	Stella	Peter	Damian	Francis	Stilwell	Sassetti	Calapez	Bettencourt
Nº de ouvintes a identificar correctamente o nome	7	0	2	3	2	1	1	2

Tabela 15 – Número de ouvintes que identificaram correctamente os nomes estrangeiros.

Como demonstra a tabela acima, os nomes estrangeiros contribuem para uma menor percentagem de nomes identificados e para uma menor percentagem de nomes completos identificados sem qualquer erro.

Os apelidos menos frequentes, no geral, também suscitam problemas. Casos de nomes como: Froufe, Balula, Xara, Tenorio, Valdés, Carreiro, Aguedo, Lélita, Cardim, apenas foram identificados correctamente, no máximo, por um ouvinte. Isto implica uma menor taxa de acerto nas duas métricas consideradas.

Outro factor que influencia as percentagens de identificação dos nomes está relacionado com o número de palavras que constitui um nome completo. Se o número de palavras for superior a cinco, há uma grande probabilidade de ocorrerem dificuldades na

identificação de nomes, pois os ouvintes têm a tendência de se concentrarem nas primeiras palavras, comprometendo a compreensão das seguintes.

Por fim, é também possível verificar que através da análise dos resultados, as pessoas de uma faixa etária superior aos 50 anos têm maior dificuldade na identificação dos nomes.

Apesar de todos estes factores, os nomes mais comuns foram, em geral, facilmente reconhecidos (e.g. José Sebastião Ramos Freitas ou Francisco José Costa Pereira, identificados correctamente por todos os ouvintes).

Capítulo 5 - Conclusão

Neste último capítulo, será apresentado um pequeno resumo do trabalho realizado, algumas conclusões relevantes, assim como algumas propostas para uma continuação futura deste trabalho.

5.1. Resumo do Trabalho Realizado

Motivado pela falta de trabalhos desenvolvidos na síntese de nomes em português, propôs-se com esta dissertação a criação de uma voz capaz de ler nomes próprios em português. Este trabalho poderá tornar viáveis, a médio prazo, aplicações, como por exemplo, a chamada automática dos utentes num hospital.

Para o processo de criação da voz, inicialmente, tentou-se a utilização exclusiva do sistema SPICE, para apoio à criação de novas vozes para Festival. O SPICE conta com um sistema de *upload* de dados e com algoritmos já mecanizados dentro da sua própria estrutura. Os algoritmos disponíveis são capazes de processar estes dados, de modo a construir os vários módulos de uma voz, de uma forma que se anuncia (segundo os seus criadores) fácil e rápida.

Foi necessária a criação de uma lista de nomes com um tamanho considerável; a construção de um ficheiro de *prompts* nos formatos e requisitos necessários à sua gravação; a definição de todos os fonemas da língua portuguesa, contornando-se as várias limitações do SPICE, no que diz respeito à ausência de representações para vogais, semi-vogais e ditongos nasais; a geração de um conjunto de regras LTS, assim como a criação de um dicionário.

Inicialmente, estas etapas foram desenvolvidas no SPICE de forma exploratória, pois pretendia-se, antes da criação da voz, de uma forma rigorosa conhecer todo o processo necessário para a sua geração.

Após a construção da primeira voz, sem grande rigor e conhecimento de todo o processo para a sua criação, passou-se ao passo seguinte: construir a voz com o máximo rigor, com o objectivo de obter uma voz com a melhor qualidade possível.

Inexplicavelmente[1], a construção de uma voz de qualidade, com o uso exclusivo do SPICE, deixou de ser possível, devido à incapacidade de envio de dados áudio para os seus servidores, o que inviabilizou a construção de alguns dos modelos de prosódia e a própria síntese *clustergen*, técnica usada no SPICE para a geração de ondas sonoras.

Foi necessário resolver este contratempo para que a construção da voz fosse possível. Assim, optou-se pelo uso de uma solução híbrida, onde se aproveitou a voz de má qualidade criada no SPICE como base e se substituiu a parte de geração do sinal pelo MBROLA, alterando-se grande parte dos ficheiros gerados pelo SPICE.

No que diz respeito aos modelos de duração, devido à falta de treino, usou-se uma lista de durações de fonemas provinda de uma outra voz já construída.

Para ser possível a utilização da síntese externa MBROLA, foi necessário a alteração de alguns ficheiros da voz criada pelo SPICE, de modo a bloquear a síntese *clustergen*.

