



Daniel Victor Montoril Negrão

Darus Cordeiro Pastorello

Maristela Aparecida Scheffer

Ronaldo Roberto Silva

Vagner Rodrigues

IMPLEMENTAÇÃO DE RECURSOS DE RECONHECIMENTO DA FALA EM
APLICAÇÃO PARA TESTES PSICOLÓGICOS BASEADO EM SISTEMAS
INTELIGENTES

Projeto Final apresentado como requisito á
obtenção do grau de Técnico em
Informática, curso de Tecnologia em
Informática, setor Escola Técnica, da
Universidade Federal do Paraná.

ME
005.1
N385

Prof. Mauro José Belli
Orientar

Prof. Roberto Tadeu Raittz
Co-Orientador

CURITIBA
2004

AGRADECIMENTOS

Ao Prof. Orientador Mauro José Belli pela confiança, incentivo e total apoio na realização deste trabalho.

Ao Prof. Co-Orientador Roberto Tadeu Raitz pela colaboração e partilha de seu conhecimento sobre Redes Neurais Artificiais, que foi crucial para o desenvolvimento de nosso projeto.

Ao Tiago Sak que nos ajudou na questão da programação da Rede FAN.

SUMÁRIO

LISTA DE ILUSTRAÇÕES.....	viii
LISTA DE TABELAS.....	xi
RESUMO.....	xii
ABSTRACT.....	xiii
1 INTRODUÇÃO.....	1
2 PROBLEMA.....	4
3 OBJETIVOS.....	4
3.1 OBJETIVO GERAL.....	4
3.2 OBEJETIVO ESPECÍFICO.....	5
4 SISTEMA FAN (FREE ASSOCIATIVE NEURONS).....	5
5 PROCEDIMENTOS PARA O RECONHECIMENTO.....	6
5.1 CAPTURA DO SINAL DA VOZ.....	7
5.2 SUPRESSÃO DO SINAL CAPTURADO.....	10
5.3 FORMATAÇÃO DO SINAL SUPRIMIDO.....	12
5.4 EXTRAÇÃO DAS CARACTERÍSTICAS.....	15
5.5 USO DA REDE FAN.....	19
6 LABORATÓRIO DE RECONHECIMENTO.....	24
6.1 CAPTURA DE VOZ.....	25
6.2 GRAVAR PADRÕES NO BANCO.....	28
6.3 GRAVAR REDE NO BANCO.....	29
6.4 REFINADOR.....	30

6.5 CONTADOR.....	32
7 PROTÓTIPO.....	33
7.1 INTRODUÇÃO.....	33
7.2 TESTE 2.1 PRODUÇÃO ORAL DE PALAVRAS.....	34
7.3 BASE DE DADOS.....	36
7.4 RECONHECIMENTO.....	38
8 LABORATÓRIO FAN.....	39
9 TAREFAS DESENVOLVIDAS.....	43
9.1 TAREFAS MÊS: MARÇO.....	43
9.2 TAREFAS MÊS: ABRIL.....	43
9.3 TAREFAS MÊS: MAIO.....	44
9.4 TAREFAS MÊS: JUNHO.....	44
9.5 TAREFAS MÊS: JULHO.....	44
9.7 TAREFAS MÊS: AGOSTO.....	45
9.8 TAREFAS MÊS: SETEMBRO.....	45
9.9 TAREFAS MÊS: OUTUBRO.....	45
9.10 TAREFAS MÊS: NOVEMBRO.....	46
9.11 TAREFAS MÊS: DESEMBRO.....	46
10 LIMITAÇÕES.....	47
10.1 MICROFONE.....	47
10.2 RUÍDO AMBIENTE.....	49
10.3 ERRO NA GRAVAÇÃO DOS PADRÕES.....	50
10.4 PROCESSO DE REFINAMENTO DOS DADOS.....	52

10.5 RUIDOS DO SISTEMA.....	52
10.6 MODO DE FALAR.....	53
10.7 SONS DIFÍCEIS.....	53
10.8. GRANDE NÚMERO DE LOCUTORES.....	54
11 FERRAMENTAS UTILIZADAS.....	54
11.1 BORLAND DELPHI 7.....	54
11.1.1 TAUDIO.....	55
11.1.2 ZEOS.....	56
11.1.3 FLATSTYLE.....	57
11.1.4 XPMENU.....	57
11.2 LABORATÓRIO FAN.....	58
11.3 MICROSOFT VISIO 2002 PROFESSIONAL.....	58
11.4 LTZ POSTGRESQL.....	59
11.5 MATLAB 6.5.....	59
12 MODELAGEM.....	60
13 CONCLUSÃO.....	65
13.1 RESULTADOS.....	66
13.2 TRABALHOS FUTUROS.....	67
14 REFERÊNCIAS BIBLIOGRÁFICAS.....	68
15 ANEXOS.....	70
15.1 ANEXO I MODELAGEM DO PROTÓTIPO.....	71
15.2 ANEXO II MODELAGEM DO LABORATÓRIO DE RECONHECIMENTO.....	123

15.3 ANEXO III DICIONARIO DE DADOS.....	144
15.4 ANEXO IV CÓDIGO FONTE DO PROTÓTIPO.....	202
15.5 ANEXO V CÓDIGO FONTE DO LABORATÓRIO DE RECONHECIMENTO.....	238

LISTA DE ILUSTRAÇÕES

Figura 1.1 - Cartela 1 do Teste 2.1 (Produção Oral de Palavras), em destaque a palavra Bode.....	2
Figura 5.1.1 - Mostra 40 posições de uma parte do vetor do sinal de som capturado.....	8
Figura 5.1.2 - Gráfico da palavra Bode capturada a frequência de 22050 Hz, no eixo Y as amplitudes variando de 0 a 255 e no eixo X o número de pontos, no caso 32768 pontos.....	9
Figura 5.1.3 Mostra o gráfico da palavra Bode capturada a frequência de 44100 Hz, nota-se o aumento do número de pontos em relação à figura acima, agora com 57344 pontos.....	9
Figura 5.2.1 Mostra o início e fim da palavra, no caso apenas as amplitudes entre os pontos de início e fim serão utilizados.....	10
Figura 5.2.2 Mostra o sinal suprimido, ou seja, sem o silêncio anterior e posterior à palavra capturada, nota-se a diferença do número de pontos em relação à figura acima.....	11
Figura 5.3.1 Mostra o gráfico do mesmo sinal capturado, agora ordenado pelas suas amplitudes.....	13
Figura 5.3.2 Mostra o ponto de corte, onde acima dele todos os valores passam a ser 1 e abaixo passam a ser 0.....	13
Figura 5.3.3 mostra vetores, binário e ponto de mudança sucessivamente.....	15

Figura 5.4.1 Mostra o vetores, digitalizado com 0 e 1, os pontos de mudança, e o último mostra o vetor Diferença sucessivamente, nota-se a relação entre eles.....	16
Figura 5.4.2 Gráfico das características da palavra Bode.....	17
Figura 5.4.3 Gráfico das características da palavra Bote.....	17
Figura 5.4.4 Gráfico das características da palavra Gato.....	17
Figura 5.4.5 Gráfico das características da palavra Gado.....	18
Figura 5.4.6 Gráfico das características da palavra Vaca.....	18
Figura 5.4.7 Gráfico das características da palavra Faca.....	18
Figura 5.5.1 Mostra Tela de captura do Laboratório de Reconhecimento com as palavras em ordem de pertinência.....	20
Figura 5.5.2 Gráfico de comparação entre as palavras Bode em amarelo e Bote em azul.....	22
Figura 5.5.3 Gráfico de comparação entre as palavras Gado em branco e Gato em marrom.....	23
Figura 5.5.4 Gráfico de comparação entre as palavras Vaca em vermelho e Faca em roxo.....	23
Figura 6.1 Tela Principal do Laboratório.....	25
Figura 6.1.1 Tela de captura do Laboratório de Reconhecimento.....	28
Figura 6.2.1 Tela de Gravar Padrões no Banco.....	29
Figura 6.3.1 Tela de Gravar Rede no Banco.....	30
Figura 6.4.1 Tela do Refinador sem dados.....	31
Figura 6.4.2 Tela do Refinador com dados.....	31

6.5.1 Tela do Contador de Padrões.....	32
Figura 7.2.1 Primeira cartela do teste 2.1 Produção Oral de Palavras.....	35
Figura 8.1.1 Tela principal do LabFAN com os arquivos de treinamento e teste carregados.....	39
Figura 8.1.2 Configuração dos parâmetros de treinamento.....	40
Figura 8.1.3 Tela de treinamento.....	41
Figura 8.1.4 Figura mostrando a tabela de confusão.....	42
Figura 10.1.1 Ruído.....	48
Figura 10.1.2 Sem Ruído.....	48
Figura 10.1.3 controlador de som Windows XP.....	49
Figura 10.3.1 mostra Laboratório de Reconhecimento identificando a palavra falada pela pessoa.....	50
Figura 10.3.2 mostra tela de reconhecimento e tela(Palavra Falada) das seis palavras.....	51
Figura 10.7.1 mostra palavra Faca com distância de 2 cm da boca ao microfone.....	53
Figura 10.7.2 mostra a palavra Faca falada perto do microfone.....	53
Figura 11.1.1.1 mostra Object Inspector do Delphi 7.....	56
Figura 11.1.2.1 Paleta do Zeos no Delphi 7.....	57
Figura 11.1.3.1 paleta FlatStyle no Delphi 7.....	57
Figura 11.1.4.1 mostra paleta do XPMenu no Delphi 7.....	58

LISTA DE TABELAS

Tabela 7.1.1 – Bateria Completa de testes do Protótipo.....	34
Tabela 16.1.1 Resultados dos testes realizados.....	66

RESUMO

Este trabalho tem pôr objetivo implementar em um protótipo para testes psicológicos, em desenvolvimento pelo orientador do projeto (Mauro J. Belli), o recurso de reconhecimento de voz em um universo definido de palavras (bode, bote, gato, gado, vaca, faca), que fazem parte da primeira cartela do teste 2.1 produção oral de palavras, que consiste em 15 cartelas, cada uma contendo 6 figuras. Durante a apresentação deste trabalho iremos nos referir como Protótipo a aplicação que contém a bateria de testes que foi apresentado como trabalho de conclusão de curso de pós-graduação de Mauro José Belli, e Laboratório de Reconhecimento a aplicação a qual desenvolvemos para auxiliar no desenvolvimento do reconhecimento das palavras definidas. Utilizaremos a RNA (Redes Neurais Artificiais) para desenvolvimento da rede de reconhecimento de palavras em nosso trabalho.

Devemos lembrar que houve desistência por parte da integrante Maristela Aparecida Scheffer no início do projeto.

ABSTRACT

The objective of this project is to implement in a prototype for psychology tests, being developed by Mauro J. Belli (the teacher who orientated us through the development of this project), the voice recognition resource in a limited universe of words (bode, bote, gato, gado, vaca, faca), that is contained in the first chart of the test 2.1 oral production of words, which consists in 15 charts, each one containing 6 pictures. During the presentation of this work we will refer as Prototype, the application that contains the battery of tests which were presented as a Master Degree final project from Mauro Jose Belli; and Laboratory of Recognition, the application that we built to assist the development of the recognition of the definite words. We will use the Artificial Neural Network (ANN) to develop the net of recognition of words in our project.

We must remember that it had desistance on the part of the integrant Maristela Aparecida Scheffer in the beginning of the project.

1 INTRODUÇÃO

O protótipo é uma aplicação para testes psicológicos baseados em sistemas inteligentes, para identificação de causas de problemas de aprendizagem de crianças. A aplicação tem uma bateria de testes no qual objetivo é detectar sintomas mais evidentes sobre desvios da recepção oral e escrita, e respectiva produção. O recurso de reconhecimento de palavras foi implementado apenas na Cartela 1 do Teste 2.1 que trata da Produção Oral de Palavras que contém quinze cartelas com seis figuras cada. O teste procede da seguinte forma, uma figura entre as seis da cartela será destacada e a pessoa deverá identificar a figura e falar ao microfone qual a figura em destaque, e o Protótipo deverá reconhecer se a palavra falada corresponde a figura em destaque, assim sucessivamente para as demais figuras .

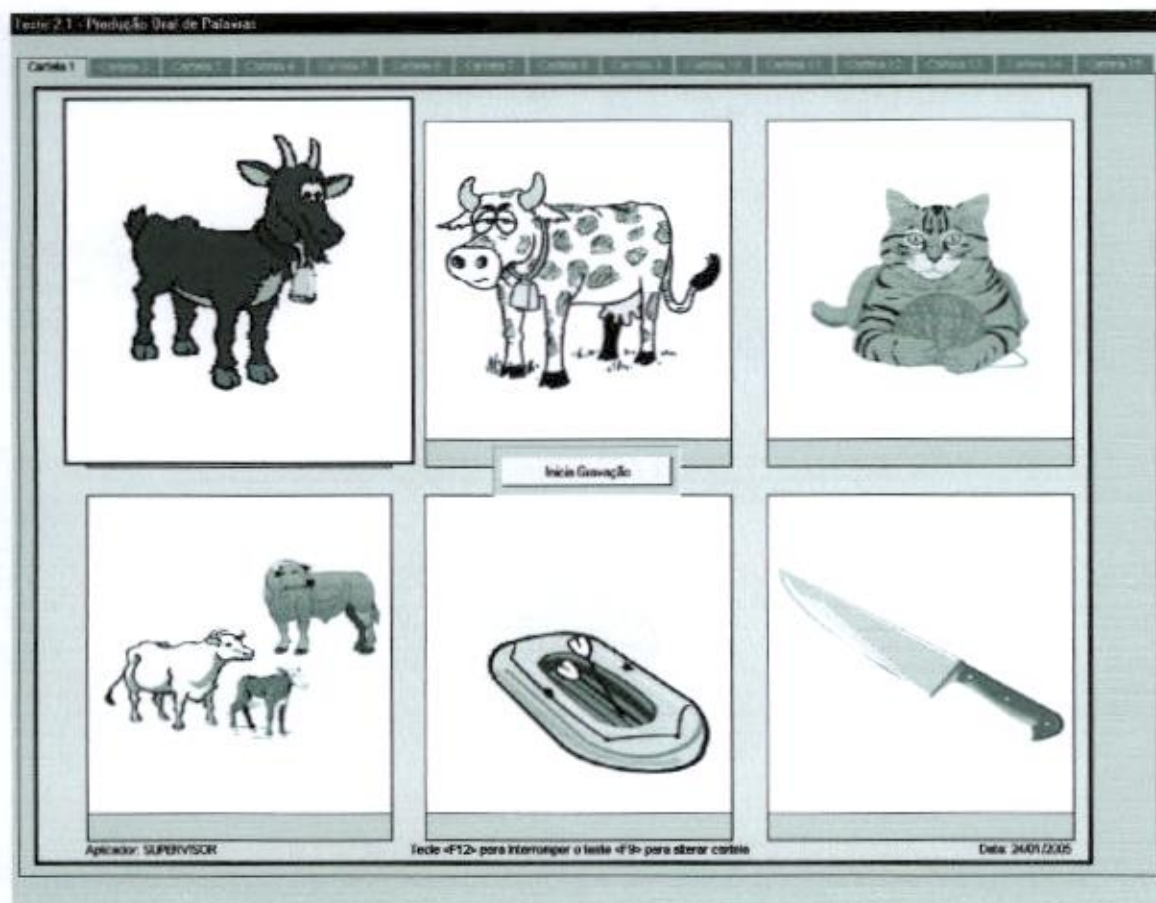


Figura 1.1 Cartela 1 do Teste 2.1 (Produção Oral de Palavras), em destaque a palavra Bode.

A aplicação construída pela equipe, a qual nomeamos de Laboratório de Reconhecimento, foi onde todo projeto foi desenvolvido. Através do Laboratório de Reconhecimento que dispomos de ferramentas para auxiliar no desenvolvimento de nosso trabalho e capturamos a palavra falada pela pessoa e assim estudamos o melhor método para chegar ao reconhecimento da palavra.

Utilizamos em nossa aplicação a RNA (Redes Neurais Artificiais), um dos nichos da Inteligência Aplicada, que propõe uma metodologia para solucionar problemas relacionados à área de inteligência artificial, modelando sistemas através

de conexões que possam simular o sistema nervoso humano, abrangendo a capacidade que mesmo possui de aprender, agir perante as mais diversas situações.

A primeira etapa do reconhecimento consiste em realizar a captura do sinal de voz através de uma interface analógica, por exemplo, uma placa de som, consistindo basicamente em entrarmos com a informação (voz) de forma analógica e trabalharmos essa informação de forma digital. A segunda etapa (pré-processamento) procura extrair do sinal capturado as características que descrevem adequadamente o sinal de voz. A terceira etapa consiste em identificar os dados selecionados na fase do pré-processamento e utilizá-los para o treinamento do sistema (treinamento da rede). A quarta e última etapa seria utilizar a rede já treinada para reconhecer a palavra que foi falada (reconhecimento).

O que chamamos de "rede" seria a Rede Neural, ou seja, a parte da aplicação que "pensa", há vários tipos de redes neurais, a rede que utilizamos foi o Sistema FAN (*Free Associative Neurons*), que é um exemplo de sistema híbrido, sua base são as redes conexionistas, modelagem difusa e representação de padrões.

O sistema FAN basicamente associa um conjunto de características a determinadas classes pré-definidas, no nosso caso haverá seis classes, que serão as seis palavras (bode, bote, gato, gado, vaca, faca), e são extraídas sete características de cada palavra que no decorrer do trabalho será bem exemplificado como é feito todo o processo de extração dessas sete características até o uso do FAN para o reconhecimento.

2 PROBLEMA

O reconhecimento de voz não é novidade. Já existem soluções tecnológicas há bastante tempo, mas a maioria das tentativas não tem dado certo.

A razão é simples: é difícil interpretar a fala humana.

O maior problema no reconhecimento da voz está em identificar as diversas limitações da fala humana, ou seja, regionalismo, timbres, sotaques. A dificuldade do nosso projeto está em detectar os traços fonéticos que diferenciam os vocábulos da língua portuguesa, no português do Brasil, dentre as palavras bode, bote, gato, gado, vaca, faca, que são palavras foneticamente muito parecidas, como vaca e faca, por exemplo.

Há ainda o fato de se tratar de uma área ainda em expansão, que seria a Inteligência Aplicada, que mesmo sem ter alto nível de conhecimento sobre a área a equipe se dispôs a enfrentar esse desafio, por esse motivo o auxílio dos professores especialistas foi indispensável.

3 OBJETIVOS

3.1 OBJETIVO GERAL

Nosso trabalho tem por objetivo implementar no Protótipo em desenvolvimento o recurso de reconhecimento de palavras, recurso que será implementado na primeira cartela do Teste 2.1 (Produção Oral de Palavras), que deverá identificar se a palavra falada condiz com a figura em destaque, deverá apontar certo ou errado

dentre as seis palavras, no caso, bode, bote, gato, gado, vaca, faca. Essa informação (certo ou errado) é gravada em uma tabela de resultados do Protótipo, que posteriormente é usada para fazer o diagnóstico do problema de aprendizado da criança testada.

3.2 OBJETIVOS ESPECÍFICOS

- Conseguir uma porcentagem de acerto entre 67 por cento (quatro das seis palavras) a 100 por cento (seis das seis palavras).
- O reconhecimento deverá ser independente de locutor, ou seja, deverá reconhecer a voz de qualquer pessoa.
- O tempo de processamento a partir da captura até a resposta do sistema (reconhecimento) não deverá ultrapassar 3 segundos.

4 O SISTEMA FAN (FREE ASSOCIATIVE NEURONS)

A escolha do sistema FAN como arquitetura da Rede Neural se deve ao fato da sua facilidade de adaptação para aplicações em reconhecimento de padrões. No Reconhecimento de Padrões, a unidade FAN de reconhecimento é associada a uma classe específica. Todas as relações entre as características de uma determinada classe são estabelecidas unicamente na unidade que a representa, livre portanto das relações diretas com as demais classes do domínio do problema.

Uma unidade FAN é um esquema de combinações de características, associado a uma classe, e um conjunto de parâmetros resultantes da utilização desse esquema no seu treinamento.

Ao ser apresentado um novo padrão de treinamento, o FAN (unidade) pode ser treinado para reconhecê-lo/esquecê-lo, através dos reforços/enfraquecimentos nos seus parâmetros. No teste de um padrão, o FAN permite atribuir seu grau de pertinência em relação à classe que representa.

O processo de aprendizagem de uma rede FAN, para o reconhecimento de várias classes, é realizado treinando-se cada unidade com os padrões relacionados à classe que representa. O conjunto de todas as unidades (neurônios) do sistema FAN é chamado "rede", pois as interligações entre os neurônios cria uma rede, daí o nome "Rede Neural".

A discriminação da classe à qual um determinado padrão deve estar relacionado em um teste é feita através da escolha da classe cuja unidade FAN correspondente indicar maior grau de pertinência (similaridade sobre um aspecto) para o referido padrão.

Com base nisso podemos através das sete características extraídas de cada palavra, associar esse conjunto de características a uma classe (palavra), e assim conseguir reconhecer a voz.

5 PROCEDIMENTOS PARA O RECONHECIMENTO

Para se chegar ao propósito de reconhecer a voz, é necessário uma série de passos, o primeiro seria a captura da voz, que é feita através de um microfone, a voz

é então passada para a forma analógica e com isso é possível manipulá-la, a partir daí começa-se o processo para se chegar a uma resposta do sistema (reconhecimento).

O próximo passo é suprimir o sinal capturado, o que chamamos de supressão nada mais é do que retirar o silêncio anterior e posterior à palavra falada, para que apenas a parte que realmente representa a palavra seja processada.

Em cima do sinal suprimido é feito todo o processamento que chamamos de formatação do sinal, com essa formatação são extraídas algumas variáveis, novos vetores são criados e manipulados para serem usados na extração das características que seria o próximo passo, esse conjunto de características é então usado para treinar a rede FAN, com isso a rede "aprende" o padrão de cada palavra, com a rede treinada é possível reconhecer a palavra falada extraindo as mesmas características usadas no treinamento da rede, assim a rede retorna um conjunto de valores, que seria o grau de pertinência, a palavra que tiver o maior grau de pertinência seria a palavra que a rede apontou como certa.

5.1 CAPTURA DO SINAL DA VOZ

Para capturar a voz é necessário apenas um microfone conectado à uma placa de som, a passagem do sinal capturado pela placa de som para a aplicação (Protótipo ou Laboratório de Reconhecimento) é feita através do componente TAudio, que grava as amplitudes do sinal em um "Memory Stream", que nada mais é que um vetor de memória dinâmica, ou seja, o som passa a ser um vetor, em que cada ponto do vetor representa uma amplitude (Figura 5.1.1).

