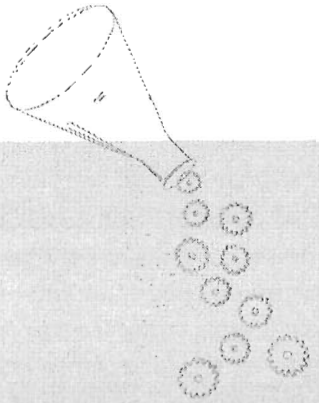


Sistema de Reconhecimento Automático de Voz (Parte 1)

1º Ten William George Siqueira Salles
1º Ten Dirceu Gonzaga da Silva
1º Ten Romualdo Begale Prudêncio

RESUMO

Este trabalho documenta o projeto de um sistema automático de reconhecimento de palavras isoladas. Procurou-se constituir um sistema *on line*, com características de sistema de tempo real, sem o uso de *hardware* adicional, além do necessário para digitalizar as amostras. Foi enfocado especialmente a implementação de algoritmos eficientes para a extração de características das palavras amostras e para o processo de decisão. As características escolhidas foram: doze coeficientes mel-cepstrum, doze coeficientes delta mel-cepstrum, o log-energia e o delta log-energia .O processo de decisão foi balizado pelo *Hidden Morkov Models*(HMM).

O sistema foi validado num teste de reconhecimento de palavras isoladas, independente de locutor, emum conjunto de dezessete palavras. Não foram feitos, no momento da aquisição dos dados, tratamentos filtragem de ruído nem de cancelamento de eco.

O trabalho será exposto em três partes: a primeira parte é composta de uma introdução histórica, de informações gerais sobre placa de som e sobre os ambientes de programação disponíveis para PC, e da documentação do projeto do sistema; a segunda parte versará sobre o conjunto de características utilizado; e a última parte descreverá o sistema de decisão.

INTRODUÇÃO

A necessidade de integração homem-máquina vem-se aumentando à medida que a tecnologia da informação se torna mais acessível ao leigo. Admite-se que seja uma barreira, para a maioria das pessoas, o desconhecimento do funcionamento das atuais interfaces homem-máquina, no tratamento de problemas cotidianos, como transações bancárias em caixas eletrônicos, troca de informações por código via telefone, entre outros. Este problema tem estimulado um conjunto de pesquisadores e tecnólogos a se dedicar ao desenvolvimento de métodos e tecnologias para que os equipamentos eletrônicos se habilitem ao meio de comunicação natural das pessoas: a fala.

Impulsionados por essa realidade os temas síntese e análise de voz saltam dos registros da ficção científica para as pranchetas dos homens sérios da ciência, ganhando força, fruto da necessidade urgente de manter a globalização e indiscriminação da informação.

A problemática resume-se então em fazer-se o homem ser entendido pela máquina, para que satisfaça seus desejos, e que a máquina responda ao homem de forma mais inteligível. Entretanto deve-se estar ciente que a análise de voz é sensível ao idioma, sendo importante o desenvolvimento de tecnologia nacional, voltada para a língua portuguesa, a fim de se ganhar autonomia nesta área de pesquisa.

HISTÓRICO DA INTRODUÇÃO DE COMPUTADORES NO TRATAMENTO DO SOM

Os computadores são evoluções das máquinas de calcular, e sempre buscaram efetuar cálculos mais rápido que o homem, para agilizar-lhe as tarefas. Desde 1957, nos primórdios da ciência da computação, já se registra interações do computador com a produção de música. *Lejarin Hiller* e *Leonard Isaacson*, baseado no trabalho de *Johann Joseph Fux*, que identificara regras para definir o estilo de computadores contrapontistas, criaram, no computador *Illiac* da Universidade de Illinois, um programa para gerar seqüência de notas aleatórias. Como um dos fragmentos da música produzida foi agradável aos ouvidos, considerou-se a produção da primeira peça musical por computador: a *Illiac Suite*.

Outro precursor da música eletrônica foi *Max Matheus*, que também em 1956, trabalhando com os computadores da *Bell Telephone Laboratories*, produziu sons musicais complexos. Apesar de ter desenvolvido uma linguagem orientada para a música, chamada MUSIC V, não conseguiu uma produção *on line*.

Ao final dos anos 60, *Matheus* desenvolveu o GROOVE: um programa que trouxe o domínio do tempo real a produção de música por computador. A técnica desenvolvida apelava aos dispositivos de síntese externos e ao processamento distribuído. Embora rústica, ela sobreviveu até os anos 80, influenciando inclusive o atual protocolo MIDI, que estabelece as interfaces entre os computadores e os instrumentos musicais eletrônicos.

A capacidade de cálculo fornecida pelos computadores foi utilizada com sucesso pelo compositor grego *Iannis Xerakis*, que concebia música com base em leis probabilísticas.

No final dos anos 70, com a crescente expansão dos computadores pessoais, a produção de música com auxílio das máquinas eletrônicas ganhou espaço entre o público mais modesto. Nomes como *Lynclawer*, *Fairlight*, *Mc Leyvier* e *Can Brio*, marcam esta fase.

Desde os primeiros microcomputadores, suas capacidades sonoras são exploradas: o *Commodore C 64* possuía um chip de síntese sonora conhecido por SID (*Sound Interface Device*); o *Tandy TRS-80* podia receber uma placa chamada *Orchestre-80*, que era programada por uma linguagem semelhante a GROOVE; o *Apple II*, que já contava em sua arquitetura com os acessos diretos a memória (DMA), teve placas de som desenvolvidas especialmente para ele, sendo a mais famosa a *Mountain Computer System*.