Por fim, já com todo o processo prático concluído e a voz criada, usou-se o Festival, para a leitura de um conjunto de nomes, guardando-se o resultado da conversão grafema-fone e o resultado da síntese (em ficheiros .wav), para a avaliação do sistema criado.

A avaliação consistiu na realização de dois tipos de teste: um para avaliar a conversão grafema-fone; outro para avaliar a identificação dos nomes por um conjunto de 8 pessoas.

5.2. Principais Resultados e Conclusões

Como principal resultado desta dissertação, tem-se uma voz capaz de sintetizar nomes próprios em português, a partir da sua representação escrita.

Foi possível verificar, através da análise de resultados do teste de avaliação para a conversão grafema-fonema, que a voz criada tem uma taxa de acerto elevada (na conversão grafema-fonema). Os valores para esta taxa de acerto poderiam ser ainda melhores, caso

[1] Situação que não conseguimos ultrapassar apesar de várias tentativas de contacto aos responsáveis pelo SPICE.

fossem acrescentadas ou corrigidas algumas das regras de conversão grafema-fonema, por parte de um especialista em linguística. No entanto, não era esse o objectivo da dissertação, onde para além da criação de uma voz, foi, entre outras coisas, proposta a avaliação do sistema SPICE, neste caso particular, na geração de regras LTS.

Conclui-se também, que apesar dos resultados obtidos, nos testes de identificação de nomes, se considerarmos uma aplicação prática, onde apenas sejam lidos nomes em português (pois a voz foi criada apenas para o português), se tivermos em conta apenas nomes comuns (e.g. da chamada dos utentes de um hospital, onde que cada pessoa tem o seu nome como familiar) e se for feita mais do que uma repetição para a chamada de cada nome (evitando-se os problemas de identificação, em casos de nomes completos com muitas palavras), a utilização da voz criada pode apresentar bons resultados, sendo inteligível na grande maioria dos nomes portugueses comuns.

Através deste trabalho, foram adquiridos vários conhecimentos relacionados com a síntese de voz, nomeadamente: métodos de síntese, todos os componentes de um sistema TTS e as suas funções, a aprendizagem das funcionalidades e da utilização dos programas SPICE, Festival e MBROLA.

Por fim, pode-se concluir que o SPICE apresenta um conjunto de limitações para a língua portuguesa, não havendo espaço para a representação, no mapa de fonemas, de vogais nasais, semi-vogais e ditongos nasais. Este *software*, devido às sucessivas alterações na sua estrutura e ao seu constante desenvolvimento, nem sempre é de confiança para a realização de um trabalho. Como tivemos oportunidade de verificar, pode deixar de funcionar de forma inexplicável.

5.3. Sugestões de Continuação

Para a continuação deste trabalho, poderão ser melhorados os modelos de entoação, de duração e de fraseamento prosódico, de modo a tornarem o discurso sintetizado mais natural e inteligível.

Outro melhoramento a fazer consiste em tornar a voz capaz de ler nomes estrangeiros, possibilitando uma melhor compreensão destes e a aplicação da voz em sistema multi-língua.

Uma das formas de tornar o sistema multi-língua consistiria em criar um pré-processamento para adaptar a grafia dos nomes estrangeiros, de modo a que as regras de conversão grafema-fonema funcionassem melhor. Uma outra abordagem possível seria criar um dicionário de excepções com nomes estrangeiros devidamente transcritos, o que implicaria a aquisição de um enorme volume de dados.

Bibliografia

Livros

[1] Taylor, P.; "Text-to-Speech Synthesis"; University of Cambridge, capítulo 14 e 16, 2009.

Teses

[2] Paiva, S.M. P.; "Síntese por concatenação de variantes regionais: falar do Porto"; Tese de Mestrado, Universidade de Aveiro, 2005.

[3] Oliveira, C.A.M.; "Do Grafema ao Gesto Contributos Linguísticos para um Sistema de Base Articulatória", Tese de Doutoramento, Universidade de Aveiro, 2009.

[4] Rua, C.M.A.T.; "Ditongos orais no português europeu"; Tese de Mestrado, Universidade de Aveiro, 2005.

[5] Martins, C.A.D.; "Dynamic language modeling for European Portuguese"; Tese de Doutoramento, Universidade de Aveiro, 2008.

[6] Silva, D.F.M.B.M.; "Algoritmos de Processamento da Linguagem Natural para Sistemas de Conversão Texto-Fala em Português", Tese de Douturamento, Universidade da Coruña, 2008.

[7] Gómez, A.J.; "Adaptación del sistema Texto a Voz "Festival" al Catalán"; Universidad Politécnica de Cataluña, 2007.