67 33 6 0 0 0 0 6 12 24 47 78 112 139 159 176 187 197 207 213 215 214 210 204 196 188 179 171 165 161 159 158 157 154 149 143 136 129 122

Figura 5.1.1 Mostra 40 posições de uma parte do vetor do sinal de som capturado.

A amplitude do sinal no eixo y varia de 0 a 255, onde a metade dessa amplitude, aproximadamente 127, representa ausência de som, e as extremidades seriam o máximo valor possível (Figura 5.1.2), pois o som é uma onda, por esse motivo ele varia de um valor positivo para um valor negativo, vendo a amplitude 127 como 0, a amplitude iria de -127 a $+127$. Como o sinal da voz passa a ser um vetor, o número de pontos desse vetor depende do tempo de gravação da voz e da frequência que o sinal está sendo gravado, a frequência utilizada pela aplicação para a captura é de 22050 Hz, dessa forma 1 segundo de gravação resulta em um vetor de 22050 pontos. A escolha dessa frequência se deve ao fato de que a escolha de uma frequência menor não representaria de forma exata o sinal capturado depois de todas as modificações realizadas posteriormente, ou seja, o baixo número de pontos poderia comprometer a integridade do sinal, e a escolha de uma frequência maior resultaria num aumento no tempo de processamento pela aplicação, pois há uma série de modificações que são realizadas, por esse motivo haveria muito mais dados a serem processados deixando todo o processo muito lento (Figura 5.1.3).

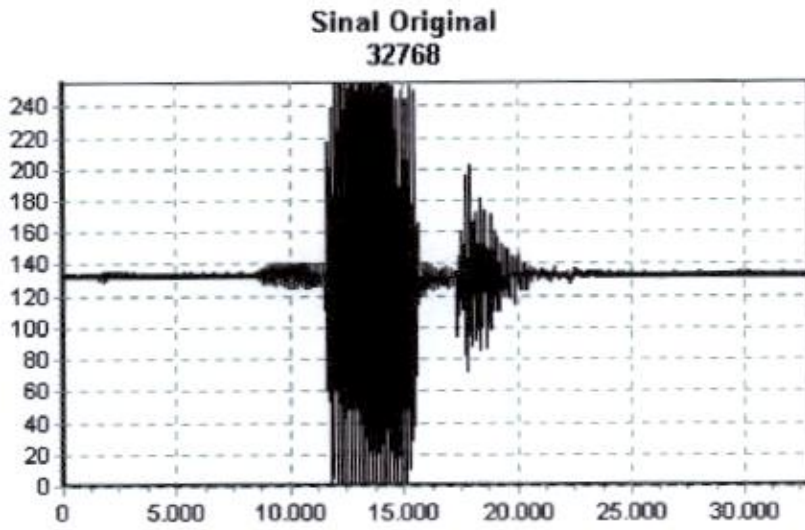


Figura 5.1.2 - Gráfico da palavra Bode capturada a frequência de 22050 Hz, no eixo Y as amplitudes variando de 0 a 255 e no eixo X o número de pontos, no caso 32768 pontos.

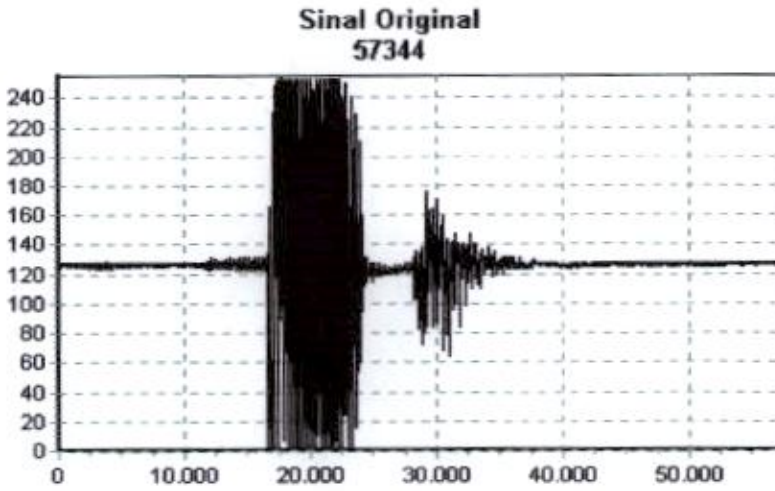


Figura 5.1.3 Mostra o gráfico da palavra Bode capturada a frequência de 44100 Hz, nota-se o aumento do número de pontos em relação à figura acima, agora com 57344 pontos.

5.2 SUPRESSÃO DO SINAL CAPTURADO

A supressão do sinal, como já dito anteriormente, nada mais é do que remover o silêncio anterior e posterior à palavra falada, ou seja, remover a porção que não faz parte da palavra, para processar apenas os pontos do vetor que realmente representam a palavra (Figura 5.2.1).

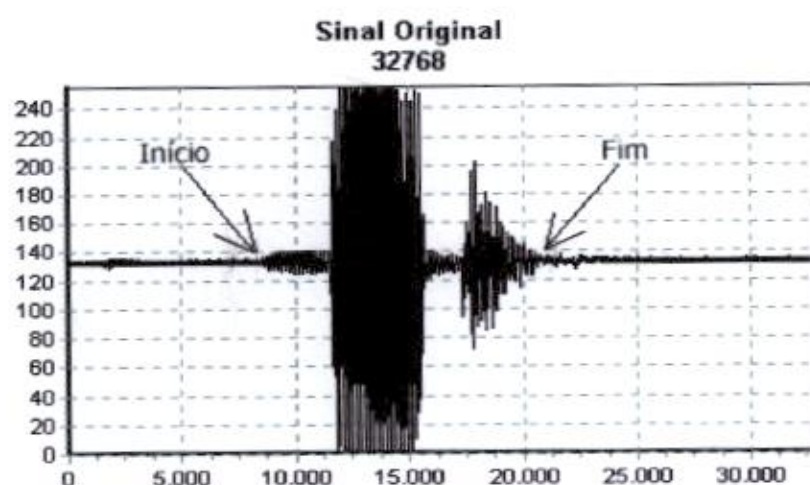


Figura 5.2.1 Mostra o início e fim da palavra, no caso apenas as amplitudes entre os pontos de início e fim serão utilizados.

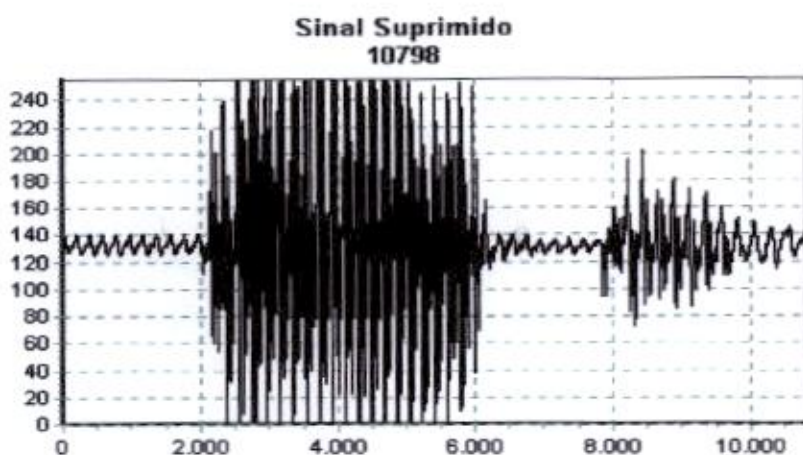


Figura 5.2.2 Mostra o sinal suprimido, ou seja, sem o silêncio anterior e posterior à palavra capturada, nota-se a diferença do número de pontos em relação à figura acima.

A supressão ocorre da seguinte forma, é retirada a média das amplitudes de todo o vetor do sinal capturado (média da palavra), começa -se varrendo o vetor a partir do ponto zero até encontrar uma amplitude que seja 5% acima da média da palavra, quando esse ponto é encontrado ele é marcado como sendo o início da palavra, mas antes é feita uma confirmação se realmente a palavra começou ou é apenas algum ruído, para isso é retirada a média dos 20 pontos anteriores ao ponto encontrado e compara-se com a média dos 20 pontos posteriores, se a média posterior for maior que a média anterior então significa que a palavra realmente começou, mas se a média posterior for menor que a média anterior então continua-se varrendo o vetor a partir do ponto marcado como início da palavra até encontrar novamente outro ponto 5% maior que a média da palavra até se confirmar o ponto de início da palavra. O mesmo procedimento é realizado para se achar o final da palavra, apenas indo de trás para frente, começando do último ponto do vetor indo em direção ao primeiro ponto do vetor até se encontrar uma amplitude 5% maior que

a média da palavra, e aplica-se o mesmo procedimento para verificação se é apenas algum ruído ou se realmente a palavra começou (terminou, pois está se fazendo o processo inverso, do final para o começo), comparando as médias dos 20 pontos anteriores e posteriores.

5.3 FORMATAÇÃO DO SINAL SUPRIMIDO

Com os pontos de início e fim da palavra definidos, começa-se a formatação do sinal suprimido, mexendo apenas com os valores entre os pontos de início e fim da palavra, os outros pontos são descartados, ou seja, se mantém o vetor de amplitudes excluindo os valores antes do ponto marcado como início da palavra e os valores depois do ponto marcado como final da palavra. A primeira modificação é a ordenação, essa etapa de ordenação se deve ao fato de ser necessário modificar os dados do som sem fazer com que se perca as suas propriedades, ou seja, mesmo depois de se fazer isso se possa voltar ao estado original do som. Assim sendo o vetor então é ordenado pelas suas amplitudes em ordem crescente, assim as amplitudes mais próximas de zero ficam por primeiro e as amplitudes mais próximas de 255 ficam por último (Figura 5.3.1).



Figura 5.3.1 Mostra o gráfico do mesmo sinal capturado, agora ordenado pelas suas amplitudes.

Com o vetor ordenado fazemos então o que chamamos digitalização, ou seja, transformamos todos os valores em 0 e 1, os 30% últimos valores (os mais próximos de 255) são transformados em 1, e os 70% restante são transformados em 0 (Figura 5.3.2).



Figura 5.3.2 Mostra o ponto de corte, onde acima dele todos os valores passam a ser 1 e abaixo passam a ser 0.

Esse procedimento de digitalização é uma técnica inédita, jamais havia sido aplicada em outros trabalhos. Esse procedimento é realizado para haver uma diferenciação entre os valores, pois é necessário chegar a outra forma de se representar o som, uma forma mais simples ao computador, ou seja, diminuindo o número de dados que constituem o som, mas sem que esse perca suas propriedades. Com o vetor digitalizado e ainda ordenado passa-se então os seus valores para a sua posição original, isso é feito através de seu índice de posição, ou seja, antes de ordená-lo é feita a numeração de cada posição, então na verdade o vetor passa a ser uma matriz, em que uma linha passa a conter o valor (a amplitude), e a outra linha passa a conter a sua posição (índice), assim sendo ele é primeiramente ordenado pelas amplitudes e depois volta-se a sua posição original ordenando-o pelo índice, após essas duas etapas de ordenação e digitalização o vetor passa a conter diversos 1 e 0 espalhados por toda sua extensão.

Com esse vetor digitalizado criado começa-se a marcar os pontos de mudança. Os pontos de mudança seriam quando os valores do vetor mudam de 1 para 0, ou de 0 para 1, como mostra a figura 5.3.3.

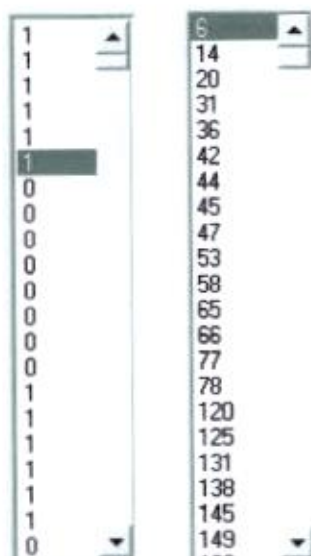


Figura 5.3.3 mostra vetores, binário e ponto de mudança sucessivamente.

Com esses pontos de mudança que é sempre crescente, é capaz reproduzir o som gravado, com isso pode-se verificar a integridade das propriedades do som gravado, ou seja, confirmar se mesmo após ser modificado a reprodução do mesmo ainda condiz com o que foi gravado. Essa reprodução do som através dos pontos de mudança só é capaz de ser realizada no Matlab, através de sua função "sound (x, f)", que recebe como parâmetro um vetor (x), e a frequência (f), no caso o vetor seria os pontos de mudança da palavra, e a frequência seria a mesma que o som foi gravado.

5.4 EXTRAÇÃO DAS CARACTERÍSTICAS

Para extrair as características que serão usadas na Rede FAN, utilizamos um novo vetor, que é criado a partir dos pontos de mudança, fazemos isso tirando a diferença do primeiro ponto de mudança em relação ao ponto de mudança seguinte, e assim sucessivamente (Figura 5.4.1), esse novo vetor é chamado de Vetor

Diferença, através dele que extraímos as sete características que irão representar a palavra gravada.

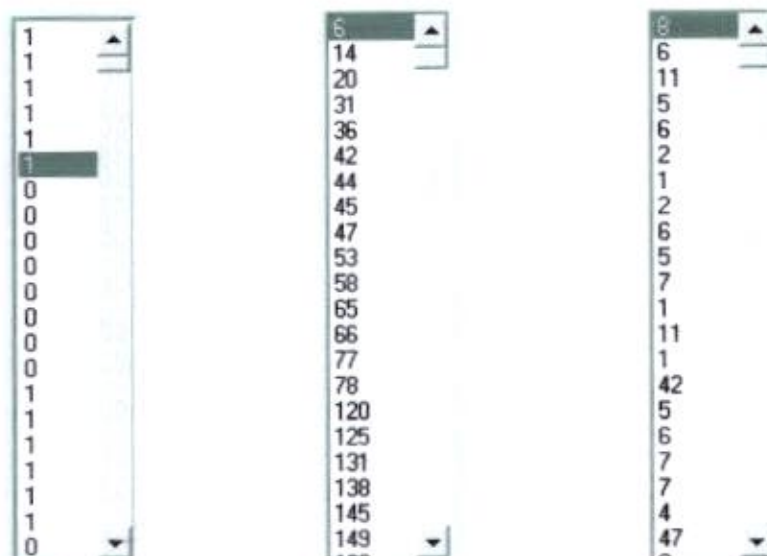


Figura 5.4.1 Mostra o vetores, digitalizado com 0 e 1, os pontos de mudança, e o último mostra o vetor Diferença sucessivamente, nota-se a relação entre eles.

Com o Vetor Diferença criado é tirado então o logaritmo neperiano de cada valor do vetor, dessa forma o valor resultante do logaritmo será sempre um valor entre 1 e 7, assim sendo quando o resultado for 1, a característica 1 é incrementada, quando o resultado do logaritmo for 2 a característica 2 é incrementada e assim por diante. Com isso cada palavra terá um padrão de características, quando esses valores das sete características são representadas num gráfico observa-se que cada palavra tem uma certa forma (padrão).

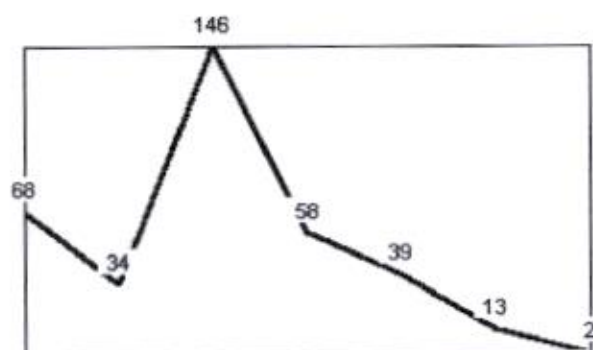


Figura 5.4.2 Gráfico das características da palavra Bode

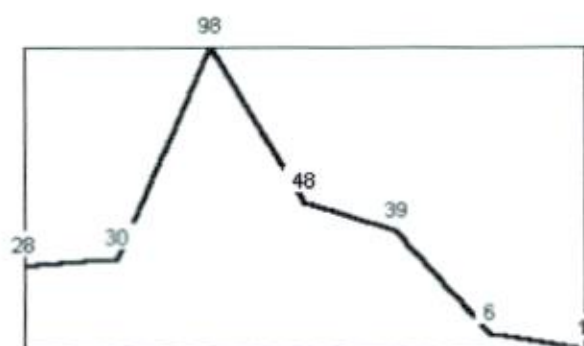


Figura 5.4.3 Gráfico das características da palavra Bote

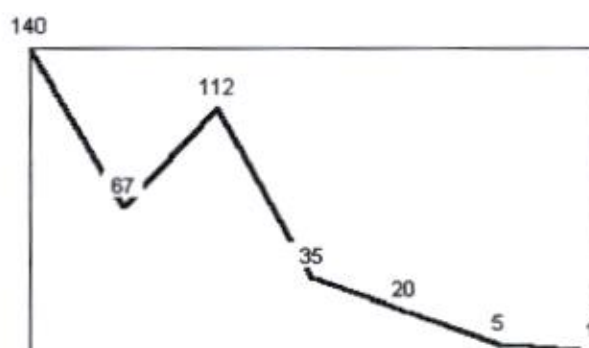


Figura 5.4.4 Gráfico das características da palavra Gato

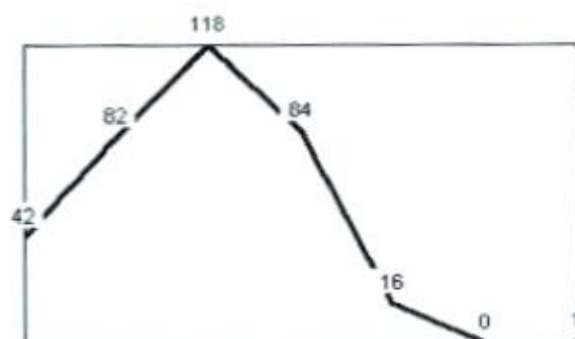


Figura 5.4.5 Gráfico das características da palavra Gado

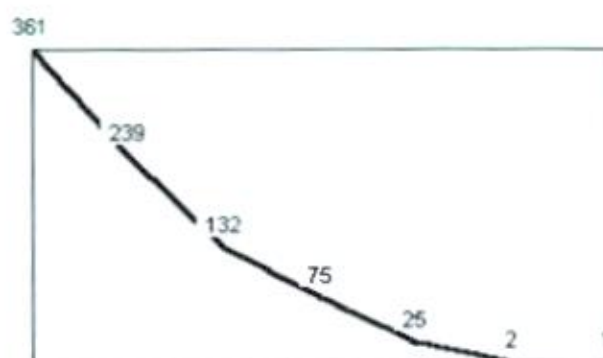


Figura 5.4.6 Gráfico das características da palavra Vaca

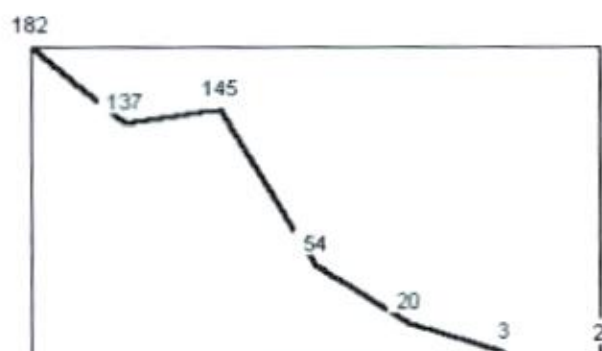


Figura 5.4.7 Gráfico das características da palavra Faca

5.5 USO DA REDE FAN

Para a Rede FAN reconhecer cada palavra é necessário antes treiná-la, para que ela aprenda os padrões de cada palavra. Para treinar a rede usamos o aplicativo Laboratório FAN (LabFAN), desenvolvido por André Luis Biaggi e a rede original por Roberto Tadeu Raittz no projeto CIASHOP, Roberto Tadeu Raittz, o qual foi um dos idealizadores do Sistema FAN, nos concedeu o uso desse aplicativo que foi de extrema serventia, pois é o Laboratório FAN que cria a Rede Neural, o qual é capaz de gravá-lo em um arquivo com extensão ".fan", e com esse arquivo ".fan" a rede é "carregada" no nosso Laboratório de Reconhecimento para testar a qualidade da rede em tempo real.

Esse teste se baseia em diversos locutores falarem as seis palavras em ordem randômica e a cada nova palavra falada o Laboratório de Reconhecimento retorna o resultado em ordem do maior grau de pertinência para o menor, assim podemos saber se o resultado retornado condiz com a palavra falada (acerto) e sabemos ainda se a rede está próxima do acerto (quase acerto) ou se o resultado não está condizente (erro), por exemplo, a palavra falada foi bode, a rede retornou bote em primeiro e bode em segundo, ou seja, o resultado é errado, mas condizente, pois ainda é aceitável que a rede confunda bode com bote, agora quando a palavra falada foi faca e a rede retorna bode em primeiro e faca em terceiro significa que está longe do acerto, assim deduzimos que a rede necessita de mais treinamento.

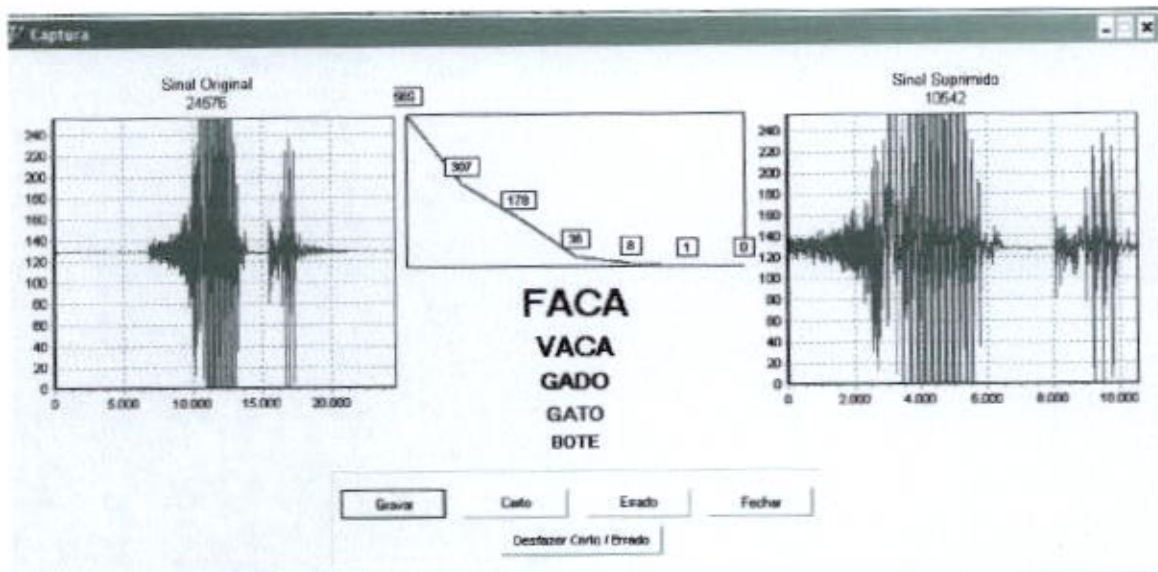


Figura 5.5.1 Mostra Tela de captura do Laboratório de Reconhecimento com as palavras em ordem de pertinência.