Com o aparecimento do protocolo MIDI, no início dos anos 80, que permitia o controle eficaz de instrumentos específicos de síntese musical, e a evolução da capacidade de cálculo destes instrumentos, o mercado de microcomputadores passou a se dedicar a síntese de sons para incrementar os jogos que disponibilizavam.

A redução dos custos das máquinas, aliado ao aparecimento de circuitos integrados para processamento digital de sinais (DSP) a baixo custo, como o *Motorola DSP 56001* e o *Texas Instruments TMS320*, possibilitaram a produção de máquinas dedicadas que suportavam *software* e *hardware* para o desenvolvimento musical. O maior representante desta geração foi o *NeXT* que era programado pela linguagem *Score*, padrão para notação musical introduzida por *Herland Smith*, em 1972.

O *NeXT* foi amplamente aceito nos circuitos acadêmicos devido a sua excelência para aplicações musicais, que proporcionou a troca e comparação de resultados, integrando a comunidade científica mundial, fomentando a produção de *software*, e enlaçando, definitivamente a ciência da Computação com o tratamento do som e da música.

Entretanto, fora do mundo acadêmico, eram os PC's que se espalharam entre os consumidores e se projetavam como padrão. Como o enfoque eram os *softwares* de escritório, por longo tempo os PC's ficaram desprovidos de capacidades sonoras adequadas, principalmente as solicitadas pelo mercado de jogos.

O primeiro esforço de integração dos PC's na música foi por volta dos meados dos anos 80, com o lançamento dos dispositivos MIDI MPU-401 da *Roland*, que se tornou padrão para PC's. Acompanhando este lançamento a IBM introduziu uma placa idêntica ao sintetizador *Yamaha FB-01*, mas que não respeitava o padrão MPU-401, não obtendo o sucesso esperado. Em 1977, a *Adlib* lançou *Personal Computer Music System*: uma placa de som de baixo custo que alcançou sucesso comercial devido a ansiedade dos produtores de jogos por essas novidades.

A versatilidade e o baixo custo dos microcomputadores propiciaram que os fabricantes de equipamentos profissionais os integrassem na produção musical, introduzindo no mercado placas e sistemas completos. Contudo o preço dos sistemas eram demasiados para tornarem-se populares. Somente no final de 1989 foi introduzido uma placa a baixo custo. A *Creative Technology* lançou uma placa compatível com a *Adlib*, com suporte para *joystik*, porta Midi, capacidade de síntese de voz, e gravação e reprodução de audiodigital. A placa chamada de *Sound Blaster* mantém-se até hoje como líder no mercado de placas de som.

O último grande passo para a estabilidade das placas de som foi o estabelecimento do MPC (*Multimedia Personal Computer*), através de um consórcio, promovido pela *Microsoft*, em que se deu forma a um conjunto de especificações de *hardware* dos PC's, que, em conjunto com o *Windows*, fornecem uma plataforma de baixo custo e com alta compatibilidade ao usuário.

ARQUITETURA DE UMA PLACA DE SOM

Na verdade, um placa de som é um conjunto integrado de periféricos e facilidades para o PC. Geralmente são constituídas por duas seções: sintetizador e processador de audiodigital. Elas podem gravar e reproduzir audiodigital simultaneamente, reproduzir CD áudio; gravar e reproduzir Midi, controlar equipamentos musicais externos, fazer interface com CD-ROM, além de oferecer um misturador que comanda várias entradas e saídas e um amplificador de áudio.

Como a intensidade do som é analógica e o computador é digital, as placas tem entre a entrada do microfone e o processador digital de sinais, um conversor analógico-digital (A/D); da mesma forma, para interfacear a saída do processador e a entrada dos auto-falantes há um conversor digital-analógico (D/A). A conversão analógico-digital pode ser feita em diversos formatos. Um formato que se popularizou foi o *Pulse Code Modulation (PCM)*. Nele as amostras são geradas diretamente a partir do sinal analógico, mantendo a eficiência em termos de largura de banda utilizada.