[8] Pérez Martínez, A., Vicente Cabeza, O. ; "Diseño y creación de un corpus oral para su aplicación en el modelo de síntesis de voz Mbrola"; Universidad de Valladolid, 2002.

[9] Lemmetty, S.; “Review of Speech Synthesis Technology”; Helsinki University of Technology, 1999.

[10] Moulines, E.; "Algorithmes de codage et de modification des paramètres prosodiques pour la synthèse de la parole à partir du texte"; École National Supérieure des Télécommunications, 1990.

Artigos

[11] Trancoso, I., Viana, C.; “Issues in the pronunciation of proper names: the experience of the Onomastica Project” Lisbon, Portugal.

[12] Jannedy, S., Mobius, B.; “Name pronunciation in German text-to-speech synthesis”; USA,1997.

[13] Kominek, J., Schultz, T., Black, A. W.; “Voice Building from Insufficient Data Classroom Experiences with Web-based Language Development Tools”; 6th ISCA Workshop on Speech Synthesis, Bonn, Germany, August 2007.

[14] Black, A.W.; “CLUSTERGEN: A Statistical Parametric Synthesizer using Trajectory Modeling”; InterSpeech 2006, Pittsburgh, USA, September 2006.

[15] Dutoit, T., Leich, H.; “ Tetx-to-speech synthesis based on an MBE re-synthesis of the segments database”; Mons,Belgique.

[16] DUTOIT, T.; “An Introduction to Text-to-Speech Synthesis”; Kluwer Academic Publishers, 1996.

[17] Schultz, T., Black, A.W., Badaskar, S., Hornyak, M., and Kominek, J.; “SPICE: Web-Based Tools for Rapid Language Adaptation in Speech Processing Systems”, InterSpeech2007, Antwerp, Belgium, August 2007.

[18] Teixeira, A., Vaz, F.; "Europen Portuguese Nasal vowels: An EMMA study"; Eurospeech 2001, Alborg, Dinamarca, 2001.

[19] Teixeira, J.P., Barros, M.J., Freitas, D.; "Sistemas de conversão Texto-Fala"; 3º Congresso Luso-Moçambicano de Engenharia, Maputo, Moçambique, 2003.

[20] Montero, J.M., Córdoba, R., Macías-Guarasa, J., San-Segundo, R., Gutiérrez-Arriola, J., Pardo, J.P.; "Parameter selection for prosodic modelling in a restricted-domain spanish Text-to-speech system"; Madrid, Spain.

[21] Magri, A., Cukier-Blaj, S., Karman, D.,F., Camargo, Z.,A.; "Correlatos perceptivos e acústicos dos ajustes supraglóticos na disfonia"; Revista CEFAC - Saúde e Educação, vol.9, no. 4, pp. 512-518, Outubro/Dezembro 2007.

Páginas da internet

[22] Projecto MBROLA:
http://tcts.fpms.ac.be/synthesis/mbrola.html

[23] Manual do Festival:
http://festvox.org/docs/manual-1.4.3/festival_toc.html

[24] Web site para acesso ao SPICE:
http://plan.is.cs.cmu.edu/Spice/spice

[25] Informações sobre o SPICE:
MULTILING2006-Schultz-Spice[1].pdf

[26] Aula da Física da Fala e Audição:
http://www.ifi.unicamp.br/~knobel/f105/fono9b.pdf

[27] SAMPA computer readable phonetic alphabet:
http://www.phon.ucl.ac.uk/home/sampa/index.html

[28] Unit selection synthesis:
http://en.wikipedia.org/wiki/Speech_synthesis#Unit_selection_synthesis

[29] Domain-specific synthesis:
http://en.wikipedia.org/wiki/Speech_synthesis#Domain-specific_synthesis

[30] Capítulo 1 da cadeira de processamento de fala:
http://www.deetc.isel.ipl.pt/comunicacoesep/disciplinas/pdf/sebenta/pdf/introducao_1.pdf

[31] Da escrita à fala – da fala à escrita, documento de Isabel Trancoso, Luís Oliveira e João Neto:
http://www.l2f.inesc-id.pt/~jpn/artigos/Trancoso-escrita00.pdf

[32] A Teixeira "Material de apoio à disciplina de Processamento Digital de Voz" , DETI, Universidade de Aveiro, 2009:
http://www.ieeta.pt/~ajst/pdv/

[33] CSLU toolkit:
http://cslu.cse.ogi.edu/