Podemos notar na figura 5.5.1, que a palavra "Faca" tem o maior grau de pertinência, ou seja, é palavra que Laboratório de Reconhecimento diz como "Certa", e, seguida temos a palavra "Vaca", que possui grau de pertinência menor que o da palavra "Faca", a palavra "Gado" tem grau de pertinência menor que "Faca" e "Vaca" e assim sucessivamente para as demais palavras.

Vale ressaltar ainda que o arquivo de extensão ".fan" criado pelo LabFan (Laboratório Fan) e carregado no nosso Laboratório de Reconhecimento localmente, pois é maior a versatilidade de manuseio com os arquivos pela equipe de desenvolvimento, porém quando é usado no Protótipo o arquivo extensão ".fan" é carregado do banco de dados, pois é necessário se manter a integridade dos dados, pois o Protótipo poderá ser manuseado por outras pessoas que não tem conhecimento sobre o funcionamento de todo o processo de reconhecimento, assim o arquivo extensão ".fan" que ficaria no computador poderia ser deletado ou

modificado por descuido de alguém, mantendo o arquivo de extensão ".fan" no banco de dados garantimos a integridade da rede.

Para o LabFAN criar o arquivo de extensão ".fan" (a rede), é necessário ter um arquivo de padrões, esse arquivo de padrões nada mais é que um arquivo texto (extensão .txt) em que cada linha representa um padrão de uma palavra, uma linha seria então os valores das sete características e por último a classe (palavra) a qual essas sete características pertencem, por esse motivo foi necessário definir um valor para cada palavra, no caso ficou assim: 1 – Bode; 2 – Bote; 3 – Gato; 4 – Gado; 5 – Vaca; 6 – Faca. O arquivo de padrões é criado a partir do nosso Laboratório de Reconhecimento, a cada palavra falada se tem a opção de adicionar as características ao arquivo de padrões indicando sempre qual foi a palavra falada, assim cada linha do arquivo de padrões representa uma palavra gravada, é necessário ter um arquivo de padrões com um grande número de linhas (padrões), para a rede final o arquivo de padrões tinha 2222 padrões, aproximadamente 550 padrões de cada membro da equipe.

O aprendizado se baseia em duas etapas, a rede lê todos os padrões vendo suas respectivas classes para que ele possa assimilar cada padrão à sua classe, esse seria o treinamento, e em seguida testa tentando acertar a classe lendo apenas os padrões, e tentando "adivinhar" a classe que esse padrão pertence, a cada acerto o neurônio (classe) é estimulado e a cada erro o neurônio é desestimulado, a cada vez que as duas etapas se completam é chamado uma época, a cada época que se passa a rede vai cada vez aprendendo mais, por esse motivo é preciso ter um grande número de padrões gravados e um treinamento de diversas épocas para que a rede aprenda as pequenas diferenças que possam haver em padrões da mesma

palavra e defina com clareza as diferenças entre os padrões das seis palavras. Abaixo mostraremos gráficos de comparação das características entre as palavras de maior semelhança, no caso, bode com bote, gado com gato e vaca com faca, em cada gráfico de comparação foram gravados 10 palavras de cada.

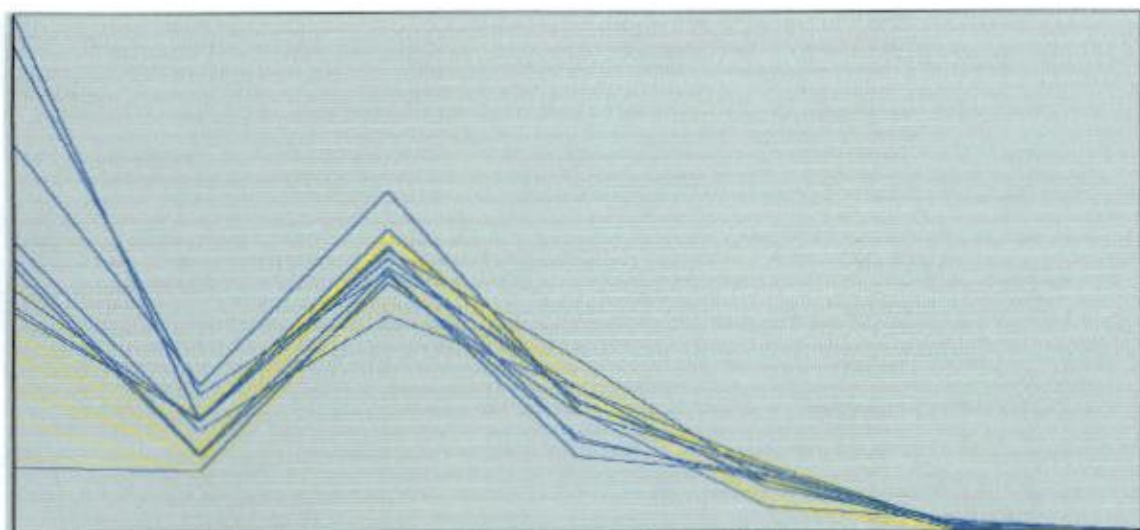


Figura 5.5.2 Gráfico de comparação entre as palavras Bode em amarelo e Bote em azul.

Na figura acima nota-se grande semelhança entre as duas palavras, há poucas diferenças entre as suas características, a maior diferença percebe-se na primeira característica, na característica 3 e 4 pouca diferença e nas demais praticamente iguais isso torna a geometria das duas palavras muito parecidas.

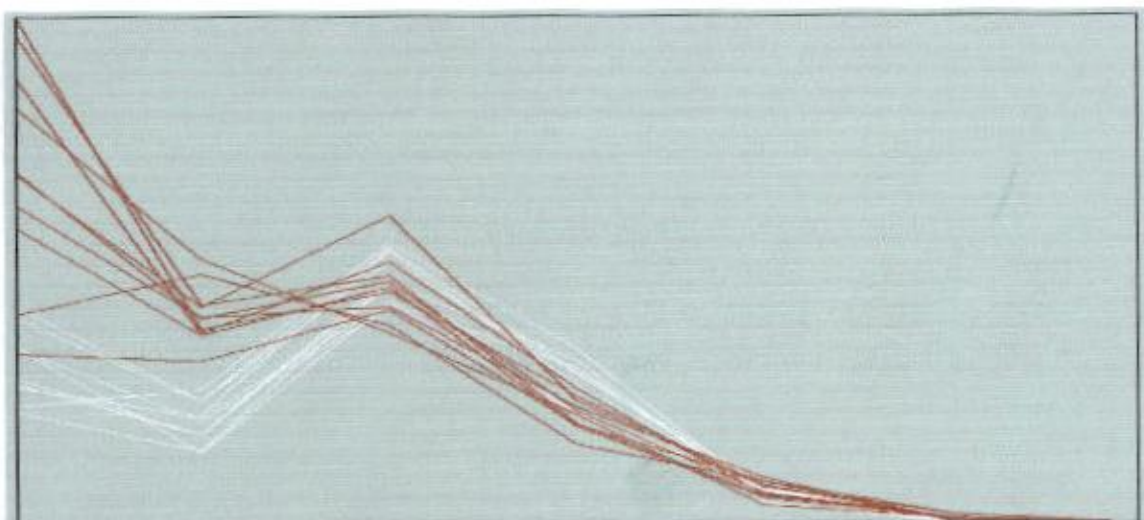


Figura 5.5.3 Gráfico de comparação entre as palavras Gado em branco e Gato em marrom.

Já na comparação entre gado e gato percebe-se grande diferença de geometria entre as duas palavras, fazendo com que a rede não confunda tanto uma com a outra, pois nota-se que a características 1 e 2 na palavra gado se mantêm bem abaixo da palavra gato, da característica 3 em diante já há uma certa semelhança.

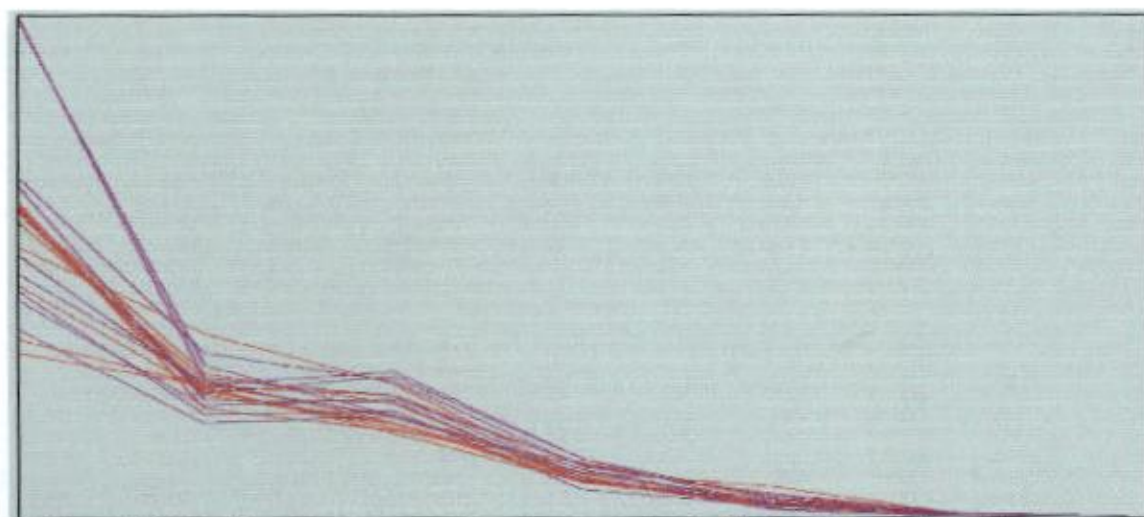


Figura 5.5.4 Gráfico de comparação entre as palavras Vaca em vermelho e Faca em roxo.

Comparando as palavras vaca e faca percebe-se a maior semelhança em relação às comparações anteriores, sendo mínima as diferenças entre as suas características, o que podemos notar nessa figura é que a características da palavra vacas se mantiveram muito mais estáveis.

Ao observamos esses gráficos de comparação podemos perceber que as diferenças entre as palavras não são tão notáveis, mas o suficiente para a rede FAN aprender as suas diferenças, e outro fator que ficou bem claro são as variações que ocorrem entre os padrões da mesma palavra, ressaltando ainda, que as palavras gravadas (10 de cada), foram gravadas todas pelo mesmo locutor, ou seja, as variações nas características se intensificariam quando o som é gravado por outro locutor, por isso o treinamento da rede final foi realizado com padrões de quatro locutores diferentes.

6 LABORATÓRIO DE RECONHECIMENTO

O Laboratório de Reconhecimento é o programa que contém todas as ferramentas que utilizamos, todas elas desenvolvidas por nós, com exceção do Laboratório FAN. O Laboratório de Reconhecimento foi desenvolvido com a finalidade de organizar as nossas ferramentas e apresentar algo mais concreto para finalizar o projeto, ao invés de vários programas separados.

As ferramentas que constituem o Laboratório de Reconhecimento são: Captura de Voz, Gravar Rede no Banco, Gravar Padrões no Banco, Refinador de Padrões, Contador de Padrões e o Laboratório FAN, como se pode ver na figura 6.1. Nos itens a seguir serão detalhadas cada ferramenta contida no Laboratório.

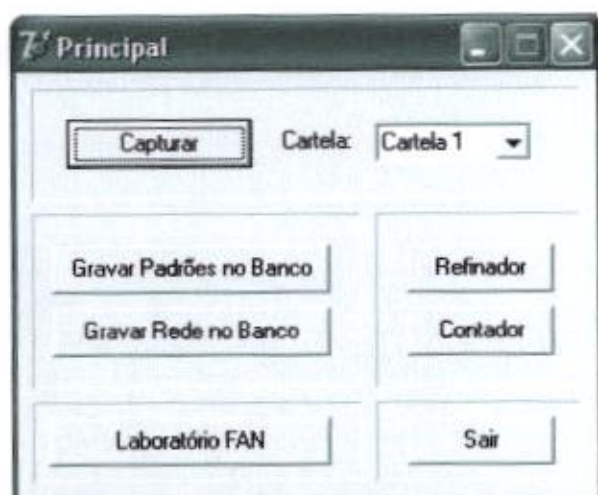


Figura 6.1 Tela Principal do Laboratório

6.1 CAPTURA DE VOZ

Na captura de voz, é feito todo o processo de captura, extração de padrões e reconhecimento. Podendo também haver o treinamento on-line da rede usada. O programa é constituído de 1 painel de resultados, 3 gráficos e 5 botões que serão explicados a seguir.

- **Painel de Resultados:** É o lugar onde aparece a lista de palavras, ordenados de acordo com o grau de pertinência, a primeira palavra (Faca) figura 6.1.1 , é a que será usada como certa, e em seguida estão as palavras que o programa usaria como segunda, terceira, quarta e quinta opções respectivamente.
- **Gráfico "Sinal Original":** Consiste no gráfico das amplitudes da palavra falada no microfone, sem nenhum tipo de supressão ou compactação.

- **Gráfico “Sinal Suprimido”:** É o sinal suprimido da palavra falada no microfone, ou seja, é retirado do sinal original os ruídos de início e fim da palavra e posto apenas o gráfico da palavra em si no gráfico.
- **Gráfico das características:** É o gráfico do centro, onde estão indicadas as características que definem o padrão da palavra falada.
- **Botão “Gravar” e “Parar”:** De início, o botão é nomeado “Gravar”, que é o botão que se clica para iniciar a gravação, após clicado, o botão é redefinido como “Parar”, que deve ser clicado após falada a palavra no microfone, ao clicar o botão “Parar”, todos os gráficos são atualizados e os dados da palavra são gravados na memória para exibição painel de resultados.
- **Botão “Certo”:** Este botão é usado exclusivamente para agilizar o processo de gravação de padrões e treinamento on-line, ao falar a palavra e caso o valor de maior pertinência no painel de resultados seja correspondente à palavra falada, deve-se clicar no botão certo, que o programa automaticamente adiciona o padrão da palavra falada no arquivo de padrões e ao mesmo tempo estimula o neurônio correspondente a palavra para treinamento on-line.
- **Botão “Errado”:** É usado caso a palavra de maior grau de pertinência indicado no painel de resultados não corresponder à palavra falada no microfone, ao clicar no botão errado, aparece uma lista popup das possíveis palavras que podem ser correspondentes ao que foi falado, deve-se escolher a palavra certa, para que o padrão seja adicionado

corretamente no arquivo de padrões e o neurônio correspondente à palavra que apareceu como de maior pertinência seja desestimulado.

- **Botão “Desfazer Certo / Errado”** : A pessoa a qual está gravando os padrões fala a palavra “Bode”, e o Laboratório de Reconhecimento identifica como “Bote”, e por descuido essa pessoa clica no botão “Certo”. O que ocorre é que a pessoa estará gravando no arquivo de padrões as características da palavra “Bote” como classe pertencente ao “Bode”, assim prejudicando futuro treinamento da rede. O botão “Desfazer Certo / Errado” foi feito exatamente para isso, assim o Laboratório de Reconhecimento deleta a ultima linha inserida no arquivo de padrões. Vale lembrar caso o Laboratório de Reconhecimento acerte a palavra falada pela pessoa, e equivocadamente esta clica em “Errado”, também poderá utilizar a opção “Desfazer Certo / Errado”.
- **Botão “Sair”**: Sai da tela de captura e volta para a tela principal do Laboratório de Reconhecimento.

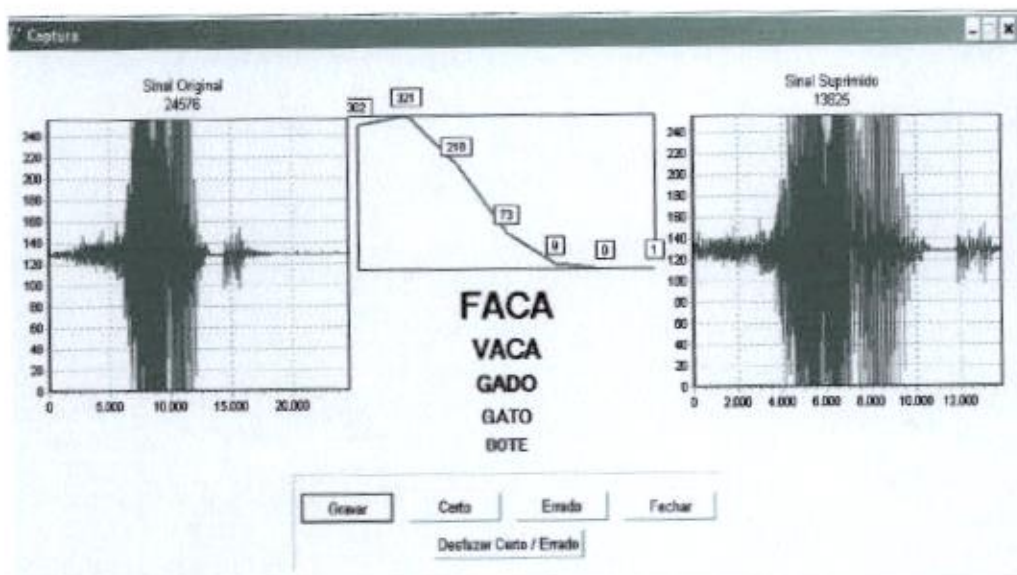


Figura 6.1.1 Tela de captura do Laboratório de Reconhecimento.

Vale lembrar que pessoa antes de clicar em "Capturar" (Figura 6.1), deve selecionar a cartela correspondente a qual será gravado o conjunto de padrões.

6.2 GRAVAR PADRÕES NO BANCO

É um pequeno programa desenvolvido apenas para gravar um arquivo de padrões no banco de dados do protótipo, para uso exclusivamente do mesmo.

Para utilizar o programa, deve-se clicar em "Abrir", selecionar o arquivo de padrões desejado e em seguida clicar em "Executar" para que os padrões sejam gravados no banco de dados. Na figura 6.2.1 foi carregado um arquivo com 756 padrões.

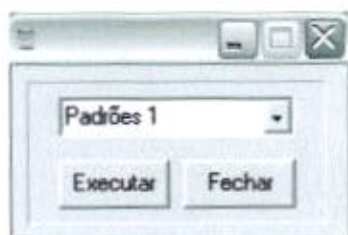


Figura 6.2.1 Tela de Gravar Padrões no Banco

6.3 GRAVAR REDE NO BANCO

É um programa semelhante ao "Gravar Padrões no Banco", foi desenvolvido apenas para gravar um arquivo de rede no banco de dados para uso exclusivo do protótipo, pois o Laboratório de Reconhecimento não faz acesso ao banco de dados para resgatar valores de padrões e da rede. Para melhorar o desempenho, os valores são resgatados de arquivos locais correspondentes e são facilmente manipuláveis.

Para utilizar o programa, deve-se escolher na caixa "Combo Box" qual a rede que deseja-se enviar ao banco, sendo que cada rede corresponde a uma cartela do Protótipo, após escolhida a rede, deve-se clicar em "Executar" para enviar os dados ao banco (Figura 6.3.1).



Figura 6.3.1 Tela de Gravar Rede no Banco

6.4 REFINADOR

O Refinador, é um programa usado para melhorar a qualidade da rede após cada sessão de treinamento no Laboratório FAN, como já foi explicado, após um treinamento no LabFAN, há um arquivo gerado dentro do diretório do próprio LabFAN nomeado "fc.fsy", neste arquivo contém os dados do treinamento, indicando quais padrões que obtiveram sucesso e quais não obtiveram sucesso durante o treinamento, ou seja, quais acertaram e quais erraram. O programa simplesmente retira os padrões que não obtiveram sucesso durante o treinamento e salva em um novo arquivo para uso em treinamento posterior, para obter melhor qualidade da rede, diminuindo assim a taxa de erro.

Para utilizar o Refinador, deve-se ter o arquivo "fc.fsy" gerado pelo LabFAN dentro do diretório do Laboratório, em seguida deve-se clicar em "Abrir" e escolher o arquivo de padrões que foi usado no treinamento, o programa irá manter o arquivo de padrões original e criará um arquivo refinado, agora basta clicar em "Salvar" que o arquivo refinado será salvo.