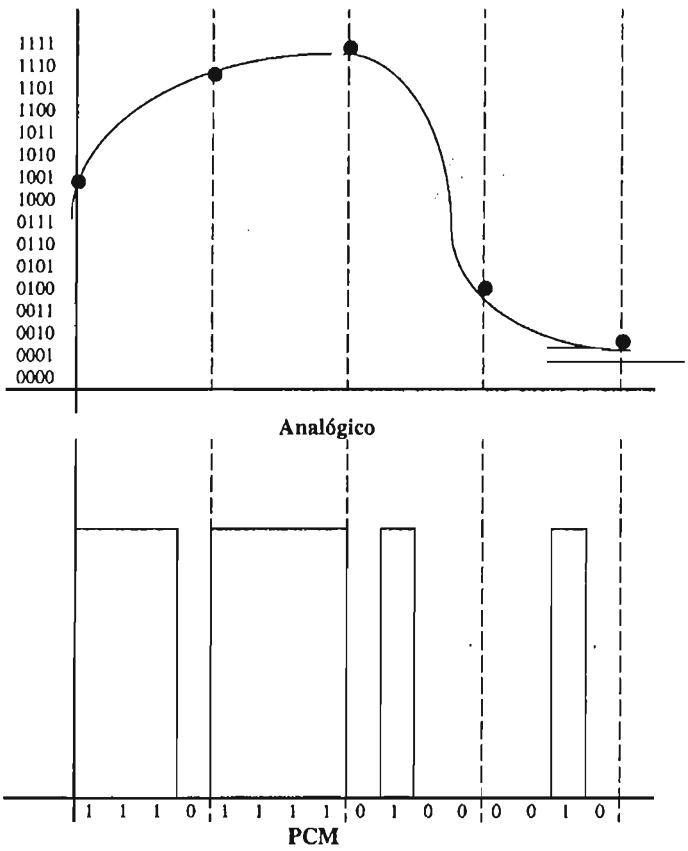
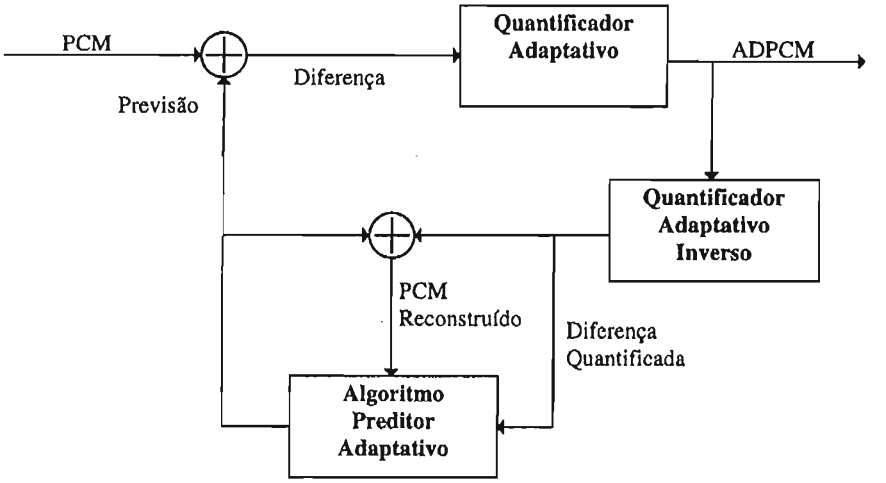


Fig 1 - *Pulse Code Modulation*

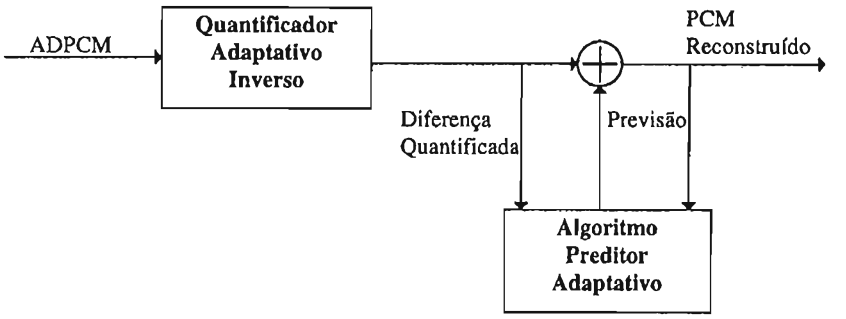
A representação do som em PCM é uma seqüência de números que representam a amplitude do sinal, tomada a intervalos constantes, com uma freqüência suficiente. A propósito da suficiência da freqüência aplicada cita-se o teorema da amostragem, enunciado por *Nyquist*:

"Qualquer forma de onda pode ser reconstituída a partir de amostras tomadas a uma cadência ou freqüência de 2 B igual ou superior ao dobro da largura de banda B do sinal, sendo B a diferença entre a componente de freqüência mais elevada e a mais baixa presente no sinal original".

Apesar das tecnologias de armazenamento e transmissão por meios eletrônicos estarem constantemente evoluindo, a necessidade conduziu o desenvolvimento de técnicas de compressão de áudio. A Lei - A, a Lei - m e o ADPCM são exemplos dessas tecnologias. Será apresentado o *Adptative Delta Pulse Code Modulation* (ADPCM) porque é uma técnica que trata o PCM já mencionado. O método consiste em usar, não os valores das amostras, mas as diferenças entre estas e uma previsão realizada com base num algoritmo específico. Esta técnica pode levar a compressão de 4:1 sem perda de qualidade. Abaixo estão os diagramas de bloco da compressão e descompressão.



Codificação ADPCM



Decodificação ADPCM

Fig 2 - Codificação e Decodificação por ADPCM

Os comentários sobre as técnicas de codificação e compressão dos sinais foram inseridos neste tópico para ilustrarem as necessidades de processamento que uma placa de som deve possuir, considerando que ela possui um processador especializado para o processamento digital dos sinais.

A placa de som escolhida para desenvolver este trabalho foi a *Sound Blaster* , devido a sua popularidade e bibliografia disponível. Ela disponibiliza a aquisição de dados em 11025, 22050, 44100 KHz, em modo estereofônico e monofônico e com amostras quantizadas por 16 ou 8 *bits*. Embora escolhida, sabe-se que não é a mais adequada pois seu *Digital Signal Processing* (DSP) não é específico para o processamento necessário ao reconhecimento da voz.

PLATAFORMAS DE DESENVOLVIMENTO

Embora já existam sistemas operacionais que implementam o reconhecimento automático da voz, como por exemplo a versão *4.0 do OS2/Warp* (Merlin), sabe-se que a grande fatia do mercado de PC's adotou o sistema operacional *Windows* rodando sobre o *DOS* ou simplesmente em sua versão mais nova: *Windows 95*.

Explorar-se-á as características destes dois sistemas, sem deixar de evidenciar a importância do Sistema *Unix*, que domina o campo das *workstations* e tem-se projetado com a explosão das redes. As redes exigem, além das interfaces mais "humanas", o desenvolvimento de aplicativos como o de videoconferência e de telefones de viva voz, correios eletrônicos e outros.

DOS - MONOTAREFA E MODO REAL

Quando as CPU's dos PC's estão rodando em modo real, tem-se acesso direto a todos os recursos de *hardware*. Além disso a alteração dos registradores da CPU e de seus periféricos controladores, durante a execução de um programa não é crítica, pois o sistema é monotarefa. Isso exige somente que, ao ser terminado o aplicativo, as antigas configurações estejam refeitas.

Os sistemas que exigem velocidade devem ser programados em uma linguagem com características de baixo nível, como o C ou Assembly, de forma a acessar o *hardware* diretamente. O Sistema de Reconhecimento Automático de Voz em foco foi implementado com essas características. Entretanto os fabricantes das placas de som fornecem d*rivers* que fazem a interface do *hardware*, com uma linguagem de auto nível. Essa nova camada de *software* aumenta o tempo de execução do aplicativo, que tem que fazer as chamadas às funções do d*river* para alterarem os registradores da placa. Como o programador não tem ingerência sobre essas funções o aplicativo também perde em flexibilidade. A vantagem de se utilizar estes *drivers* é que se tem um aumento na produtividade e não se faz necessário conhecer as nuanças da placa empregada. A *Creative Labs* fornece junto com os aplicativos que acompa-

nham sua placa de som um conjunto de *drivers*, que podem ser classificados em alto e baixo níveis. A figura abaixo mostra a hierarquia de *drivers* da *Sound Blaster*.

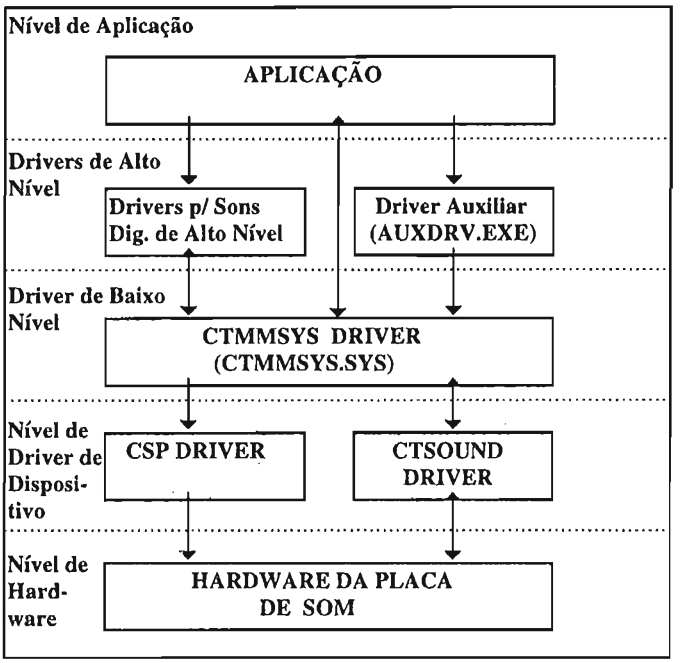


Fig 3 - Arquitetura dos *Drivers* da *Sound Blaster* 16.

A forma de se programar com *drivers* é descrita nos SDK's (*System Development Kit*) distribuídos pela *Creative Labs*.

WINDOWS - MULTITAREFA E MODO PROTEGIDO

Com a evolução do sistema operacional *Windows*, além de se tornarem disponíveis as interfaces gráficas com o usuário, foi introduzida a multitarefa. No *Windows* as tarefas podem ser carregadas simultaneamente e a CPU se dedicará a que estiver ativa. A comutação entre as tarefas pode ser feita rapidamente, sem prejuízo da aplicação. Para tal a CPU deve estar rodando em modo protegido, e neste caso o acesso direto ao hardware é restrito. O sistema operacional *Windows* oferece a cada aplicativo carregado uma máquina virtual onde rodar. Essa máquina não acessa um componente de *hardware* real; ela faz chamadas a um *driver* que virtualiza aquele componente. É um exemplo o caso do VPICD, que virtualiza o controlador de interrupções. Desta forma uma alteração neste controlador só é válida para aquela aplicação, permitindo que as outras tarefas continuem com eficiência.

Os fabricantes de placas também disponibilizam *drivers* que fazem chamadas aos *drivers* virtuais necessários para o funcionamento da placa. Além disso a própria API (*Application Programmers Interface*) do *Windows* oferece funções de alto e baixo níveis para manipulações de formatos de sons. Essas funções foram incorporadas as API's para implementar as especificações MPC (*Multimedia Personal Computer*) nível 1 e nível 2. O *Standard MPC* define uma plataforma que usuários e produtores de aplicações multimídia podem usar, assegurando um conjunto básico de facilidades genéricas.

Para usufruir ao máximo da plataforma MPC, o *Microsoft Windows* foi dotado de Extensões Multmídia, que suportam as três classes de serviços associados a áudio:

- serviços *Waveform* (audiodigital) permitem amostragem e reprodução de audiodigitalizado; são próprios para manipulação de áudio não musical.