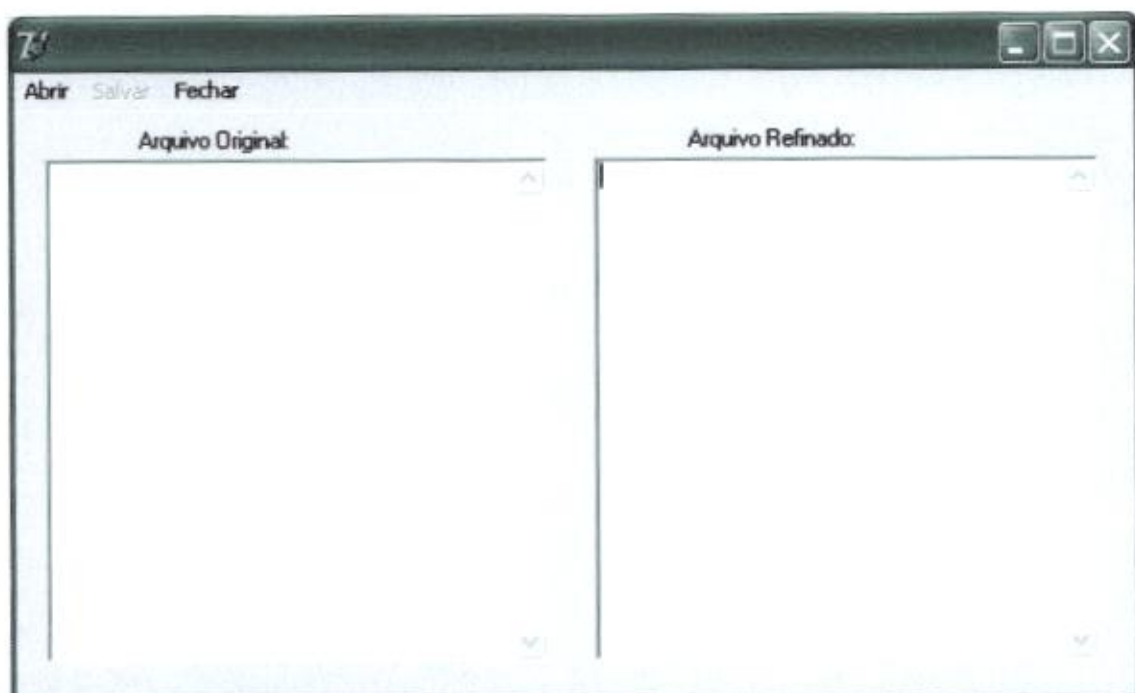


Figura 6.4.1 Tela do Refinador sem dados

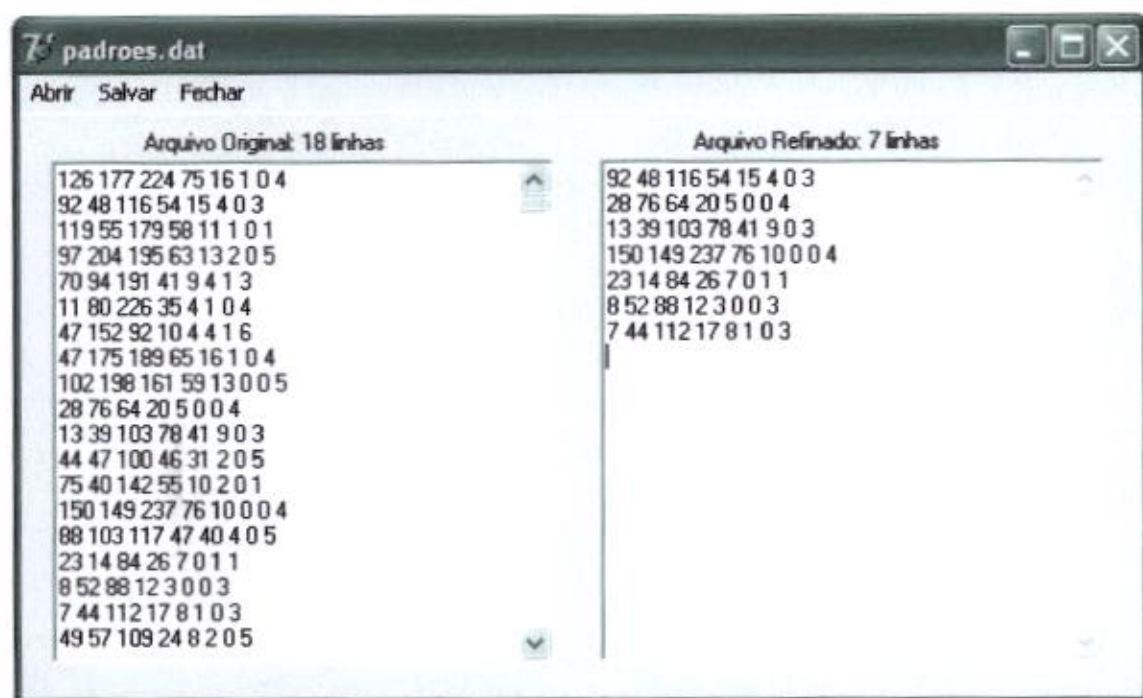


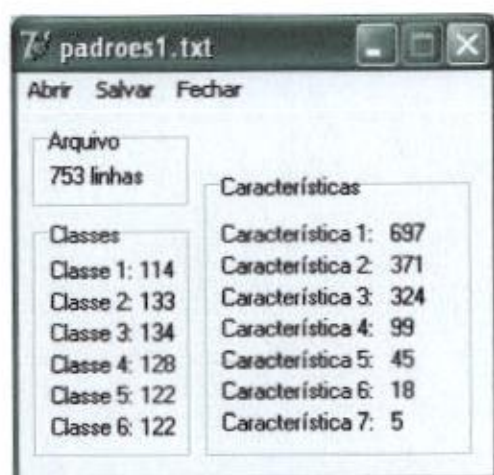
Figura 6.4.2 Tela do Refinador com dados

Podemos notar na figura 6.4.2 um exemplo de 18 padrões somente 7 correspondem aos "bons" padrões.

6.5 CONTADOR

O Contador é um programa útil feito para enumerar o número de padrões contido em um arquivo, separando-os por classe e indica qual o maior valor de cada característica de todos os padrões contidos no arquivo.

Para utilizar o Contador, deve-se clicar em "Abrir" e indicar o arquivo desejado, em seguida o programa exibirá as informações contidas no arquivo. O programa tem outra função bem útil, que é de misturar as ordens dos valores para que não fiquem duas classes seguidas uma da outra, para ter mais eficiência na hora do treinamento.



6.5.1 Tela do Contador de Padrões

6.6 Laboratório FAN

Ver item 8 LABORATÓRIO FAN.

7 PROTÓTIPO

7.1 INTRODUÇÃO

Protótipo é uma aplicação baseada em sistemas inteligentes para detectar sintomas mais evidentes sobre desvios na recepção oral e escrita e respectiva produção, é baseado em uma bateria de testes no qual estaremos implantando o recurso de reconhecimento voz em seis palavras (Bode, Bote, Vaca, Faca, Gato, Gado) da primeira cartela para posterior alocação no Teste 2.1 Produção Oral de Palavras. A seguir é contemplado o seguinte grupo de testes.

Bateria de Testes

1 – Testes de Recepção Oral:
1.1 – Recepção Auditiva
1.2 - Compreensão de Frases
2 – Testes de Produção Oral:
2.1 – Produção Oral de Palavras
2.2 – Produção Oral de Frases
3 – Inversão a Partir de uma Seqüência de Gravuras
4 – Reconto de História
5 – Emparelhamento de Palavras e Frases Escritas com Gravuras:
5.1 – Emparelhamento de Palavras
5.2 – Emparelhamento de Frases Escritas
6 – Produção Escrita a Partir de Gravuras:
6.1 – Produção Escrita de Palavras
6.2 – Produção Escrita de Frases
7 – Teste de Correspondência Fonológico-Grafêmica
8 – Teste de Correspondência Grafêmico-Fonológica
9 – Teste de Leitura em Voz Alta e de Compreensão de Leitura:
9.1 – Teste de leitura em voz alta
9.2 – Compreensão de leitura

Tabela 7.1.1 Bateria completa de testes do Protótipo

Previamente é necessário que o avaliador faça cadastro do avaliado e se por ventura for dar continuação no desenvolvimento do protótipo deverá ter os componentes do Delphi instalado (citados futuramente item 11) e base de dados para que possa acessar as tabelas contendo as informações necessárias para execução do teste 2.1 Produção Oral de Palavras.

7.2 TESTE 2.1 PRODUÇÃO ORAL DE PALAVRAS

O teste 2.1 de produção oral de palavras tem por objetivo detectar se o avaliado comanda os gestos fonoarticulatórios da sua variedade sociolingüística. O teste é composto de quinze cartelas cada uma contendo seis palavras onde a pessoa a ser avaliada irá identificar a figura destacada e falar no microfone a figura correspondente.

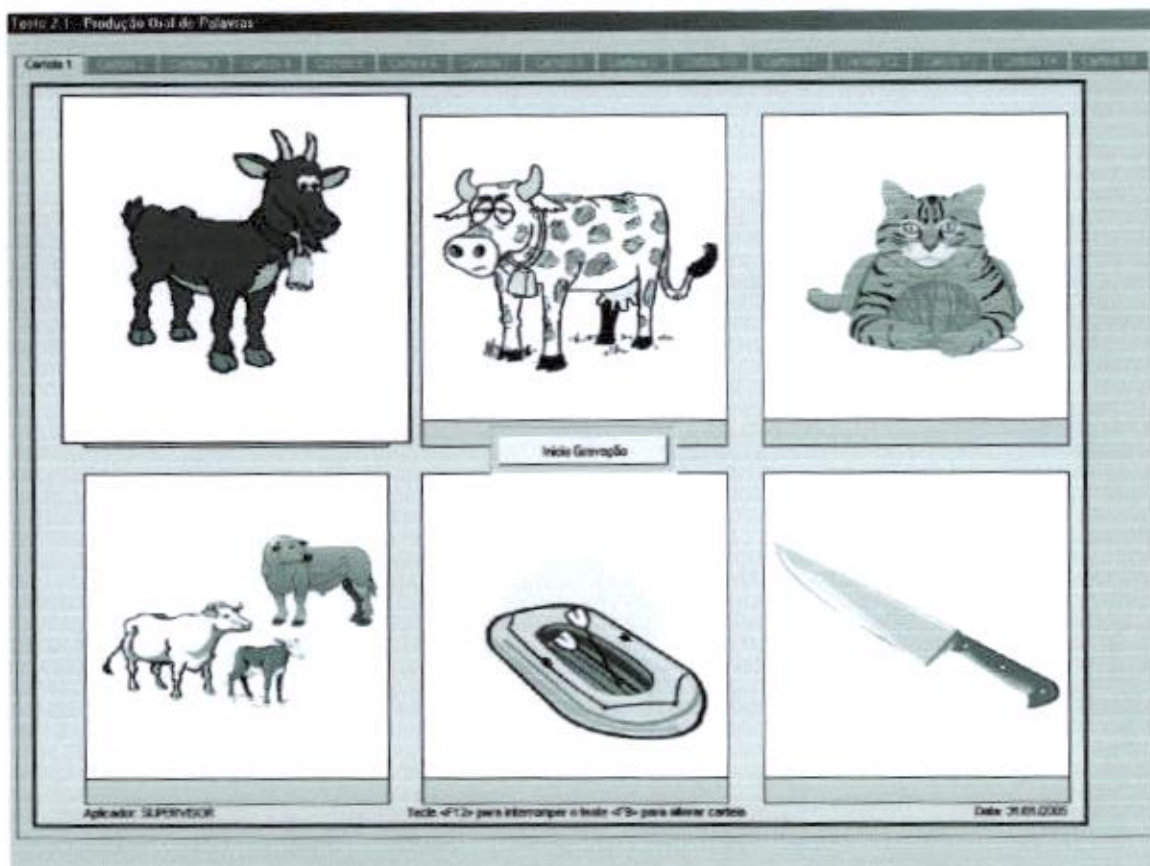


Figura 7.2.1 Primeira cartela do teste 2.1 Produção Oral de Palavras.

Acima é demonstrada a primeira cartela na qual implantamos o reconhecimento das seis palavras, nota-se a figura "Bode" em destaque a qual o avaliado irá posicionar microfone perto da boca, clicar em iniciar gravação e falar a palavra, em seguida ele clica em parar gravação, feito isso protótipo irá reconhecer a palavra falada e passará para próxima figura da cartela (Bote). Vale lembrar que protótipo no teste 2.1 sempre destacara as figuras pela ordem de similaridade, ou seja, Bode, Bote, Vaca, Faca, Gato, Gado.

7.3 BASE DE DADOS

O Protótipo guarda todos os dados necessários no banco de dados, informações do tipo aplicador, local e aluno, que são dados necessários quando um novo teste vai ser aplicado, essas informações são cadastradas pelo Protótipo e gravadas no banco de dados.

Mas os dados necessários para o reconhecimento, como a rede (arquivo .fan, que nada mais é que um conjunto bem grande de valores), os padrões (arquivo de padrões criado pelo Laboratório de Reconhecimento) e os valores "nx" (que são valores usados pela rede em conjunto com os padrões) são gravados no banco somente através do Laboratório de Reconhecimento.

O banco de dados nomeado com o nome "tese", possui as seguintes tabelas: aluno, aplicador, histórico, local_aplicacao, resultado, teste_11, teste_21, teste_51, essas tabelas citadas não são manuseadas em nenhum momento pelo Laboratório de Reconhecimento, são tabelas de uso exclusivo do Protótipo e criadas pelo elaborador do Protótipo.

Para que o recurso de reconhecimento pudesse ser implementado no Protótipo foi necessário criar as seguintes tabelas:

- palavras: Guarda todas as palavras usadas no Teste 2.1 do protótipo, ou seja 90 palavras, pois o Teste 2.1 se baseia em 15 cartelas com 6 palavras cada.
- redes: guarda as informações sobre as redes, no caso são 15 redes, uma para cada cartela do Teste 2.1, essa tabela serve apenas para verificar se uma rede já foi criada ou não.

- rede_1, rede_2, rede_3, rede_4, rede_5, rede_6, rede_7, rede_8, rede_9, rede_10, rede_11, rede_12, rede_13, rede_14, rede_15: Nessas tabelas é onde ficam gravados os valores da rede propriamente ditas, valores esses que são lidos do arquivo extensão ".fan" criado pelo LabFAN, no caso cada rede corresponde à uma cartela do Teste 2.1, rede_1 será lida quando a cartela 1 for inicializada, rede_2 quando a cartela 2 for inicializada e assim sucessivamente.
- carac_1, carac_2, carac_3, carac_4, carac_5, carac_6, carac_7, carac_8, carac_9, carac_10, carac_11, carac_12, carac_13, carac_14, carac_15: Nessas tabelas é onde ficam gravados os valores do arquivo de padrões criado pelo Laboratório de Reconhecimento, esses padrões precisam ser lidos quando a sua respectiva rede é lida, pois a rede para ser testada (apontar um resultado) precisa ter acesso ao conjunto de padrões na qual a mesma foi treinada, no caso a relação se dá da mesma forma com as cartelas, cartela 1: rede_1: carac_1 e assim sucessivamente.
- nx: Nessa tabela são gravados os valores nx, os valores nx também precisam ser lidos quando a rede precisa ser utilizada, esses valores nx se baseiam em 8 valores, os sete primeiros valores são os maiores valores de cada característica, esses maiores valores são obtidos através do arquivo de padrões, e o último valor é a soma de todos os sete valores, esses 8 valores também são interligados com a rede e os padrões, mas como os valores nx se baseiam em 8 valores foi possível se criar apenas uma tabela para todos sendo necessário apenas adicionar um campo para o código da rede.

O motivo da criação de 15 tabelas de rede e 15 tabelas de padrões se deve ao fato de que cada rede e cada arquivo de padrões contém grande número de valores e essa desnormalização foi necessária para melhor performance do processo de reconhecimento.

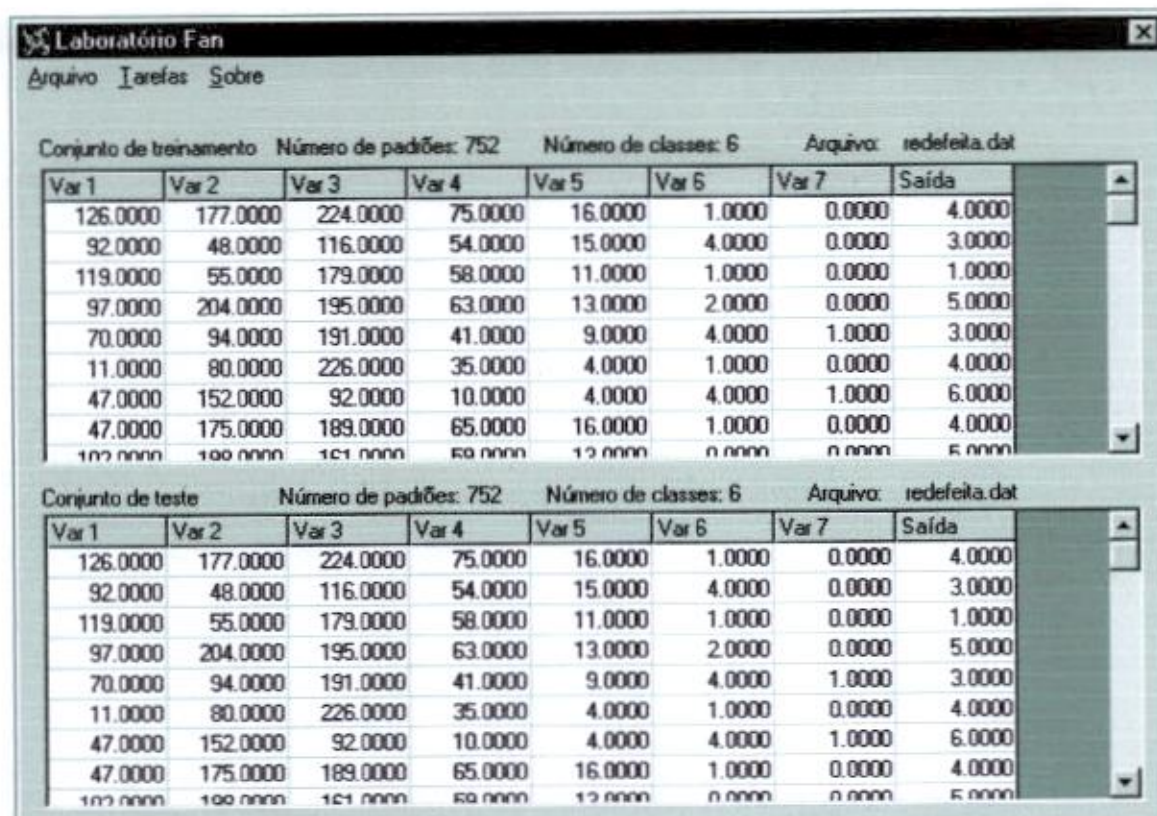
7.4 RECONHECIMENTO

O algoritmo de reconhecimento foi implantado no protótipo no "Form" o qual denominamos `t_fase_21` (vide código fonte Protótipo), logo quando ocorre a execução do protótipo o mesmo carrega base de dados remotamente e ao início do teste 2.1 a pessoa fala a palavra correspondente em seguida o protótipo faz reconhecimento e passa para próxima figura, e assim sucessivamente.

Vale lembrar que para cada cartela existe uma rede diferente, e após o reconhecimento haverá o retorno de uma das seis palavras da primeira cartela (Bode, Bote, Vaca, Faca, Gato, Gado), por exemplo, na próxima cartela que conteria as palavras (Calo, Galo, Queijo, Queixo, Doce, Doze) o protótipo carregaria a rede treinada com somente essas palavras e assim sucessivamente para as demais cartelas.

8 LABORATÓRIO FAN

Desenvolvido por André Luis Biaggi e a rede original por Roberto Tadeu Raitz no projeto CIASHOP, o Laboratório FAN, vulgo LabFAN, é uma ferramenta designada para construção e treinamento de redes FAN, os padrões de treinamento de teste são lidos através de um arquivo por exemplo (arquivo.dat), no qual os padrões estão contidos separados por linha e cada linha contem as características do padrão correspondente, sendo que a última característica é a classe. Após carregados, os padrões são alocados em um "grid" de informações.



The screenshot shows the main window of the 'Laboratório Fan' application. At the top, there is a menu bar with 'Arquivo', 'Ajuda', and 'Sobre'. Below the menu, the application status is displayed: 'Conjunto de treinamento: Número de padrões: 752 Número de classes: 6 Arquivo: redefeita.dat'. The main area contains two data grids. The top grid is for the training set, and the bottom grid is for the test set. Both grids have the same structure: 8 columns labeled 'Var 1' through 'Var 7' and 'Saída', and 10 rows of data. The data values are identical in both grids.

Var 1	Var 2	Var 3	Var 4	Var 5	Var 6	Var 7	Saída
126.0000	177.0000	224.0000	75.0000	16.0000	1.0000	0.0000	4.0000
92.0000	48.0000	116.0000	54.0000	15.0000	4.0000	0.0000	3.0000
119.0000	55.0000	179.0000	58.0000	11.0000	1.0000	0.0000	1.0000
97.0000	204.0000	195.0000	63.0000	13.0000	2.0000	0.0000	5.0000
70.0000	94.0000	191.0000	41.0000	9.0000	4.0000	1.0000	3.0000
11.0000	80.0000	226.0000	35.0000	4.0000	1.0000	0.0000	4.0000
47.0000	152.0000	92.0000	10.0000	4.0000	4.0000	1.0000	6.0000
47.0000	175.0000	189.0000	65.0000	16.0000	1.0000	0.0000	4.0000
102.0000	100.0000	101.0000	60.0000	12.0000	0.0000	0.0000	5.0000

Figura 8.1.1 Tela principal do LabFAN com os arquivos de treinamento e teste carregados.

Notamos na figura acima que tanto conjunto de treinamento quanto o de teste já estão "carregados" para dar início ao estudo dos padrões. Nota-se que (var 1, var 2, var 3, var 4, var 5, var 6, var 7) correspondem as sete características e (Saída) a classe que pertence. Neste caso foram carregados o mesmo arquivo tanto para treinamento quanto para teste. Depois feito isso damos início a configuração dos "Parâmetros de treinamento" clicas-se em (tarefas/treinamento/configurações/parâmetros de treinamento) e ajustamos os parâmetros específicos da seguinte forma como mostra a figura abaixo.

Parâmetros de treinamento

Critério de parada

Épocas

Tempo de treinamento

Dias Horas Minutos Segundos

Parar quando atingir um dos valores abaixo

Média Absoluta Média Harmônica

Parâmetros específicos

Range Raio difuso

Peso a ser editado

OK

Figura 8.1.2 Configuração dos parâmetros de treinamento.

Como podemos notar na figura acima o range segue com valor 100 e Raio difuso 6, e na opção "Peso a ser editado" todas as 6 classes recebem peso igual á 500 pois todas têm a mesma prioridade.

Depois de feito os ajustes iniciais na configuração dos "parâmetros específicos" damos inicio ao treinamento do conjunto de padrões como mostra Figura 8.1.3.

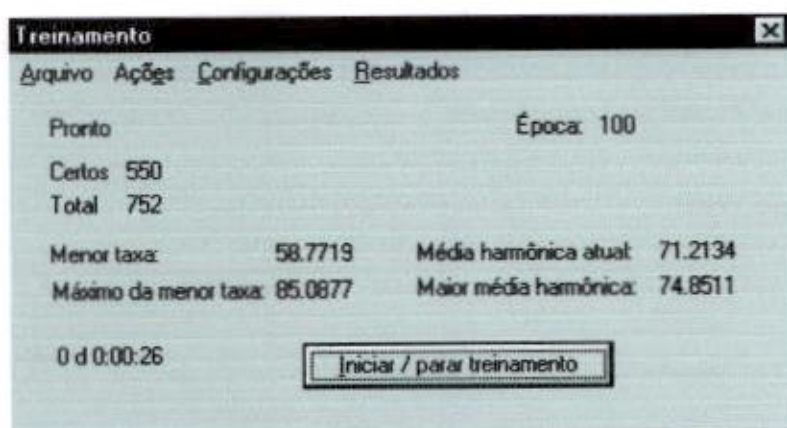


Figura 8.1.3 Tela de treinamento.

Na figura acima mostramos exemplo de treinamento de 100 (cem) épocas, obtendo uma taxa de acerto (média harmônica atual) de 71.2134 % (por cento). A quantidade de épocas que utilizamos foi aproximadamente 100.000(cem mil) para que a rede obtenha uma maior taxa de acertos em seu treinamento. Obtemos uma taxa de 99 %(por cento) no treinamento de em dois conjuntos de 752 padrões lembrando que conseguimos essa taxa depois de passar pelo processo de refinamento, onde eliminamos os padrões que prejudicam a taxa de acerto durante o treinamento e deixamos somente aqueles que contribuem para bom aprendizado da rede.