- serviços CD-áudio suportam a reprodução de áudio contida em CD-ROM e CD-áudio convencional; sua qualidade é excelente.

- serviços Midi - suportam *Standard Midi Files* e gravação e reprodução através de sintetizadores internos e externos, isto é, funções de seqüenciadores.

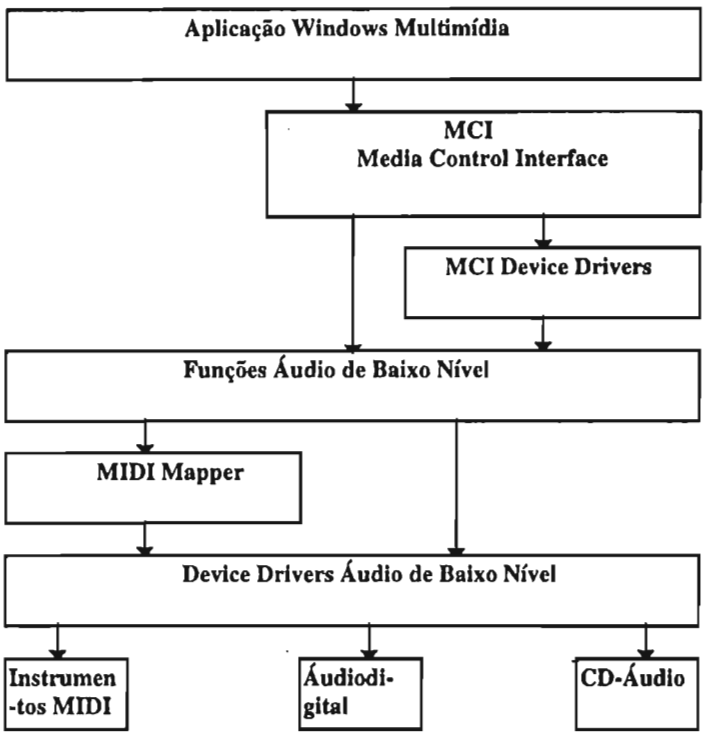


Fig 4 - Extensões Multimídia do *Windows*.

Do ponto de vista da programação, os serviços dividem-se em Alto e Baixo Nível. Para programação de Auto Nível as Extensões Multimídia estão dotadas de uma interface de controle chamada de MCI (*Media Control Interface*), que garante a aplicação e a virtualização dos *drivers* que a suportam. Quando se busca uma programação mais delicada, com maior controle de funcionalismo, tem-se no ambiente funções de baixo nível que interagem diretamente com os *devices drivers*. Para gravação e reprodução de arquivos no formato *Wave* estas funções são as *WaweIn** e *WaweOut**. Particularmente para este formato, a Extensões Multimídias estão preparadas para gravação e reprodução em 8 e 16 *bits* a taxas de 11025, 22050 e 44100 Hz, em mono ou estéreo; incluem os PCM's não lineares Lei - M e Lei - A e os ADMPC da *Microsoft*, *Yamaha* , *Creative Labs*, *OKI* e *Intel* (DVI).

DOCUMENTAÇÃO DO SISTEMA DE RECONHECIMENTO AUTOMÁTICO DE VOZ

Segundo a conceituação dada pela análise estruturada moderna, o ciclo de vida de um projeto clássico passa pelas seguintes fases: levantamento de requisitos, análise, projeto preliminar e estudo de *hardware*, projeto detalhado , codificação, teste dos subsistemas, integração e testes do sistema, operação e finalmente a manutenção. O levantamento de requisitos é essencial para a determinação das linhas gerais sob as quais deve ser estabelecida a análise. O objetivo de um sistema é satisfazer as necessidades do dono do sistema, portanto nesta fase deve-se estabelecer qual é a vontade do cliente. No caso do trabalho ora registrado as exigências eram gerais no sentido de se ter um sistema de reconhecimento automático de voz *on line* e com características de sistema de tempo real.

A fase de análise é independente de tecnologia de implementação. Dela extrai-se o projeto preliminar que é passível de modificação, já que ainda não se conhecem as características do *hardware* a ser utilizado. Há de se ressaltar que esta foi a fase mais demorada deste trabalho. Devido a gama de possibilidade para implementação do sistema, apresentadas anteriormente, o estudo de vários projetos preliminares foram feitos sendo escolhido como final aquele que atendeu plenamente os requisitos do Sistema e que pôde ser implementado no prazo especificado.

O resultado das fases de análises e projeto estão documentadas pelos os diagramas preceituados pela análise estruturada moderna.

DIAGRAMA DE CONTESTO (DC)

Este diagrama tem por finalidade mostrar os limites do sistema e suas interfaces com o mundo real. Como este sistema é muito simples e possui uma única entrada de dados e uma única saída, seu diagrama de contexto não carrega muitas informações.

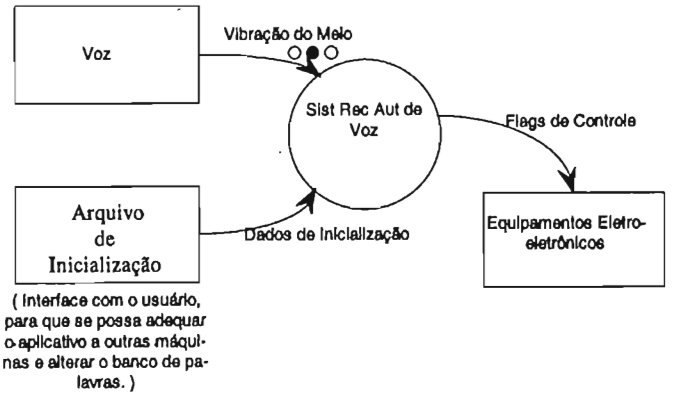


Fig 5 - Diagrama de Contexto do SRAV.

DIAGRAMAS DE FLUXO DE DADOS (DFD)

Esta ferramenta é muito útil nas modelagens de sistemas, principalmente para sistemas operativos, nos quais as funções são de fundamental importância e mais complexas do que os dados manipulados pelo sistema . O DFD é auto-explicativo, sendo que suas bolhas representam as ações sobre os dados, e as trajetórias que interligam as bolhas representam os fluxos de dados que entram e saem em cada função. O primeiro DFD, conhecido como DFD-0 é uma explosão do Diagrama de Contexto nos seus subsistemas principais. Os demais DFD's são explosões sucessivas das bolhas que devem ser esmiuçadas. A numeração das bolhas indica a seqüência das explosões. Note que, embora os DFD's estejam servindo para documentar o sistema eles são ferramentas da etapa de análise.

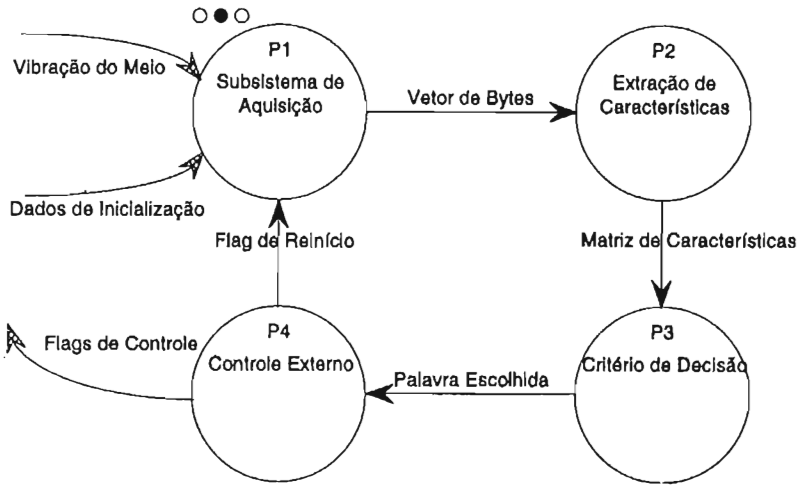


Fig 6 - Diagrama de Fluxo de Dados - 0

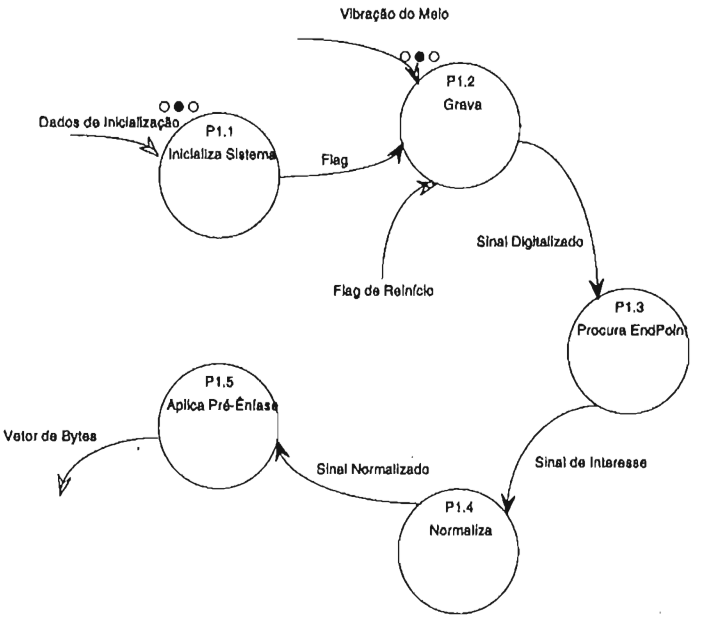


Fig 7 - DFD-1 (explosão da bolha 1)

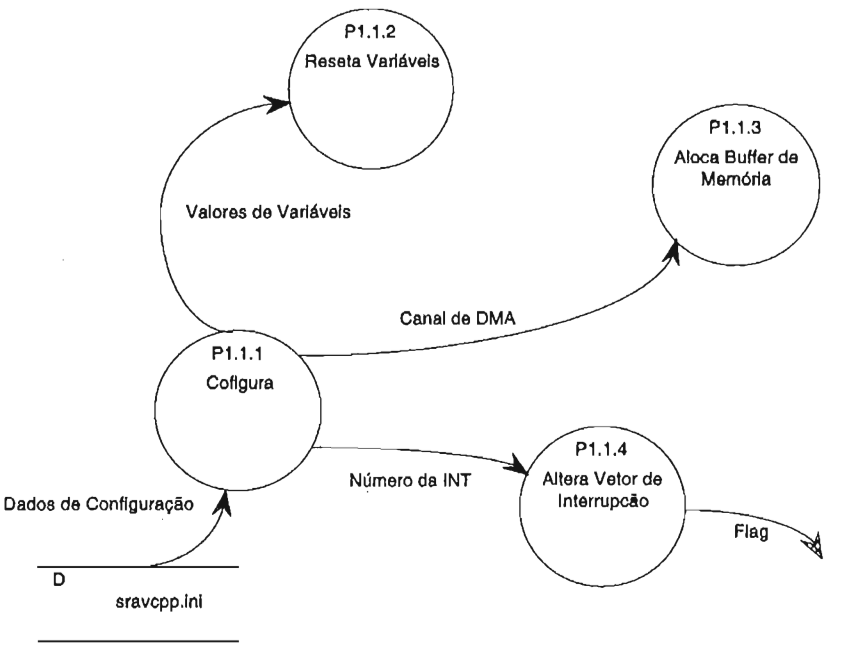


Fig 8 - DFD - 1.1 (explosão da bolha 1.1)