Também podemos analisar o treinamento da rede em relação aos acertos obtidos através da "Matriz de confusão" ou "tabela de confusão" como mostra a Figura 8.1.4.

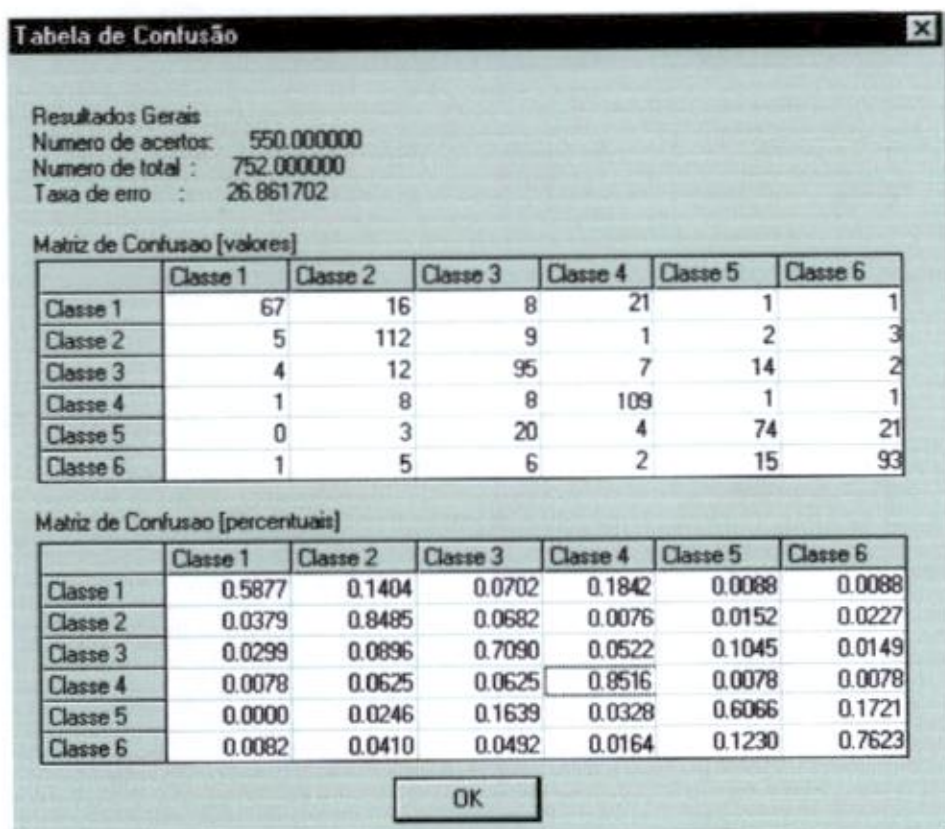


Figura 8.1.4 Figura mostrando a tabela de confusão.

Notamos na figura acima que dentre 752 (Número de total) padrões houve acerto de 550 (Numero de acertos), ou seja, uma taxa erro de 26.861702 % (por cento) ou 73.138298% (por cento) de acerto, durante treinamento de 100 (cem) épocas.

Ainda na figura acima se notamos na "Matriz de confusão [valores]" na linha da Classe 1 encontramos os seguintes valores (67, 16, 8, 21, 1, 1) que somados resultam 90, este valor é numero total de palavras da classe 1, ou seja, na linha da

Classe 1 o valor 67 corresponde ao número de palavras da classe 1 que pertencem a esta classe portanto acertou, 16 ao número de palavras da classe 1 que rede achou que pertencia a classe 2 portanto errou, e assim sucessivamente. Abaixo a "Matriz de confusão [percentuais]" mostra o percentual do dito acima.

Após isso voltamos para tela de treinamento (Figura 8.1.3) e deixamos treinar por aproximadamente 100.000 (cem mil) épocas. Logo salvamos a rede com nome qualquer e se a mesma ainda não foi "refinada", utilizamos o refinador, ferramenta que está contida no laboratório, logo em seguida realizamos um novo treinamento de 100.000 épocas (agora com padrões refinados) e salvamos a rede definitiva para ser utilizada tanto no laboratório quanto no protótipo para reconhecimento do universo definido de palavras.

9 TAREFAS DESENVOLVIDAS

9.1 TAREFAS MÊS: MARÇO

- Elaboração Plano de Projeto
- Estudos na Área de reconhecimento de voz

9.2 TAREFAS MÊS: ABRIL

- Construção Diagramas de Casos de uso.
- Elaboração da Descrição Resumida dos casos de uso.
- Estudo Código-Fonte de Programas de Captura de Som.

- Instalação componentes externos para ferramenta Borland Delphi 7.
- Entendimento do protótipo apresentado.

9.3 TAREFAS MÊS: MAIO

- Construção do diagrama de classes.
- Análise dos Diagramas já construídos.
- Instalação banco de dados nas máquinas do laboratório 3.
- Conexão Do banco de dados com protótipo apresentado.
- Execução em tempo real do protótipo assim como os programas de captura de som dentro da ferramenta Delphi 7.

9.4 TAREFAS MÊS: JUNHO

- Construção dos diagramas de objetos e dos fluxos de eventos e cenários.
- Construção dos diagramas de seqüência.
- Construção dos diagramas de colaboração.
- Desenvolvimento de uma aplicação a parte para gravação de voz com gráficos cartesianos amplitude/tempo.

9.5 TAREFAS MÊS: JULHO

- Análise das atividades realizadas pela equipe.
- Construção dos diagramas de objetos e dos fluxos de eventos e cenários.

- Construção dos diagramas de seqüência.
- Construção dos diagramas de colaboração.
- Formatação da voz capturada para futura implementação.

9.6 TAREFAS MÊS: AGOSTO

- Construção dos diagramas de objetos e dos fluxos de eventos e cenários.
- Construção dos diagramas de seqüência.
- Construção dos diagramas de colaboração.
- Formatação da voz capturada para futura implementação no FAN.

9.7 TAREFAS MÊS: SETEMBRO

- Construção dos diagramas de objetos e dos fluxos de eventos e cenários.
- Construção dos diagramas de seqüência.
- Construção dos diagramas de colaboração.
- Formatação da voz capturada para futura implementação no FAN.

9.8 TAREFAS MÊS: OUTUBRO

- Construção dos diagramas de objetos e dos fluxos de eventos e cenários.
- Construção dos diagramas de seqüência.
- Construção dos diagramas de colaboração.
- Formatação da voz capturada para futura implementação no FAN.

9.9 TAREFAS MÊS: NOVEMBRO

- Estudos para elaboração do diagrama de estados baseados no livro de FURLAN.
- Construção Diagramas de Estado.
- Estudos para elaboração do diagrama de atividades baseados no livro de FURLAN.
- Construção Diagramas de atividades.
- Implantação do FAN dentro do código de reconhecimento.

9.10 TAREFAS MÊS: DEZEMBRO

- Estudos para elaboração do diagrama de componentes baseados no livro de FURLAN.
- Construção dos diagramas de componentes.
- Análise dos diagramas construídos.
- Criação do programa para refinamento automático.
- Criação de rede e refinamento dos padrões para melhor desempenho no FAN.
- Implantação do reconhecimento de voz no protótipo.
- Melhoramento na rede FAN.
- Execução de testes finais.

10 LIMITAÇÕES

Avanços tecnológicos significativos que certamente passam por uma abordagem multidisciplinar, envolvendo aspectos acústicos, fonéticos, fonológicos, léxicos, sintáticos, semânticos e pragmáticos. A união destas áreas é que devem ser estudadas em trabalhos futuros. A seguir são encontradas as limitações que envolvem nosso trabalho.

10.1 MICROFONE

Antes de mais nada é necessário saber algumas coisas sobre os microfones adequados para serem acoplados na placa de som.

Para conseguir um reconhecimento de qualidade é preciso comprar um bom microfone. Existem diversos microfones para essa finalidade, mas são geralmente mais caros do que os comuns. Numa boa loja de informática você poderá comprar um microfone para reconhecimento de voz, com supressão de ruído de fundo. Existem microfones que vem com defeito de fábrica, sua qualidade não é mais indicada para realização de trabalho científico para reconhecimento de voz, esses microfones quando você pega-os ou oscila o fio que o conecta na placa de vídeo, produz ruídos, e com isso prejudica a captura da palavra fala.

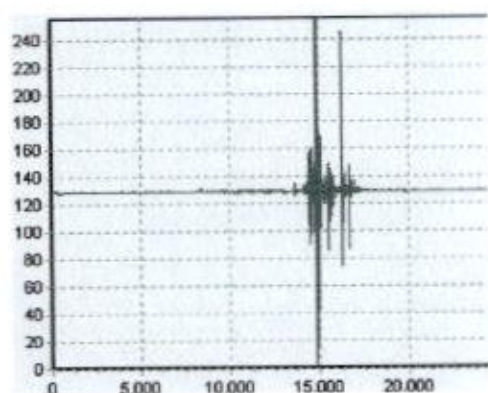


Figura 10.1.1 Ruído

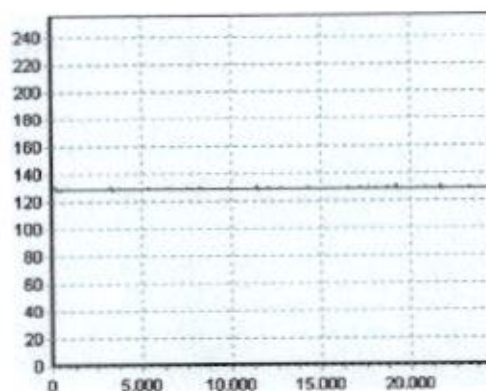


Figura 10.1.2 Sem Ruído

Como mostra a Figura 10.1.1 acima foi feito teste usando um microfone de má qualidade, onde simples movimento de oscilação do fio que conecta o microfone a placa de som ocasionou ruído. E ao lado a figura 10.1.2 mostra microfone de boa qualidade sofrendo os mesmos movimentos de oscilação.

Outro fator importante é o controlador de volume do microfone, para isso acione o controle do volume do Windows. Como o som do microfone não pode sair simultaneamente nas caixas de som, verifique na opção de propriedades de reprodução que o microfone não está para ser ecoado como mostra a figura 10.1.3.

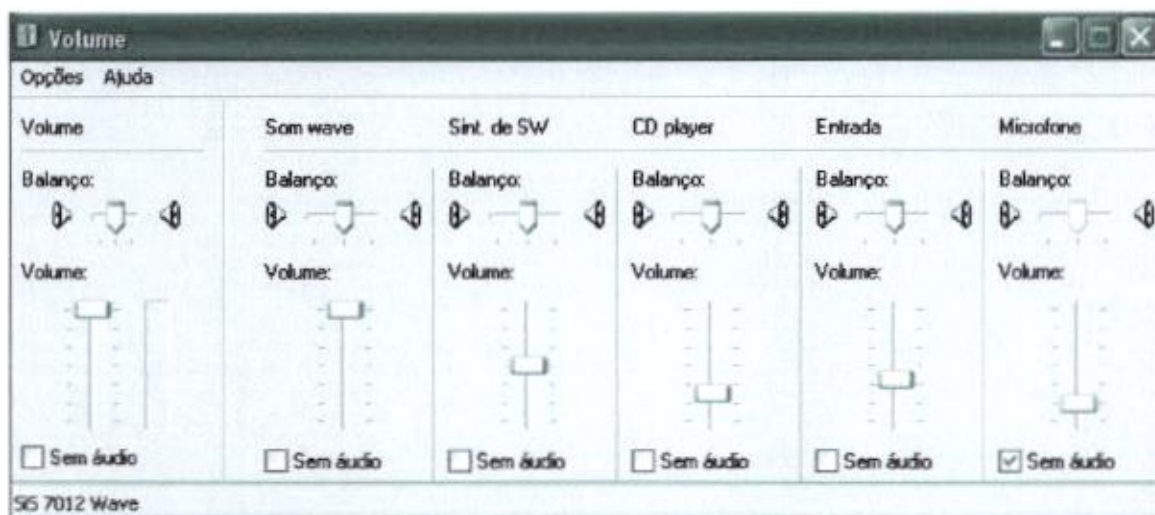


Figura 10.1.3 controlador de som Windows XP

Nota-se na figura acima na configuração do microfone, a opção "sem áudio" está habilitada, isto ocorre do fato quando a pessoa falar uma palavra não ocasionar som de fundo e assim prejudicando a gravação no reconhecimento. Além disso percebe-se o volume baixo do microfone, com o volume baixo estaremos melhorando a captura do sinal da voz assim diminuindo a captação dos ruídos externos que venham a prejudicar o sinal de voz capturado.

10.2 RUÍDO AMBIENTE

Para adquirir padrões de qualidade, sem ruídos externos, é necessário que o Laboratório de Reconhecimento capture somente a palavra falada, ou seja, pode haver padrões que contenha ruídos externos junto com padrões de qualidade, e com isso ocasionará problemas quando for treinar a rede, a ideia é que somente padrões de qualidade sejam aproveitáveis.

10.3 ERRO NA GRAVAÇÃO DOS PADRÕES

Durante a gravação dos padrões no laboratório de reconhecimento, deve-se tomar cuidado quando for clicar na opção "Certo" ou "Errado", pois a utilização errada dessas opções comprometeriam a geração do arquivo de padrões.

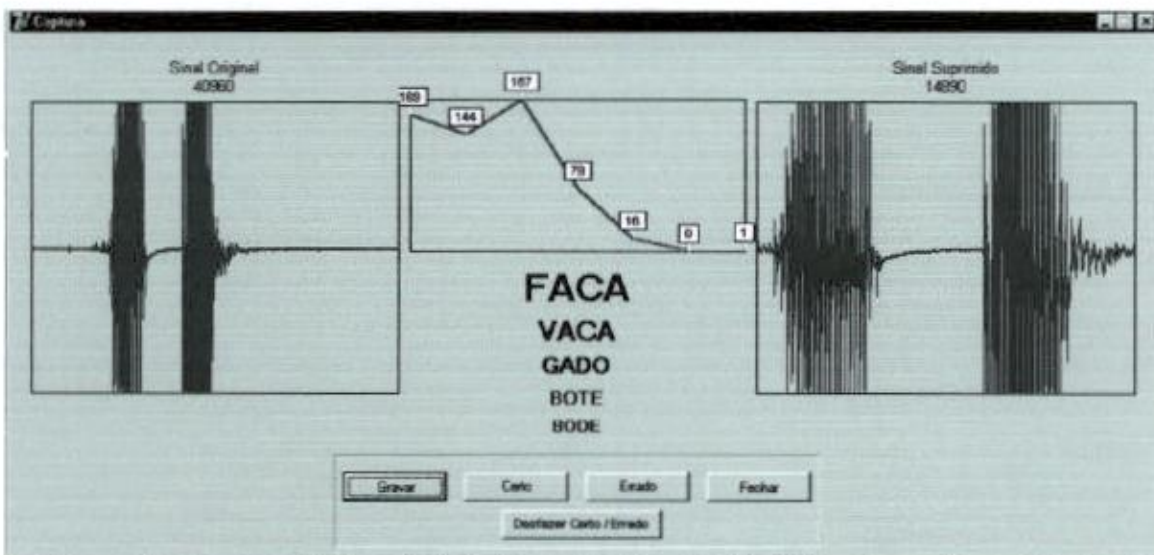


Figura 10.3.1 mostra Laboratório de Reconhecimento identificando a palavra falada pela pessoa.

No exemplo da figura a cima a pessoa falou a palavra "Faca". Como o laboratório de reconhecimento acertou a palavra falada pela pessoa, então ela clica em "Certo" para efetuar a gravação dos padrões no arquivo. Se por ventura a palavra falada fosse "Bote" por exemplo, e o laboratório de reconhecimento tivesse reconhecido como "Faca", como mostra a Figura 10.3.1 acima, então a pessoa iria clicar em "Errado" e logo surgiria "ComboBox" com as seis palavras, Bode, Bote, Gato, Gado, Vaca, Faca.

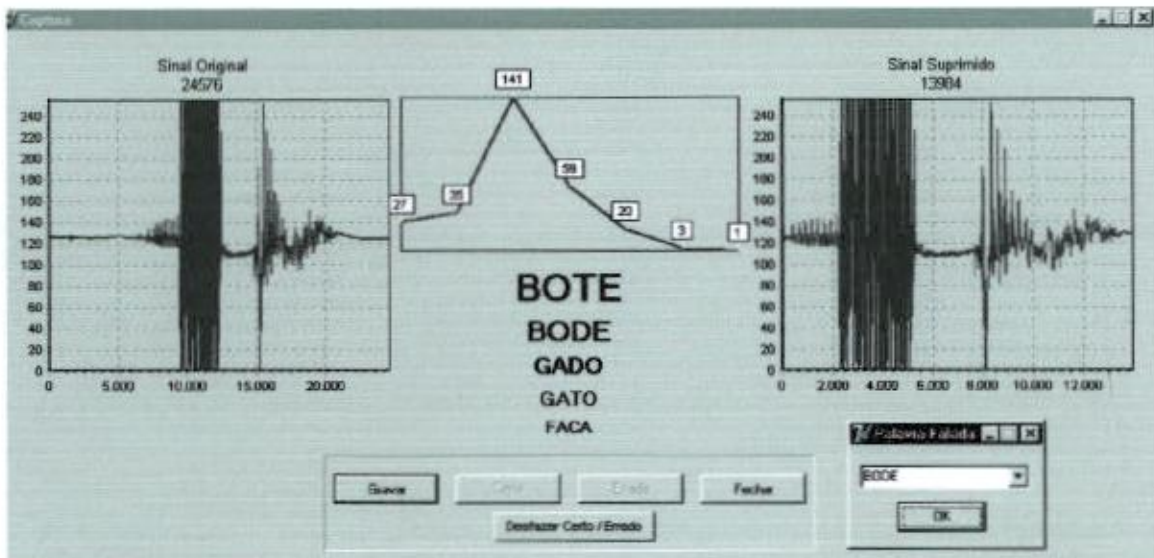


Figura 10.3.2 mostra tela de reconhecimento e tela (Palavra Falada) das seis palavras.

Notamos na Figura 10.3.2 que a pessoa por exemplo, falou a palavra "Bode" mais o Laboratório de Reconhecimento reconheceu como "Bote", quando isso ocorre a pessoa deverá clicar em "Errado" logo abrirá uma tela, com "ComboBox" contendo seis palavras, onde a pessoa irá selecionar a palavra a qual falou, e clica em "ok", para armazenar no arquivo de padrões a classe correta da palavra falada.

Se por ventura a pessoa, falar a palavra Faca, e o Laboratório de Reconhecimento identificar como vaca por exemplo, e a pessoa clicar em certo, simplesmente ela clicará na opção "Desfazer Certo / Errado" e o Laboratório de Reconhecimento irá deletar do arquivo de padrões a ultima linha nele inserida. Assim evitando que armazene um padrão com uma classe a qual ele não pertence.

10.4 PROCESSO DE REFINAMENTO DOS DADOS

Após adquirir grande quantidade de padrões e treina-los no LabFAN verifica-se quais padrões rede não acerta, estes padrões são eliminados e em seguida realizado novo treinamento com somente os padrões que a rede acertou, assim obtendo uma taxa superior a 90 % (por cento).

10.5 RUÍDOS DO SISTEMA

Se a instalação da placa de som for deficiente, serão detectados ruídos elétricos no circuito circundante e incluídos na gravação padrões. Isto é especialmente verdade se estiver a usar um microfone ligado diretamente a entrada no computador. Se tiver eliminado todas as outras fontes possíveis de ruído e continuar a ter um ruído inexplicável, a causa poderá ser esta.

10.6 MODO DE FALAR

Se a voz for demasiadamente forte, o microfone distorcerá o som, e se for muito baixo, o microfone não irá captar o som. A pessoa deverá fazer configuração ideal do microfone como descrito anteriormente nas limitações Microfone . Logo após deverá falar normalmente, ou seja, deve falar naturalmente, seguindo uma distância de 2 cm (centímetros) da boca ao microfone. Para boa geração de padrões, é aconselhável que a pessoa pronuncie bem os fonemas por exemplo palavra vaca

(Va-ca), assim melhorando o aprendizado no LabFAN . Vale lembrar também que número de pontos (Sinal Original) não deve ultrapassar 40.000 pontos.

10.7 SONS DIFÍCEIS

Algumas consoantes são difíceis de gravar, porque elas são suaves e criam grandes quantidades de vento provocadas pela respiração Ex (Faca). Deverás falar próximo do microfone, mas não permitir que o vento provocado pela respiração lhe toque. Isto lhe permite sentir o fluxo de ar que vem da sua boca quando gera estes sons, para ter a certeza que o micro está corretamente colocado. Assim o sinal da palavra "Faca", difere de se falar perto e pouco mais distante do microfone.

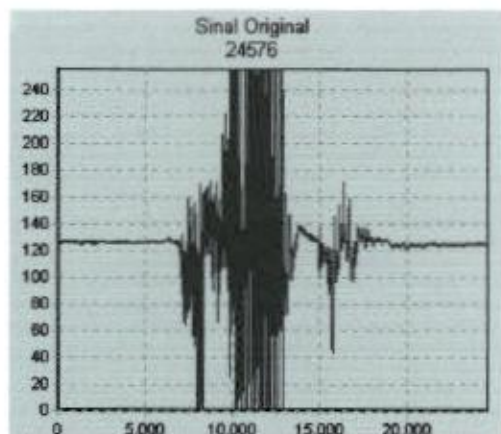


Figura 10.7.1 mostra palavra Faca com distância de 2 cm da boca ao microfone.

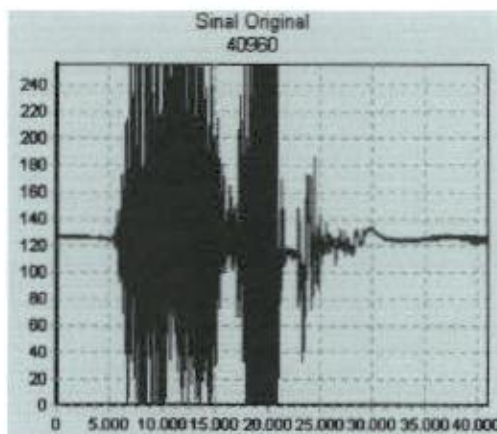


Figura 10.7.2 mostra a palavra Faca falada perto do microfone.