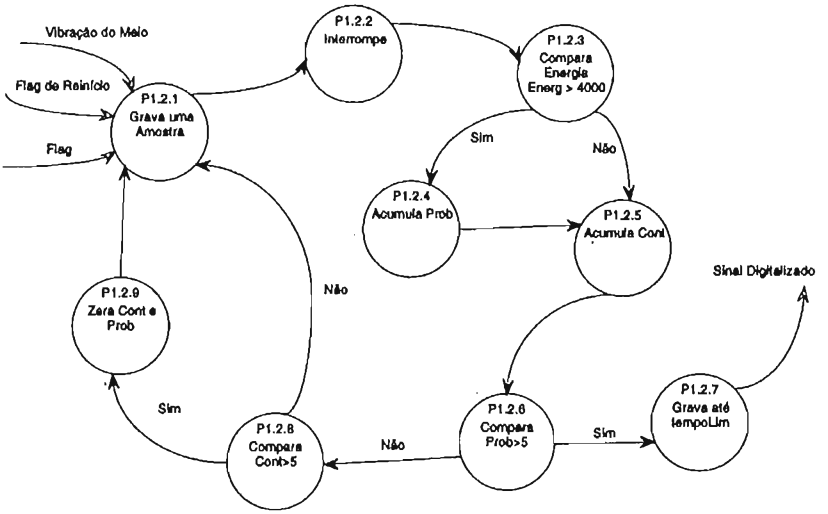


Fig 9 - DFD-1.2 (explosão da bolha 1.2)

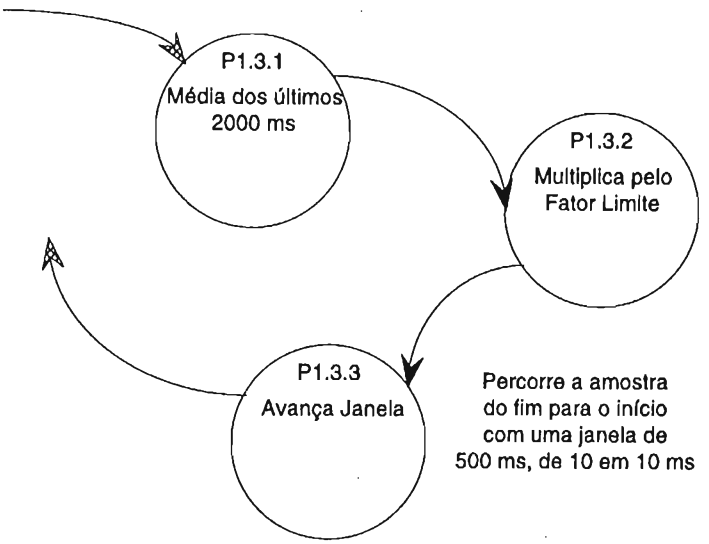


Fig 10 - DFD - 1.3 (explosão da bolha 1.3).

DIAGRAMA DE TRANSIÇÃO DE ESTADO (DTE)

O DTE é utilizado quando um determinado dado que circula nos sistema assume estados durante o sistema que caracterizam sua evolução no processo. O objetivo do DTE é tornar evidente esses estados e as condições e ações que aplicam as transições. No diagrama as caixas representam os estados e as setas as transições.

DIAGRAMA DE ESTRUTURA MODULAR (DEM)

Já entrando na fase de projeto, utiliza-se o DEM que visa converter as especificações mapeadas pelos DFD's em um suporte para a programação estruturada. Enquanto os DFD's são diagramas voltados para o entendimento do dono do sistema (cliente) o DEM é voltado para o implementador. Portanto na confecção do DEM tem-se a preocupação com o paradigma da linguagem, com a plataforma de desenvolvimento e com o *hardware* onde será aplicado. Neste diagrama cada caixa representa uma sugestão para uma função a ser implementada. As setas indicam a ordem de chamada das funções, e pode-se também indicar os parâmetros passados entres as funções.

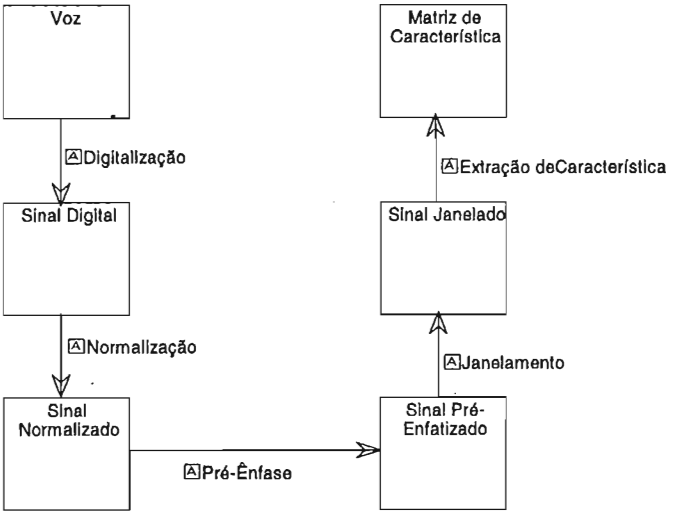


Fig 11 - Diagrama de Transição de Estado do Sinal de Voz.