Como mostra a figura 10.7.1 acima, a palavra "Faca" foi capturada perfeitamente, onde seguiu-se a determinação da distância entre a boca e o microfone, assim gerando um bom padrão para treinamento. Já figura 10.7.2

demonstra o vento da respiração tocando o microfone, onde ocasionou um aumento no número pontos, ocorrendo uma mudança na estrutura da palavra (características).

10.8. GRANDE NÚMERO DE LOCUTORES

Outro fator que influencia é o treinamento da rede, o número de locutores utilizados na gravação dos padrões de treinamento aumenta a abrangência da rede em reconhecer diferentes tipos de timbre e tons de voz. Quanto maior o número de padrões utilizados para o treinamento melhor será a possibilidade de acerto, pois com um grande número de padrões é mais garantido que a rede aprenda as mínimas variações entre uma palavra e outra, garantindo assim um melhor aprendizado.

11 FERRAMENTAS UTILIZADAS

Durante o desenvolvimento de nosso projeto, utilizamos diversas ferramentas. Abaixo segue breve descrição das ferramentas utilizadas durante a fase de implementação.

11.1 BORLAND DELPHI 7

O Borland Delphi 7 foi a ferramenta que utilizamos para o desenvolvimento de nosso projeto. Optamos pela utilização do Delphi 7, por estarmos familiarizados com ela, pois durante nossa formação acadêmica, no curso de Tecnologia em Informática, foi a ferramenta que utilizamos no desenvolvimento da maioria de nossos trabalhos acadêmicos.

O Delphi é um ambiente de desenvolvimento, orientado a objetos que utiliza a linguagem Object Pascal como base para programação. Desenvolvido pela *Borland Software Corporation*, e sendo a ferramenta mais premiada para construção de aplicativos para a plataforma Windows, o Delphi é uma ferramenta de desenvolvimento muito produtiva, permitindo metodologias de desenvolvimento capazes de construir robustos e complexos aplicativos em pouquíssimo tempo, e não exigindo códigos complexos (como ocorre no C), caso haja boa elaboração na metodologia. Maiores informações sobre esta ferramenta pode encontrar em (www.borland.com.br) ou no livro *Dominando o Delphi 7*.

Para o desenvolvimento do projeto de reconhecimento de palavras, são necessários um grupo de componentes externos utilizados pelo Delphi 7. Entraremos em melhores detalhes a seguir sobre esses componentes externos.

11.1.1 TAUDIO

Componente de áudio do Delphi, que controla dispositivos de som e mixagem, como placas de som *Sound Blaster*, permitindo que o Delphi controle a reprodução e gravação de som em formato *Wave*.

As seguintes propriedades do TAudio dentro do "*Object Inspector*" do Delphi a seguir, são necessárias para efetuar a gravação dos padrões.



Figura 11.1.1.1 mostra Object Inspector do Delphi 7.

Podemos notar acima na figura 11.1.1.1 as propriedades do TAudio as quais utilizamos para gravação das palavras.

11.1.2 ZEOS

O protótipo de testes psicológicos, desenvolvido no Delphi, é uma aplicação necessita comunicar-se com banco de dados Postgresql. Para essa comunicação é necessário a instalação do componente Zeos no Delphi.

O Zeos é um pacote de Componentes para comunicação do Delphi com PostgreSQL. Dentro deste pacote são acompanhados os seguintes componentes:

CORE, PARSE, PARSESQL, PLAIN, DBC, COMPONENT. Mais informações ou aquisição da biblioteca Zeos encontra-se em (www.zeoslib.net).

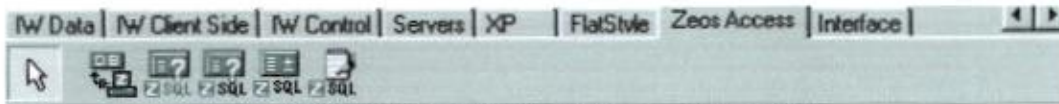


Figura 11.1.2.1 Paleta do Zeos no Delphi 7.

11.1.3 FLATSTYLE

O FlatStyle é uma paleta de componentes voltada para aperfeiçoamentos visuais das aplicações. Nela podemos encontrar diversos objetos que foram construídos com o intuito de facilitar o desenvolvimento de interfaces diferenciadas. Há também objetos não visuais que, em tempo de execução, facilitam a integração com outros componentes. Abaixo na figura 11.1.3.1 a imagem da paleta do FlatStyle.

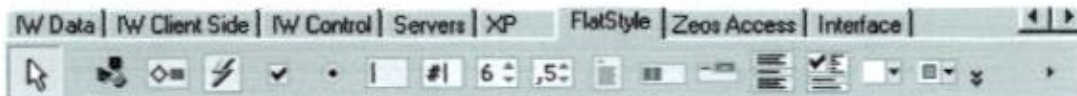


Figura 11.1.3.1 paleta FlatStyle no Delphi 7.

11.1.4 XPMENU

Para melhor atingir uma interface amigável no protótipo de testes psicológicos, foi utilizado o XPMenu.

O XPMenu é um Componente que permite deixar menus iguais ao do OfficeXp / WindowsXp.

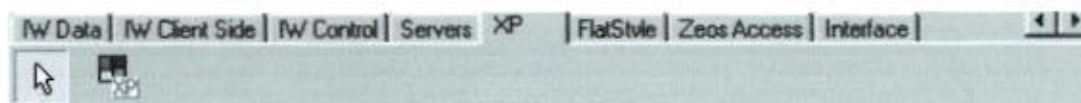


Figura 11.1.4.1 mostra paleta do XPMenu no Delphi 7.

11.2 LABORATÓRIO FAN

Software para treinamento da rede FAN, de simples utilização, sendo apenas necessário adaptar os arquivos para a formatação utilizada pelo programa para a leitura dos padrões. O Laboratório FAN é melhor explicado no item 8.

11.3 MICROSOFT VISIO 2002 PROFESSIONAL

Utilizamos o Microsoft Visio 2002 *Professional* na construção das modelagens do Protótipo e do Laboratório de Reconhecimento, pois se trata de uma ferramenta de fácil manuseio e completa para as necessidades do projeto.

O Microsoft Visio 2002 *Professional* é uma poderosa ferramenta de diagramação e documentação que ajuda os desenvolvedores em tarefas de especificação, documentação, modelagem de objetos, modelagem de dados, projetos de redes, segurança, e outros.

Podem-se organizar as idéias graficamente e compartilhá-las com outras pessoas. O Visio é uma excelente ferramenta de colaboração para pessoas que estejam trabalhando em projetos de grupo.

Adicionalmente, esta ferramenta é ótima para fazer com que as pessoas se acostumem planejar o projeto de software antes de sair codificando-o.

Para mais informação sobre o Visio 2002 *Professional*, favor visite a página do produto: <http://www.microsoft.com/brasil/office/visio>.

11.4 LTZ POSTGRESQL

A versão que utilizamos foi PostgreSQL 7.2, é um banco de dados robusto, estável e completo, possui recursos indispensáveis em banco de dados, por exemplo, triggers, sub-queries e etc. Utilizamos o banco de dados PostgreSQL pela características apresentadas, e por já estar sendo utilizado no desenvolvimento do Protótipo de Testes Psicológicos.

PostgreSQL é um banco SGBD (Sistema Gerenciador de Banco de Dados) objeto-relacional. Ele é considerado objeto-relacional por implementar, além das características de um SGBD relacional, algumas características de orientação a objetos, como herança e tipos personalizados. Maiores informações em www.postgresql.com.br ou no livro PostgreSQL : Conceitos e Aplicações.

11.5 MATLAB 6.5

Com os pontos de mudança (vetor pontos de mudança) onde é sempre crescente, é capaz reproduzir o som gravado, com isso pode-se verificar a integridade das propriedades do som gravado, ou seja, confirmar se mesmo após ser modificado a reprodução do mesmo ainda condiz com o que foi gravado. Essa

reprodução do som através dos pontos de mudança só é capaz de ser realizada no MatLab, através de sua função "sound(x,f)", que recebe como parâmetro um vetor (x), e a frequência (f), no caso o vetor seria os pontos de mudança da palavra, e a frequência seria a mesma que o som foi gravado.

MATLAB (MATrix LABoratory) é um programa interativo para cálculos (e visualização) numéricos científicos e de engenharia. MATLAB® é um produto distribuído pela empresa norte-americana The MathWorks, Inc. Originalmente voltado para a álgebra linear numérica e o cálculo matricial, MATLAB tem seus recursos básicos estendidos através de "toolboxes" (conjunto de funções ou "M-files") para áreas especializadas (p. ex. processamento de sinais – "Signal Processing Toolbox"; sistemas de controle – "Control System Toolbox"; identificação de sistemas – "System Identification Toolbox"; otimização – "Optimization Toolbox"; redes neurais – "Neural Network Toolbox", e vários outros). Os recursos instalados também podem ser estendidos pelo usuário através da implementação de funções MATLAB (M-files) ou de rotinas escritas em linguagem C ou FORTRAN. Maiores informações podem ser encontradas no livro Matlab 6.5 : Fundamentos de Programação.

12 MODELAGEM

Para a construção dos diagramas deste projeto foi utilizada a UML, uma linguagem visual para modelar sistemas de software. A referência bibliográfica foi o livro de José Davi Furlan, Modelagem de objetos através da UML e usando como ferramenta o Visio 2002 Professional da Microsoft.

A modelagem se dividiu em duas etapas: a modelagem do laboratório de reconhecimento, ou seja, da aplicação que a equipe desenvolveu e a modelagem do protótipo, por solicitação do professor orientador.

A Primeira etapa do processo de modelagem, o protótipo, se tratando de um projeto já em desenvolvimento, a tarefa de modelagem que aparentemente parecia ser fácil, se tornou um pouco mais complexa, pois necessariamente devia seguir a risca o que está codificado no teste 2.1 produção oral de palavras do protótipo, mas, junto com o professor orientador, optou-se por fazer a modelagem do protótipo de uma forma como a equipe modelaria o problema, o teste 2.1, propriamente dito.

Primeiramente foi previamente definido objetivo do projeto e transcrito, e em seguida construído o diagrama de casos de uso e a descrição resumida de cada caso de uso do protótipo. Em seguida foi construído o diagrama de classe, e este primeiro pacote da modelagem do protótipo foi mostrado ao professor orientador que deu sugestões para melhoria desse diagramas.

Em seguida foi feito um refinamento em cima dos diagramas de caso de uso e de classes, o que levou a alterar também a descrição resumida dos casos de uso do protótipo. Em paralelo foi construído o diagrama de objetos e designada a tarefa da elaboração dos cenários e fluxo de eventos do protótipo a integrante da equipe Maristela.

Após isso se iniciou a parte mais pesada da diagramação, a elaboração dos diagramas de interação do protótipo. Primeiramente foram elaborados todos os diagramas de seqüência, são eles: Diagrama de seqüência – Protótipo - Cadastrar avaliador, Diagrama de seqüência - Protótipo - Cadastrar avaliado, Diagrama de seqüência - Protótipo - Cadastrar local, Diagrama de seqüência - Protótipo - Alterar

dados do avaliador, Diagrama de seqüência - Protótipo - Alterar dados do avaliado, Diagrama de seqüência - Protótipo - Alterar dados do local, Diagrama de seqüência - Protótipo - Excluir avaliador, Diagrama de seqüência - Protótipo - Excluir avaliado, Diagrama de seqüência - Protótipo - Excluir local, Diagrama de seqüência - Protótipo - Logar no sistema, Diagrama de seqüência - Protótipo - Substituição de usuário, Diagrama de seqüência - Protótipo - Teste 2.1, Diagrama de seqüência - Protótipo - Teste 2.1 – Cancelar teste, Diagrama de seqüência - Protótipo - Teste 2.1 - Interromper e apagar dados de todos os testes realizados até o momento, Diagrama de seqüência - Protótipo - Teste 2.1 – Cancelar teste, Diagrama de seqüência - Protótipo - teste 2.1- interromper e gravar dados para retornar futuramente e Diagrama de seqüência - Protótipo - Teste 2.1 – Retomar sessão interrompida. Lembrado que foram também construídos os diagramas referentes aos testes 1.1 teste de recepção oral, 5.1 emparelhamento de palavras escritas com gravuras, 5.2 emparelhamento de frases escritas com gravuras, 7 teste de correspondência fonológico-grafêmica e 8 teste de correspondência grafêmico-fonológica, cada teste possuindo cinco diagramas, no caso o teste propriamente dito, cancelando o teste, interrompendo o teste e apagando dados, interrompendo teste e salvando dados e retomando o teste, que a principio estavam para ser implementados, mas devido a o estouro do prazo das tarefas, por parte do pessoal da implementação, foi acertada com o professor orientador a inclusão apenas dos diagramas e da implantação do reconhecimento no teste 2.1, produção oral de palavras.

Após o término dos diagramas de seqüência foram construídos os diagramas de colaboração do protótipo, com os mesmos nomes dos diagramas citados acima, trocando a palavra "seqüência" pela "colaboração".

Depois de terminado esse segundo pacote da modelagem do protótipo, foram impressos os diagramas de caso de uso, diagrama de classes, diagrama de seqüência de diagrama de colaboração e mostrados ao professor orientador que juntamente com o membro da equipe Vagner fizeram, uma análise dos mesmos, discutiram os diagramas, apontaram pontos positivos e negativos, possíveis ambigüidades, e fizeram anotações para melhoria dos mesmos.

Terminada a análise por parte do orientador e do membro da equipe foram feitas alterações principalmente nos diagramas de seqüência de colaboração e algumas alterações no diagrama de classe do protótipo, e a membra da equipe Maristela apresentou os fluxos de eventos e cenários e logo após abandonou o projeto, o que ocasionou sérios comprometimentos às tarefas que já haviam sido distribuídas.

Concluída a fase de refinamento dos diagramas de seqüência e de colaboração, foi construído o DER, diagrama entidade relacionamento do protótipo e em paralelo, realizados estudos para a elaboração dos diagramas de estados e atividades, baseando-se para isso no livro de FURLAN acima citado. Depois de realizado os estudos, primeiramente foram construídos os diagramas de estados, são eles: Diagramas de estados – Protótipo - Cadastros de avaliador, avaliado e local, Diagramas de estados – Protótipo - Substituição de usuários e Diagramas de estados – Protótipo – Teste 2.1 Produção Oral de Palavras. Terminado os diagramas de estados do protótipo, foi feito um re-estudo, agora dos diagramas de atividades baseado no livro de FURLAN e iniciada a construção dos mesmos.

Depois de concluída a construção dos diagrama de atividades do protótipo foram realizados novos estudos a respeito dos diagramas de implementação no

mesmo livro e construído o diagrama de implantação do protótipo. Com o término do diagrama de implantação foi feita uma análise nos cenários e fluxo de eventos e reformulados ambos, pois a modelagem do protótipo já havia passado por dois *feedbacks*, com várias alterações que influíam em ambos.

Após a formulação de toda modelagem foi feita uma homologação em cima da mesma, a fim de achar possíveis erros e quando encontrados estes erros eram corrigidos.

Vale ressaltar que na modelagem do protótipo foi realizada com a abordagem de implementação que a equipe daria ao teste 2.1 produções oral de palavras, havendo algumas incompatibilidades com o que já havia sido desenvolvido, pois o objetivo é implementar o recurso de reconhecimento de palavras faladas no teste 2.1, produção oral de palavras e não modelar a aplicação propriamente dita, o que ocorreu de fato foi a modelagem dos cadastros e suas funcionalidades, do recurso de substituição de usuário, de logar no sistema, e do "esqueleto" do teste semi-implementado, pois não adiantaria em nada modelar o laboratório, nossa aplicação, sem que se tivesse uma noção de como que o teste 2.1 funciona. Por esse e outros motivos além da modelagem do laboratório, o professor orientador também pediu a modelagem do protótipo.

Terminada a modelagem do protótipo iniciou-se a segunda fase, a modelagem do Laboratório de Reconhecimento. Foram construídos os diagramas de casos de uso, a descrição resumida, os fluxos de eventos e cenários primeiramente. Em seguida foram construídos os diagramas de seqüência do laboratório, são eles: Diagrama de Seqüência – Laboratório – Capturar – Reconhecimento Correto, Diagrama de Seqüência – Laboratório – Capturar – Reconhecimento Errôneo,

Diagrama de Seqüência – Laboratório – Refinador, Diagrama de Seqüência – Laboratório – Gravar Rede no Banco, Diagrama de Seqüência – Laboratório – Contador, Diagrama de Seqüência – Laboratório – Gravar Padrões no Banco. Após o término dos diagramas de seqüência foram construídos os diagramas de colaboração do laboratório de reconhecimento, com os mesmos nomes dos diagramas citados acima, trocando a palavra "seqüência" pela "colaboração". Após isto, foram construídos os diagramas de estados e em seguida o de atividades, já que como os diagramas de seqüência e de colaboração, um é depende do outro para ser feito.

Vale lembra que na modelagem do Laboratório de Reconhecimento não há diagramas de classe, diagrama entidade relacionamento, diagrama de implementação e dicionário de dados, pois estes mesmos estão embutidos na modelagem do protótipo.

13 CONCLUSÃO

Apesar da taxa de acerto não alcançar os 100%, ficou provado que o uso de redes neurais utilizando a rede FAN foi capaz de reconhecer a voz nas seis palavras definidas pelo projeto, mas a princípio, a gama de palavras pode ser aumentada mantendo a mesma taxa de erro, sendo necessário apenas incluir novas classes nos arquivos de treinamento, o que decorreria em um maior tempo gasto no treinamento por causa do maior número de padrões necessário. Há ainda a possibilidade de inclusão de novas características para maior diferenciação entre as classes, chegou-

se a estudar a inclusão de duas novas características caso a rede não se provasse eficaz.

Aprendemos muito com esse trabalho, ainda se tratando de um ramo muito interessante e em constante desenvolvimento, esperamos nosso trabalho sirva como base para que outros trabalhos sejam desenvolvidos e novas idéias apareçam para evoluir cada vez mais a aplicação da inteligência artificial nos mais diversos recursos usados pelo homem.

13.1 RESULTADOS

De acordo com as nossas expectativas e objetivos atingimos o grau de acerto esperado, realizamos quatro testes, com os quatro locutores separadamente falando as seis palavras em ordem randômica.

Os resultados são mostrados na tabela a seguir:

	Bode	Bote	Gato	Gado	Vaca	Faca	%
Locutor 1	C	E	E	C	C	E	50
Locutor 2	C	C	C	C	E	C	83,33
Locutor 3	E	C	E	C	C	C	66,67
Locutor 4	C	C	C	C	C	C	100
Total (%)	75	75	50	100	75	75	75

Tabela 16.1.1 Resultados dos testes realizados.

De acordo com estes resultados foi obtida uma taxa total de acerto de 75%, esse resultado é satisfatório e provou que o nosso sistema de reconhecimento usando o Sistema FAN é bem sucedido podendo ser aplicado a outros universos de palavras, como por exemplo as outras cartelas do Teste 2.1 do Protótipo.

13.2 TRABALHOS FUTUROS

Esse método de reconhecimento de voz usando o Sistema FAN pode ser aplicado da mesma forma às outras cartelas do Teste 2.1 do Protótipo. No caso, cada cartela terá uma rede diferente, treinada com os padrões respectivos a cada universo diferente de palavras, a aplicação desenvolvida permite a gravação dos padrões para todas as 15 cartelas, dependendo assim apenas de tempo para juntar uma base de dados satisfatória para o treinamento de cada rede.

Baseando-se nessa mesma técnica de reconhecimento pode-se avançar para o reconhecimento de frases, devendo ser aprimorada alguns pontos, como por exemplo à supressão do sinal para eliminar o silêncio entre as palavras para então dividir a frase em palavras e reconhecer por partes.

Poderá ser desenvolvido para o Protótipo, uma aplicação de avaliação on-line, afim de melhorar os resultados dos testes e realização de treinamento on-line da rede dentro do Protótipo. A aplicação funcionaria da seguinte maneira: o supervisor estaria monitorando o aluno através de outro computador, conectado através de rede local, supervisionando todos os resultados obtidos, e se porventura ocorrer algum erro de reconhecimento e o supervisor perceber o erro, ele mesmo altera nos

resultados finais do teste e ao mesmo tempo é feito o treinamento da rede de acordo com o que foi alterado.

14 REFERÊNCIAS BIBLIOGRÁFICAS

DE PAULA, M. B. **Reconhecimento de palavras faladas utilizando Redes Neurais Artificiais**. Pelotas, dezembro de 2002. Arquivo (1472 Kbytes); Internet (Disponível em: <http://www.ufpel.tche.br/prg/sisbi/bibct/acervo/info/2000/Mono-MauricioBdePaula.pdf>). Documento Adobe Acrobat.

RAITZ, R. T. **Free Associative Neurons – FAN: Uma abordagem para Reconhecimento de Padrões**. Florianópolis, dezembro de 1997. Disponível em: <http://www.eps.ufsc.br/disserta98/raitz/>.

ISKANDAR, J. I. **Normas da ABNT comentadas para trabalhos científicos**. Curitiba: Champagnat, 2000.

FURLAN, J. D. **Modelagem de Objetos através da UML**. São Paulo: Makron Books, 1998.

CANTÚ, M. **Dominando o Delphi 7: A Bíblia**. São Paulo: Makron Books, 2003.

MATSUMOTO, E. Y. **Matlab 6.5 : Fundamentos de Programação**. São Paulo: Érica, 2002.

NEVES, F. L. D. **PostgreSQL** : Conceitos e Aplicações. São Paulo: Érica, 2002.

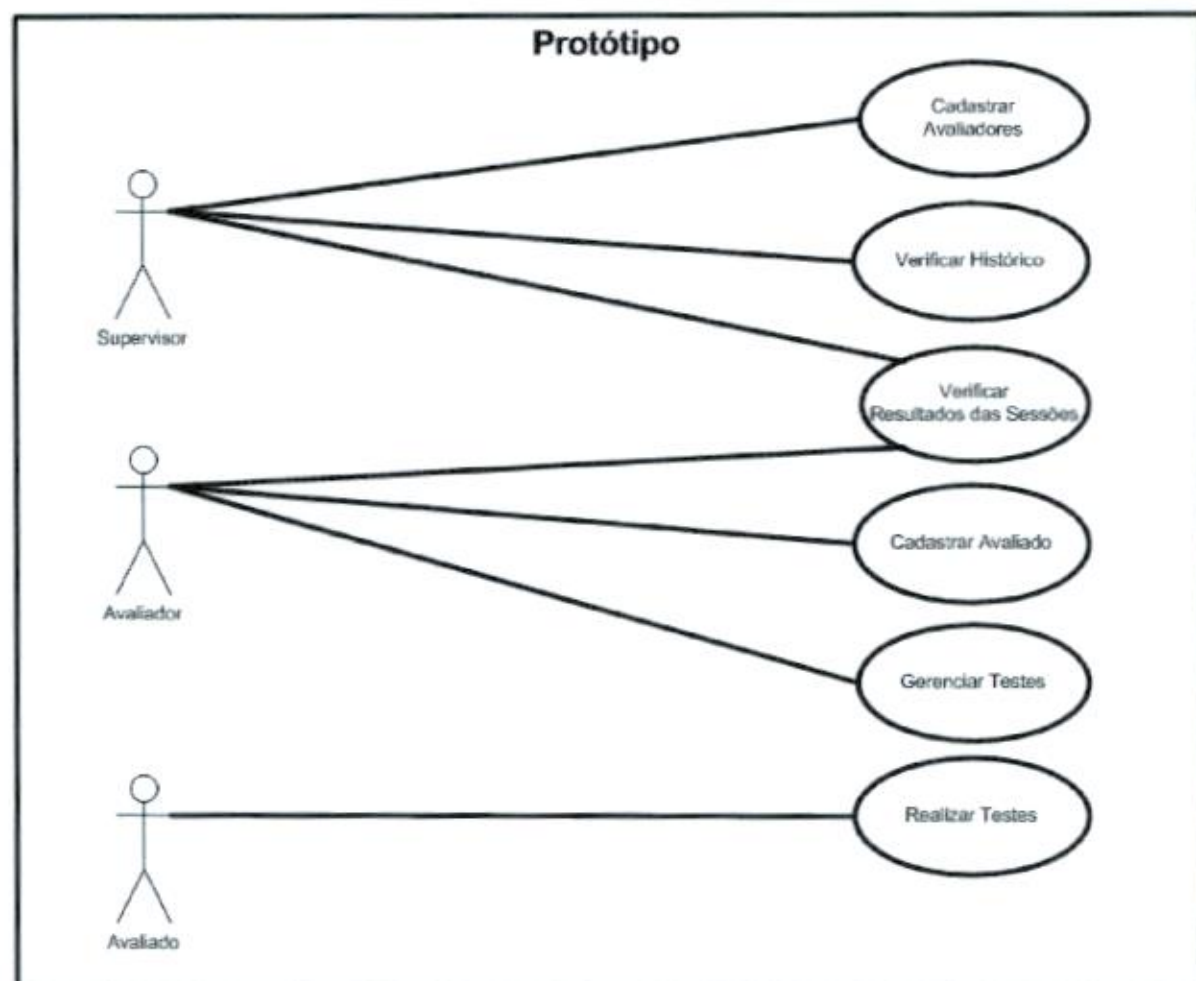
KNOTTINGHAM, E. **Aprenda Microsoft Visio 2002**. São Paulo: Campus, 2001.

LOPES, B. S. **Modelo para medição de recepção e produção da linguagem verbal.doc**. Arquivo (3488 Kbytes); Microsoft Word 97.

ANEXOS

15.1 ANEXO I
MODELAGEM DO PROTÓTIPO

DIAGRAMA DE CASOS DE USO – PROTÓTIPO



DESCRIÇÃO RESUMIDA PROTÓTIPO

CASOS DE USO	QUEM INICIA	DESCRIÇÃO
Cadastrar Avaliadores	Supervisor	Neste caso de uso o Supervisor cadastrar o(s) avaliador(es).
Cadastrar Avaliado	Avaliador	Neste caso de uso o Avaliador cadastra o(s) avaliado(s).
Realizar Testes	Avaliado	Neste caso de uso o(s) Avaliador(es) aplica(m) o(s) teste(s) ao avaliado.
Verificar Resultados das Sessões	Supervisor / Avaliador	Neste caso de uso o Supervisor ou Avaliador(es) consultam o resultado de uma determinada sessão dos testes.
Verificar Histórico	Supervisor / Avaliador	Neste caso de uso o Supervisor ou Avaliador consultam os resultados de todas as sessão dos testes, ou seja, o resultado geral, gerado após a realização de todos os testes .
Gerenciar Testes	Avaliador	Neste caso de uso, o Avaliador acompanha e gerência todo o processo da aplicação do(s) teste(s).

FLUXO DE EVENTOS PROTÓTIPO.

CASO DE USO: CADASTRAR AVALIADORES

Pré-condições: Nenhuma.

FLUXO PRINCIPAL

Este caso de uso se inicia quando o usuário seleciona na tela principal (tela de menu) a opção *aplicador* (no menu de cadastro), abrindo assim, a tela de cadastro de aplicador onde ele faz a inserção dos dados correspondentes ao usuário que será cadastrado. Em seguida ele terá as seguintes alternativas: GRAVAR, EXCLUIR, PESQUISAR e SAIR. Caso opção escolhida seja:

GRAVAR (A-1): executa o sub-fluxo "gravar dados do avaliador"

EXCLUIR (A-2): executa o sub-fluxo "excluir avaliador"

PESQUISAR (A3): executa o sub-fluxo "pesquisar avaliadores cadastrados"

SAIR (A-4): executa o sub-fluxo "cancelar a operação"

CASO DE USO: CADASTRAR AVALIADO

Pré-condições: Nenhuma.

FLUXO PRINCIPAL

Este caso de uso se inicia quando o usuário seleciona na tela principal (tela de menu) a opção *aluno* no menu de cadastro. O sistema abre a tela de cadastro de avaliado onde o usuário insere os dados correspondentes. Nesse caso de uso o

sistema oferece ao usuário as seguintes alternativas: GRAVAR, EXCLUIR, PESQUISAR E SAIR. Caso opção escolhida seja:

GRAVAR (A-5): executa o sub-fluxo "gravar dados do avaliado"

EXCLUIR (A-6): executa o sub-fluxo "excluir avaliado"

PESQUISAR (A-7): executa o sub-fluxo "pesquisar avaliados cadastrados"

SAIR (A-8): executa o sub-fluxo "cancelar a operação"

CASO DE USO: REALIZAR TESTES

Pré-condições: Devem ter sido cadastrados: local, aplicador e avaliado.

FLUXO PRINCIPAL

Este caso de uso inicia quando o aplicador seleciona um teste no menu de testes 2.1-Produção Oral de Palavras (A-9).

CASO DE USO: VERIFICAR RESULTADOS DAS SESSÕES

Pré-condições: Aplicação completa de pelo menos um Teste

FLUXO PRINCIPAL

Esse caso de uso se inicia quando o aplicador seleciona no menu de consultas a opção *resultados* que abre a tela de *resultado* dos testes aplicados. O usuário seleciona o campo a partir do qual deseja fazer a consulta e insere os dados no campo de pesquisa. O usuário clica na opção *pesquisar* e o sistema recupera os dados correspondentes a pesquisa do usuário e trás para a tela (E-1). O sistema

aguarda que o usuário feche a tela ou clique no botão CANCELAR para que o caso de uso se encerre.

CASO DE USO: VERIFICAR HISTÓRICO

O usuário seleciona a opção *histórico* no menu de consultas, na tela de menus. O sistema abre a tela de histórico onde o usuário seleciona o campo a partir do qual deseja fazer a consulta. O usuário preenche o campo de pesquisa e em seguida clica no botão PESQUISAR ou ainda (E-1).. O sistema recupera os dados procurados pelo usuário e trás para a tela O sistema aguarda que o usuário clique no botão CANCELAR ou feche a tela de histórico, e o caso de uso se encerra.

CASO DE USO: GERENCIAR TESTES

Pré-condições: Avaliado realizando teste.

Este caso de uso se inicia quando o avaliado inicia o teste, onde o avaliador gerência o teste através da aplicação de monitoramento.

FLUXOS ALTERNATIVOS

A-1: Gravar dados do Avaliador

O usuário preenche os campos (nome, nome para o sistema, login, senha e confirmação de senha) (E-2) e em seguida clica no botão GRAVAR e assim os

referidos dados serão gravados no cadastro de aplicador. O caso de uso pode ser encerrado ou continuar.

A-2: Excluir aplicador

O usuário clica no botão EXCLUIR que trás a listagem dos aplicadores cadastrados para que o usuário possa selecionar aquele que deseja excluir. O sistema aguarda a confirmação ou cancelamento da exclusão. O caso de uso pode ser encerrado ou continuar.

A-3: Pesquisar aplicadores cadastrados

O usuário clica no botão PESQUISAR da tela de cadastro que lhe trará a tela de pesquisa de aplicadores, o usuário preenche o campo de pesquisa e clica no botão pesquisar (E-3), o sistema traz as informações referentes ao aplicador pesquisado(E-4). O usuário fecha a tela de pesquisa ou clica no botão CANCELAR e o caso de uso pode ser encerrado ou continuar.

A-4: Cancelar a operação

Após abrir a tela de cadastro de aplicador o usuário poderá a qualquer momento encerrar o caso de uso clicando no botão SAIR.

A-5: Gravar dados do avaliado

O usuário preenche os campos (nome completo, nome para o sistema, nome da mãe, nome do pai) e em seguida clica no botão GRAVAR (E-7) e efetua o cadastro. O caso de uso pode ser encerrado ou continuar.

A-6: Excluir avaliado

O usuário clica no botão EXCLUIR que trás a listagem dos avaliados cadastrados no sistema. O usuário seleciona a opção correspondente ao avaliado que deseja excluir. O sistema aguarda confirmação ou cancelamento da exclusão. O caso de uso pode ser encerrado ou continuar.

A-7: Pesquisar avaliados cadastrados

O usuário clica em PESQUISAR na tela de cadastro de avaliado que lhe trará a tela de pesquisa de avaliados, onde ele seleciona um campo a partir do qual deseja fazer a pesquisa em seguida preenche o campo com os dados correspondentes, o usuário clica em PESQUISAR e o sistema trás os dados referentes ao avaliado procurado(E-8). O usuário fecha a tela de pesquisa ou clica em CANCELAR e o caso de uso se encerra ou continua.

A-8: Cancelar a operação

Após abrir a tela de cadastro de avaliado o usuário poderá a qualquer momento encerrar o caso de uso clicando no botão SAIR.

A-9: : Teste 2.1

O usuário seleciona o *Teste Produção Oral de Palavras* e o sistema trás as cartelas para aplicação do teste (cartelas 1 a 15). O sistema realça uma figura da cartela e o avaliado fala (no microfone) o nome da figura representa na quadrícula. O usuário ainda terá as seguintes opções: RETOMAR UM TESTE INTERROMPIDO, INTERROMPER TESTE, SALVAR RESULTADO. Caso a opção escolhida seja:

RETOMAR TESTE INTERROMPIDO (A-10): Executa o sub-fluxo "Retomar um Teste interrompido"

INTERROMPER TESTE (A-11): Executa o sub-fluxo " Interromper Teste"

SALVAR RESULTADO: (A-12): Executa o sub-fluxo "Salvar resultado do Teste"

CANCELAR TESTE (A-13)

A-10: Retomar um Teste Interrompido

O usuário seleciona no menu de arquivo, na tela de aplicação do Teste, a opção Retomar Teste. O sistema trás uma tela com uma relação de testes interrompidos e o usuário seleciona o Teste correspondente ao avaliado que estiver retomando a aplicação. O sistema recupera os dados exatamente no ponto onde o

Teste foi interrompido e trás para a tela. O usuário prossegue a aplicação do Teste.
O caso de uso continua.

A-11: Interromper Teste

O avaliador pressiona a tecla F12 e seleciona no menu de arquivo, na tela de aplicação do Teste, a opção Interromper Teste, o sistema salva todos os dados referentes à aplicação do teste e o exato ponto onde foi interrompido para que possa ser retomado no momento oportuno. O caso de uso é encerrado.

A-12: Salvar Resultado do Teste

O avaliador pressiona a tecla F12 e seleciona (depois de completado a aplicação de um teste) no menu arquivo, da tela de aplicação do Teste, a opção Salvar Resultados. O sistema trás a tela de resultados com os dados referentes à aplicação em questão e aguarda a confirmação do usuário. O usuário clica no botão SALVAR e o caso de uso se encerra.

A-13: Cancelar Teste

O avaliador pressiona a tecla F12 e seleciona no menu arquivo, da tela de aplicação do Teste, a opção Cancelar teste, e o sistema aguarda a confirmação do usuário. O usuário clica no botão e o caso de uso se encerra.

FLUXOS DE EXCEÇÃO

E-1: O sistema informa erro na inserção de dados. O usuário pode corrigir e continuar o caso de uso.

E-2: É informado erro na confirmação da senha. O usuário pode corrigir ou encerrar o caso de uso.

E-3: O sistema informa que o campo de pesquisa não foi preenchido. O usuário poderá continuar o caso de uso fazendo a correção ou encerrar o caso de uso.

E-4: É informado que o nome procurado não foi encontrado. O usuário pode fazer nova tentativa ou encerrar o caso de uso.

E-5: O sistema informa erro pelo não preenchimento de todos os campos. O usuário providencia a correção e retorna à execução do sub-fluxo. O caso de uso continua.

E-6: O sistema informa que o nome do local procurado não consta no cadastro. O usuário poderá continuar o caso de uso efetuando o cadastro do local ou encerrar o caso de uso.

E-7: É informado erro pelo preenchimento incorreto dos campos ou pelo não preenchimento de algum campo. O usuário pode fazer a correção. O caso de uso continua.

E-8: O sistema informa não encontrar o nome procurado. O caso de uso poderá continuar efetuando um novo cadastro ou poderá ser encerrado.

CENÁRIOS PROTÓTIPO

CASO DE USO: CADASTRAR AVALIADORES

CENÁRIO PRINCIPAL

O supervisor José da Silva seleciona a opção *aplicador* no menu de cadastro da tela principal, e insere os dados do cadastro nos respectivos campos (nome: João da Silva, nome para o sistema: João da Silva, login: aplicador, senha: aplicador, e confirma senha: aplicador) na tela de cadastro de aplicador e em seguida clica no botão gravar para efetuar o cadastro do aplicador.

CENÁRIOS SECUNDÁRIOS

O sistema informa ocorrência de erro na confirmação de senha. O usuário então executa a correção.

Não foram preenchidos todos os campos no cadastro e ao tentar gravar os dados o sistema informa a ocorrência e o usuário completa a inserção de dados.

O usuário tenta realizar uma pesquisa, o sistema informa que o campo de pesquisa não foi preenchido. O usuário preenche o campo e prossegue a pesquisa.

O sistema informa não existir registro referente ao aplicador procurado numa operação de pesquisa, o usuário efetua o cadastro para o determinado aplicador.

CASO DE USO: CADASTRAR AVALIADO

CENÁRIO PRINCIPAL

O supervisor José da Silva seleciona a opção *avaliado* no menu de cadastro da tela principal e em seguida insere os referidos dados nos respectivos campos na tela de cadastro de avaliado (nome: Maria de Sá, nome para o sistema: Maria de Sá, nome da mãe: Ana de Sá, nome do pai: Pedro de Sá) e em seguida conclui o cadastro clicando no botão GRAVAR.

CENÁRIOS SECUNDÁRIOS

Ao se realizar pesquisa dentre os avaliados cadastrados no sistema, o nome do avaliado Pedro Manoel não é encontrado. Ao verificar-se que Pedro Manoel não está cadastrado no sistema, o usuário poderá efetuar o cadastro de Pedro Manuel.

Ao se tentar efetuar cadastro para José dos Santos, o sistema informa não completar a operação pelo não preenchimento de todos os campos no cadastro de avaliado, o usuário faz as correções necessárias e efetua o cadastro de José dos Santos.

CASO DE USO: CONSULTAR HISTÓRICO

CENÁRIO PRINCIPAL

O supervisor José da Silva seleciona a opção *histórico* no menu de consulta, ao clicar sobre essa opção abre a tela de histórico, onde o José da Silva seleciona a opção local como campo de pesquisa. Fornece o nome: Escola Técnica da UFPR e clica sobre o botão PESQUISAR que trás a relação de pessoas avaliadas nesse local: Maria de Sá, Pedro de Souza e Ana dos Santos. José da Silva seleciona Maria de Sá e o sistema trás os dados referentes aos testes aplicados para Maria de Sá e seus respectivos resultados: Teste 1.1 – não apresentou erros, Teste 2.1 – não apresentou erros.

CENÁRIOS SECUNDÁRIOS

O sistema informa ocorrência de erro durante a tentativa de se realizar uma pesquisa, então o usuário poderá providenciar a correção.

O avaliado cadastrado no sistema não completou a execução de um nenhum teste, sendo assim não possui dados no histórico.

CASO DE USO: VERIFICAR RESULTADOS DAS SESSÕES

CENÁRIO PRINCIPAL

José da Silva seleciona a opção resultados no menu de consulta na tela de menu e abre a tela de consulta de resultados, onde ele seleciona a opção avaliado como campo de pesquisa. José da Silva fornece o nome: Maria de Sá, e o sistema retorna uma relação dos testes aplicados para Maria de Sá: Teste 1.1 e Teste 2.1. José da Silva seleciona o Teste 1.1 e o sistema retorna o resultado obtido por Maria de Sá : Teste 1.1- não apresentou erros.

CENÁRIOS SECUNDÁRIOS

O avaliado possui cadastro, mas não possui registro de resultados por não Ter sido completada a aplicação de nenhum Teste para o referido avaliado.

O nome Paulo da Silva não foi encontrado. O sistema informa o usuário e ele faz as correções necessárias para realizar a pesquisa.

CASO DE USO: REALIZAR TESTES

CENÁRIO PRINCIPAL

José da Silva seleciona o teste seleciona o Teste 2.1 no menu de Testes na tela de menu. O sistema abre a primeira tela do teste (cartela 1). José da Silva

apresenta as figuras da cartela para Maria de Sá (avaliada). José da Silva clica sobre cada uma das 6 figuras da cartela (na ordem em que ela aparecem em sua folha de anotações) dizendo o nome da figura para Maria da Sá: bode, bote, faca, vaca, gato e gado. Passa para a Segunda cartela: doce, doze, calo, galo, queixo e queijo. Passa para a terceira cartela: porta, torta, cama, cana, faca e saca. Quarta cartela: espada, escada, bola, gola, folha e rolha. Quinta cartela: porta, porca, sopa, roupa, vela e velha. Sexta cartela: pilha, filha, taco, saco, caco e carro. Sétima cartela: barata, batata, roda, rosa, dente e lente. Oitava cartela: bola, mola, pêra, pena, casa e cana. Nona cartela: calo, cano, pilha, pinha, dado e nado. Décima cartela: vala, vara, muleta, mureta, pelada e perada. Décima primeira cartela: bicho, bucho, pente, ponte, reza e rosa. Décima Segunda cartela: piso, peso, linha, lenha, murro e morro. Décima terceira cartela: feira, fera, vovô, vovó, touca e toca. Décima Quarta cartela: mola, mala, bola, bala, sola e sala. Décima Quinta cartela: trança, traça, trens, três concha e coxa. José da Silva retorna a primeira cartela, fala o nome de uma figura e Maria de Sá clica sobre a figura correspondente àquela que José da Silva se referiu. Repetem o procedimento para todas as figuras da cartela (na ordem em que elas aparecem na folha de anotações) e passam para a próxima cartela, e assim sucessivamente até que cheguem novamente à cartela de número 15. Maria da Sá erra as figuras: pilha, linha e bola. José da Silva anota os erros no espaço correspondente a essas figuras na folha de anotações. José da Silva abre o menu arquivo na tela de aplicação do teste e seleciona a opção salvar resultado do Teste. O sistema trás os dados referentes à aplicação do teste 1.1 para Maria de Sá. José da Silva clica em *salvar*. O sistema efetua a gravação dos dados.

Obs: Para os devidos fins será implementada apenas a primeira cartela do teste.

CENÁRIOS SECUNDÁRIOS

Não foi efetuado o cadastro de avaliado e/ou local e é feita uma tentativa de realização de teste. O sistema informa ao usuário que faz a correção efetuando o(s) referido(s) cadastro(s) e prosseguindo a operação.

CASO DE USO: GERENCIAR TESTES

José da Silva (avaliador) através da aplicação de monitoramento acompanha e verifica o teste que Maria de Sá (avaliada) está realizando.