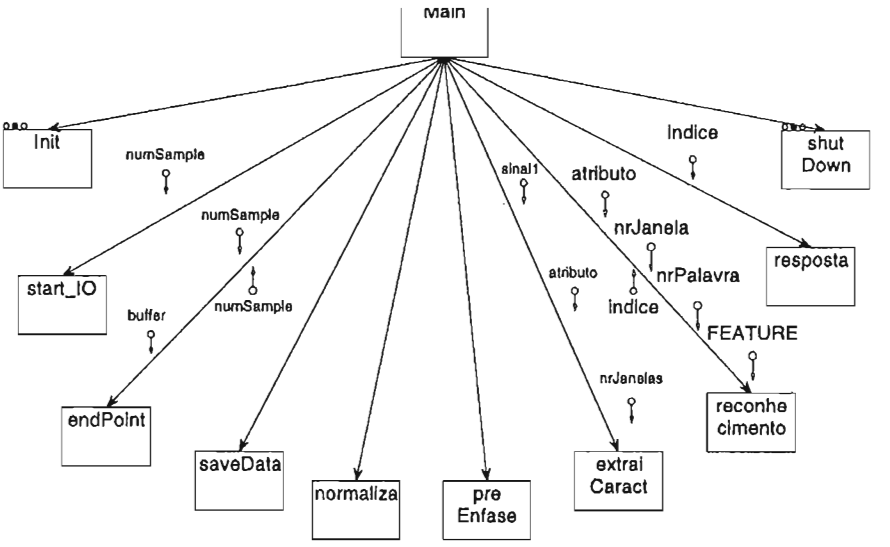


Fig 12 - Diagrama de Estrutura Modular para a Função *main*().

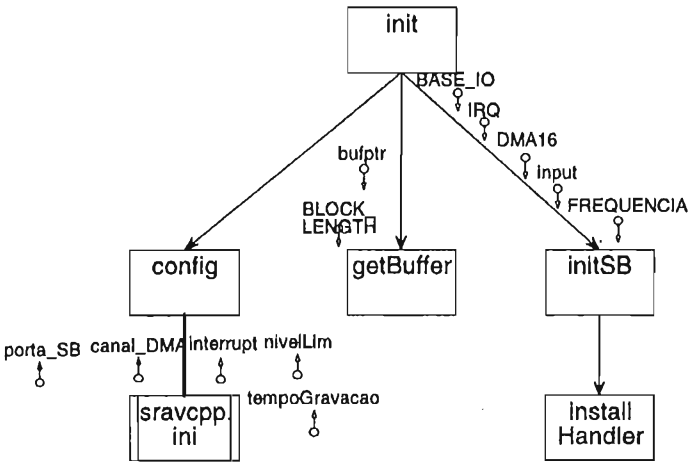


Fig 13 - Diagrama de Estrutura Modular para a Função *Init*().

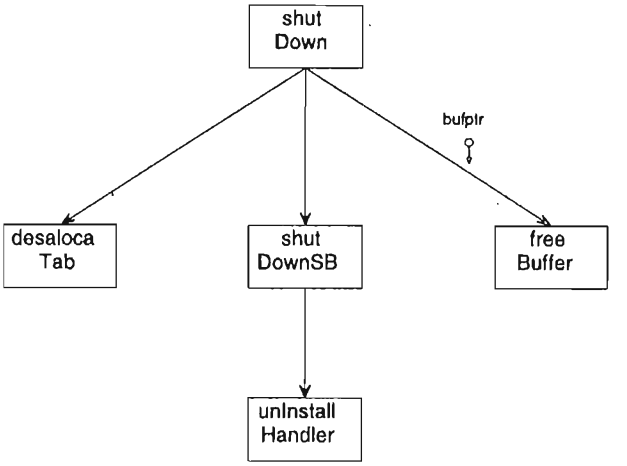


Fig 14 - Diagrama de Estrutura Modular para a Função *ShutDown*().

CONCLUSÃO PARCIAL

Outras ferramentas poderiam ser usadas para analisar, projetar e documentar um sistema. Por exemplo, os Diagramas de Entidade e Relacionamento, se os relacionamentos entre os dados fossem complexos; os Dicionários de Dados se a estrutura dos dados fosse complexa; os fluxogramas, entre outras. Poder-se-ia, também, acompanhar o projeto com um cronograma físico-financeiro, auxiliado por um planejamento da utilização do tempo e da identificação de caminhos críticos, como os diagramas PERT-CPM. No entanto, devido a flexibilidade dada ao projeto, e por se tratar de uma iniciação a pesquisa, não se exigiu tantos recursos, bastando a exploração das ferramentas anteriormente descritas. Esta parte do trabalho procurou balizar um caminho para que pretendem se aventurar na pesquisa de voz, no que toca os ambientes de trabalho, o *hardware* disponível no mercado, e a própria constituição de um sistema. Na segunda parte do artigo serão tratadas as características utilizadas no reconhecimento, e na última parte será apresentado o sistema de decisão que é baseado no HMM.