DIAGRAMA DE TELAS – PROTÓTIPO

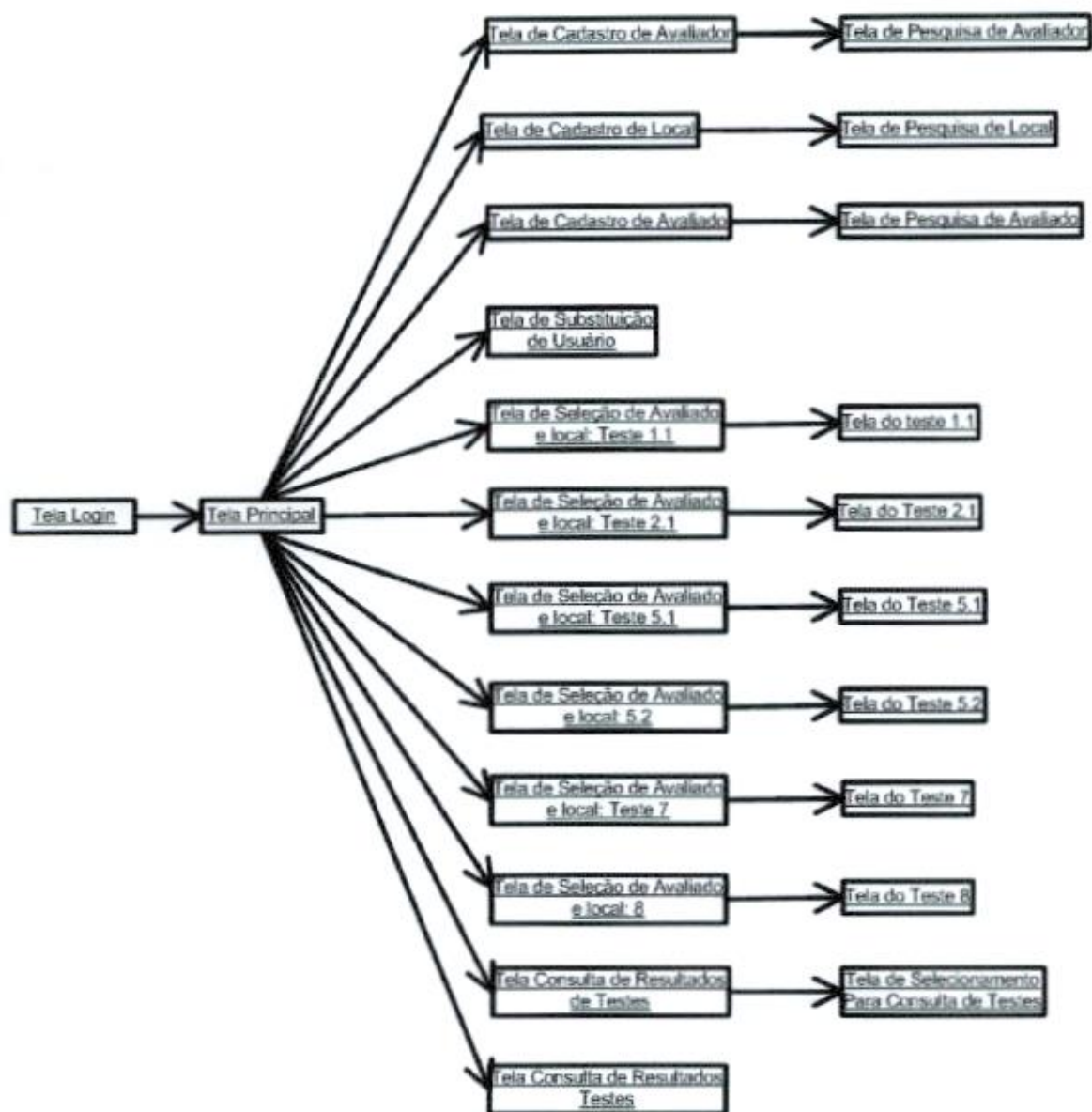


DIAGRAMA DE CLASSES

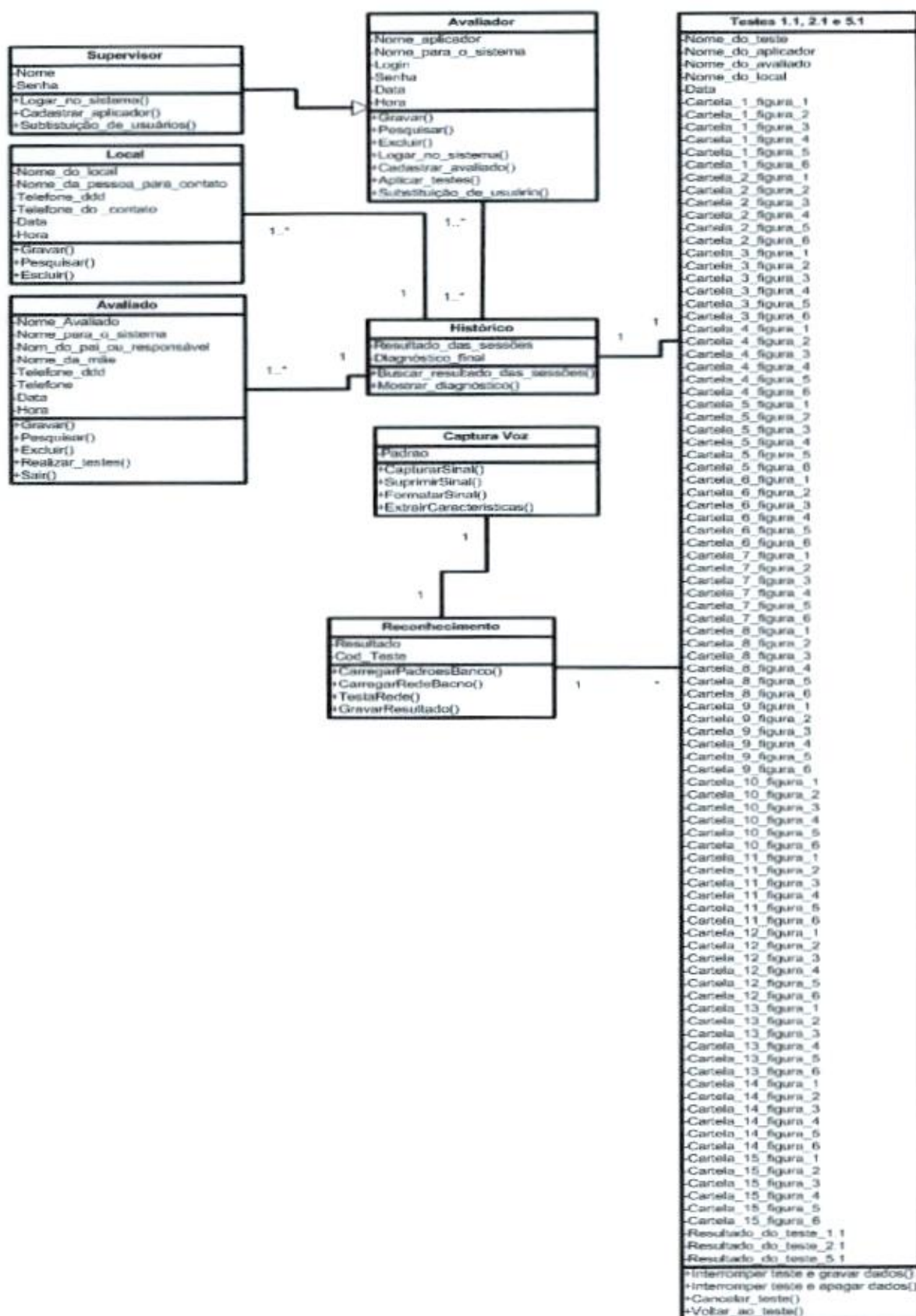


DIAGRAMA DE OBJETOS DO PROTÓTIPO

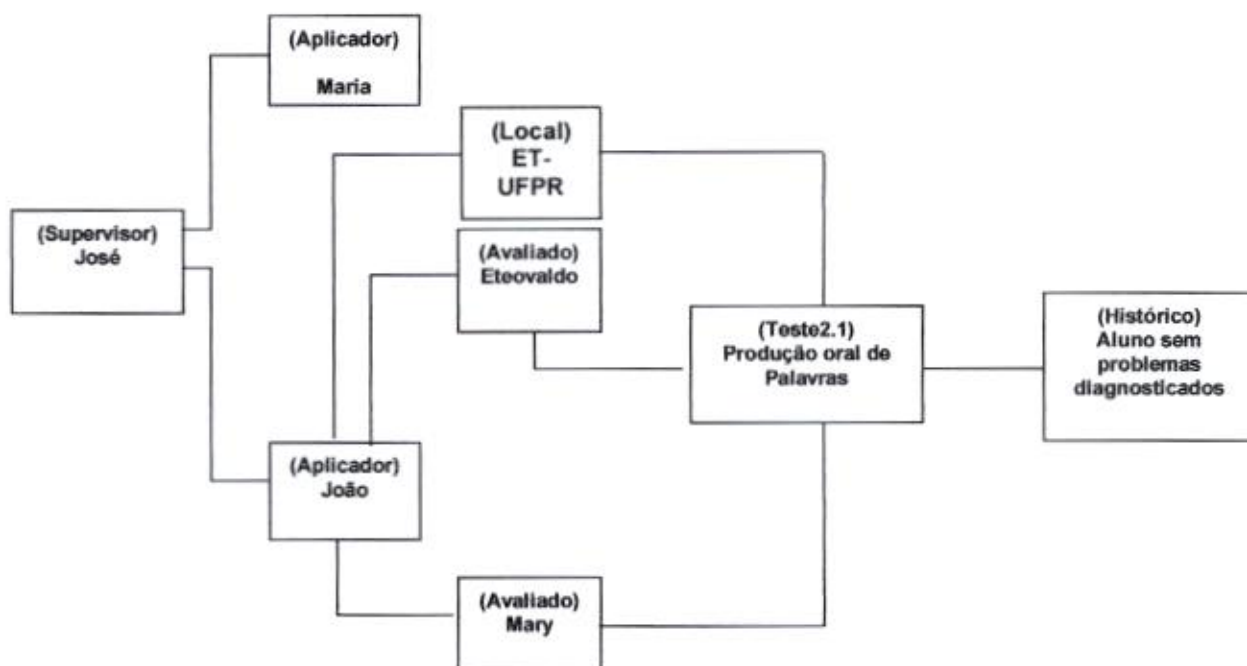


DIAGRAMA DE SEQÜÊNCIA - PROTÓTIPO - CADASTRAR AVALIADO

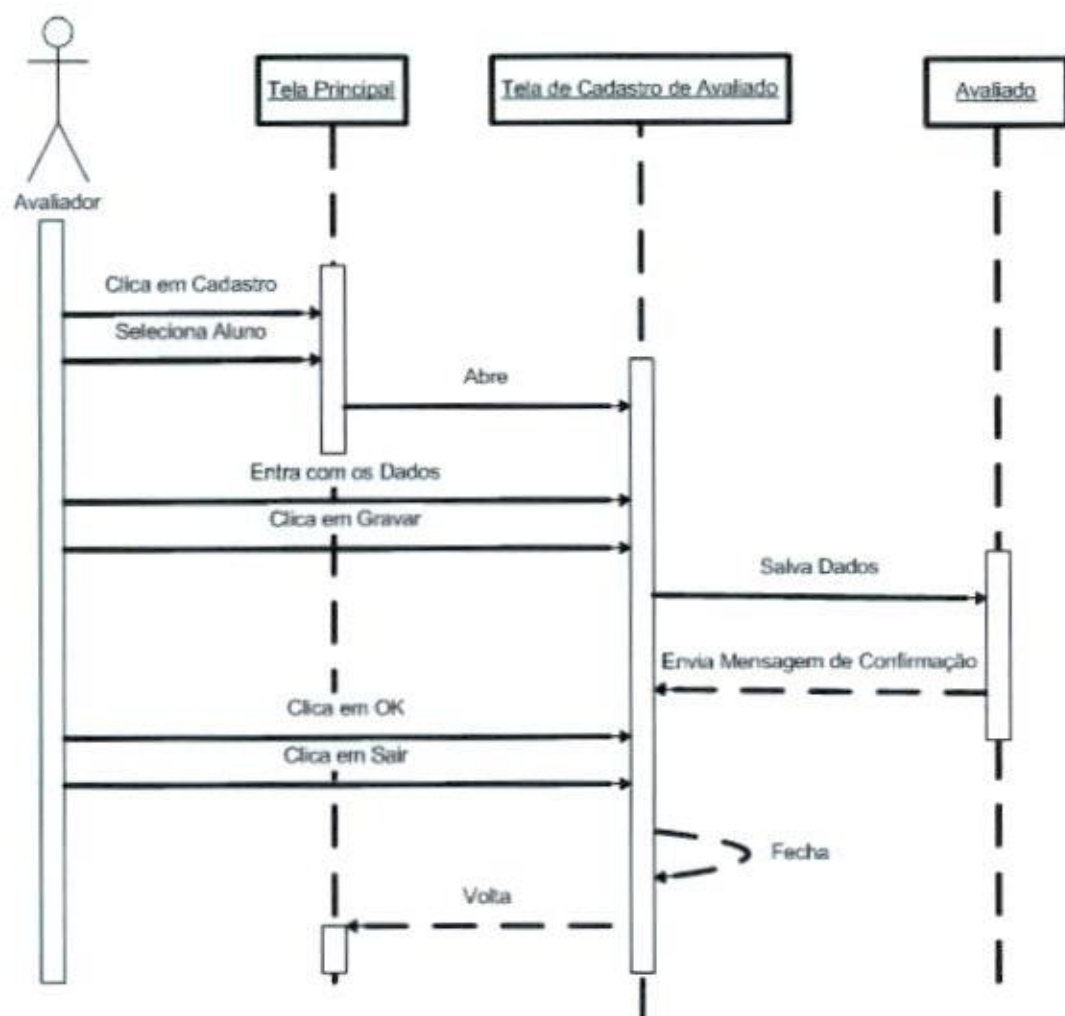


DIAGRAMA DE SEQÜÊNCIA - PROTÓTIPO - ALTERAR DADOS AVALIADO

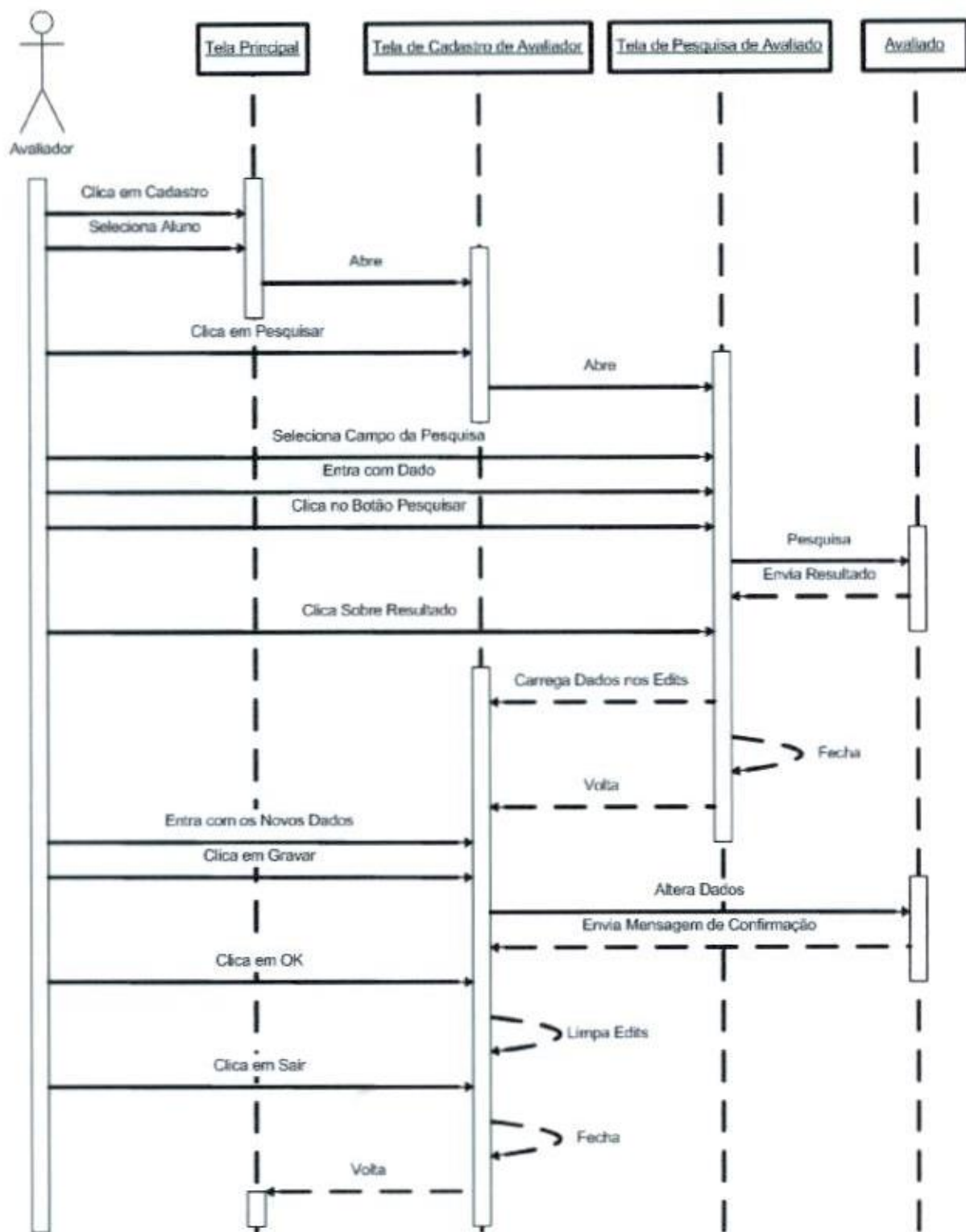


DIAGRAMA DE SEQÜÊNCIA - PROTÓTIPO - EXCLUIR AVALIADO

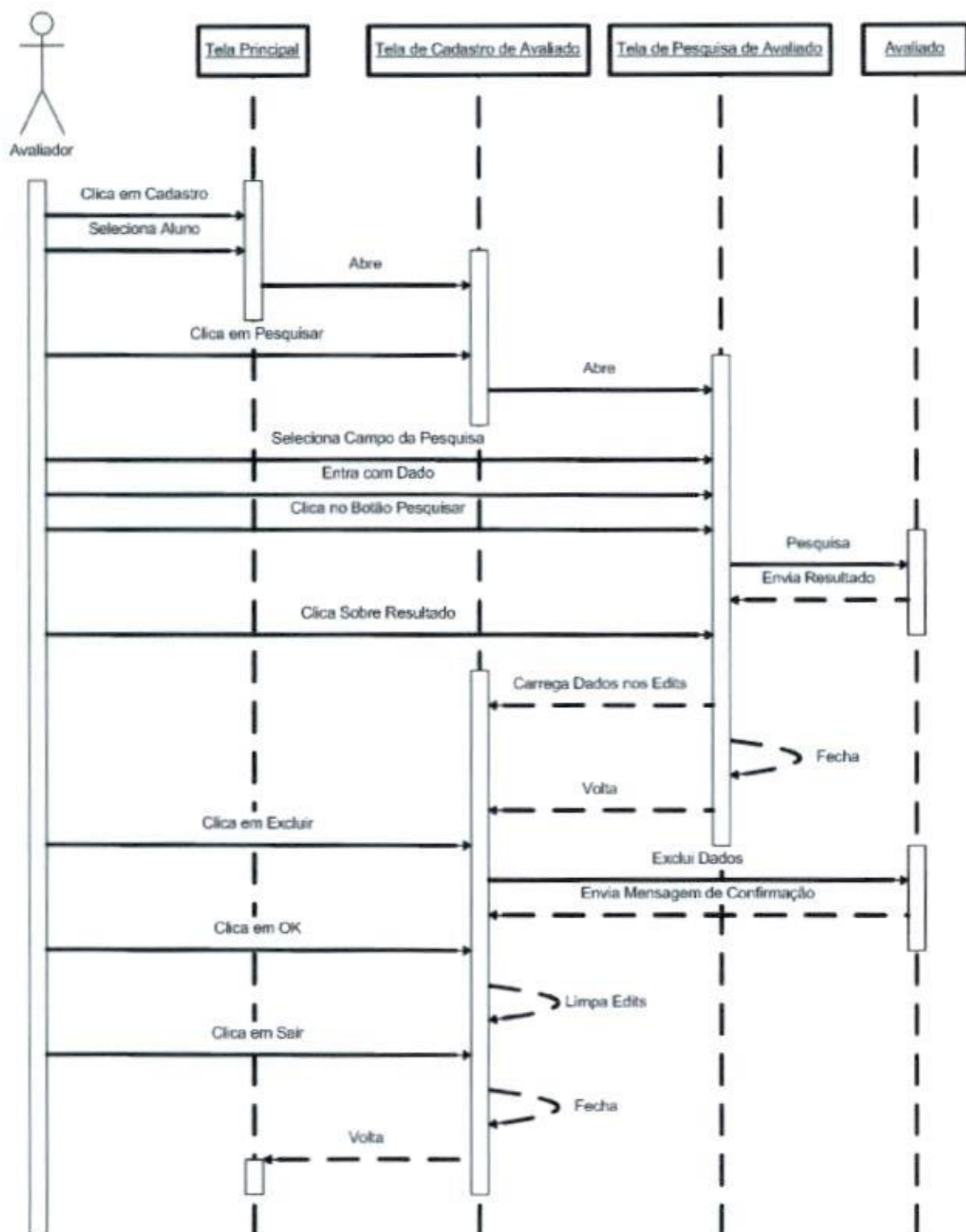


DIAGRAMA DE SEQÜÊNCIA - PROTÓTIPO - CADASTRAR AVALIADOR

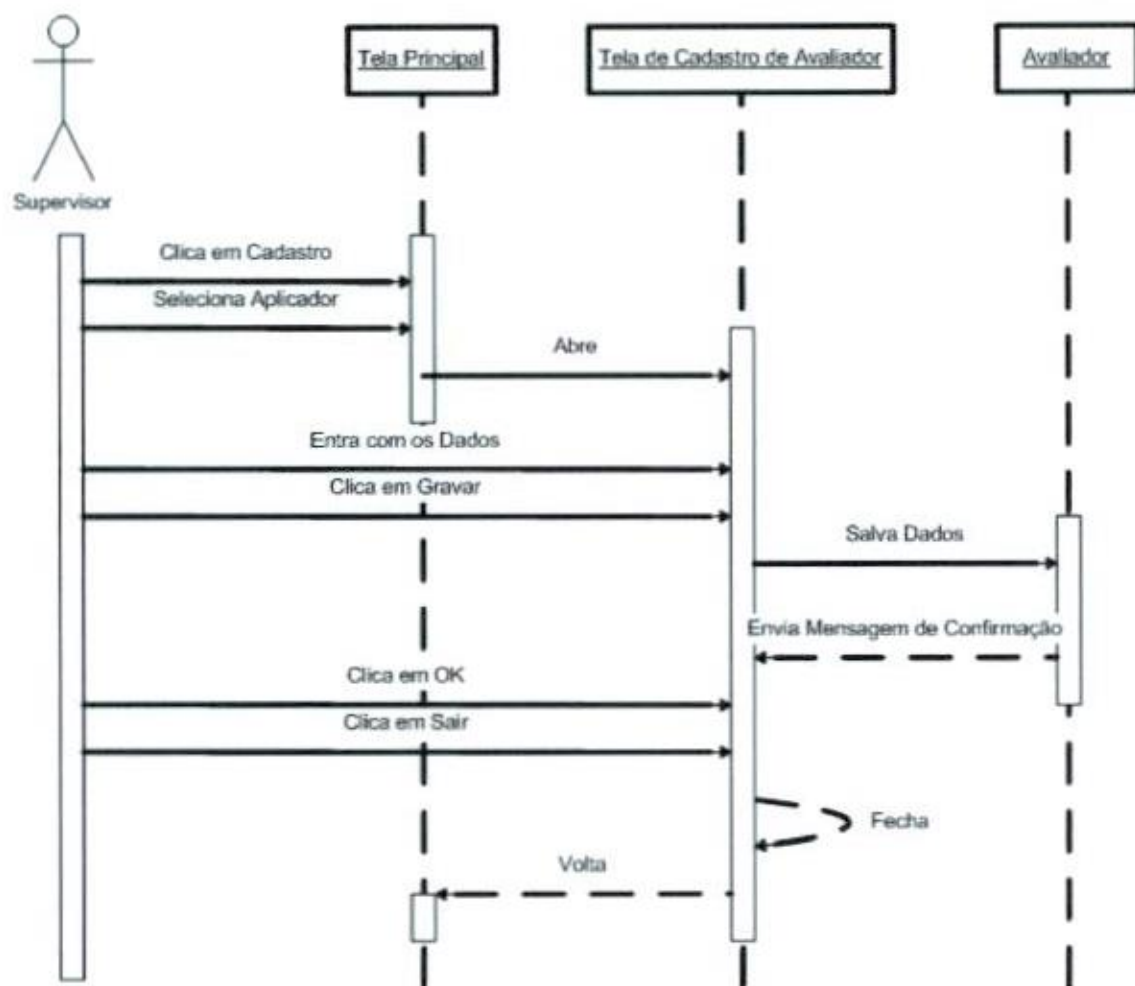


DIAGRAMA DE SEQÜÊNCIA - PROTÓTIPO - ALTERAR DADOS AVALIADOR

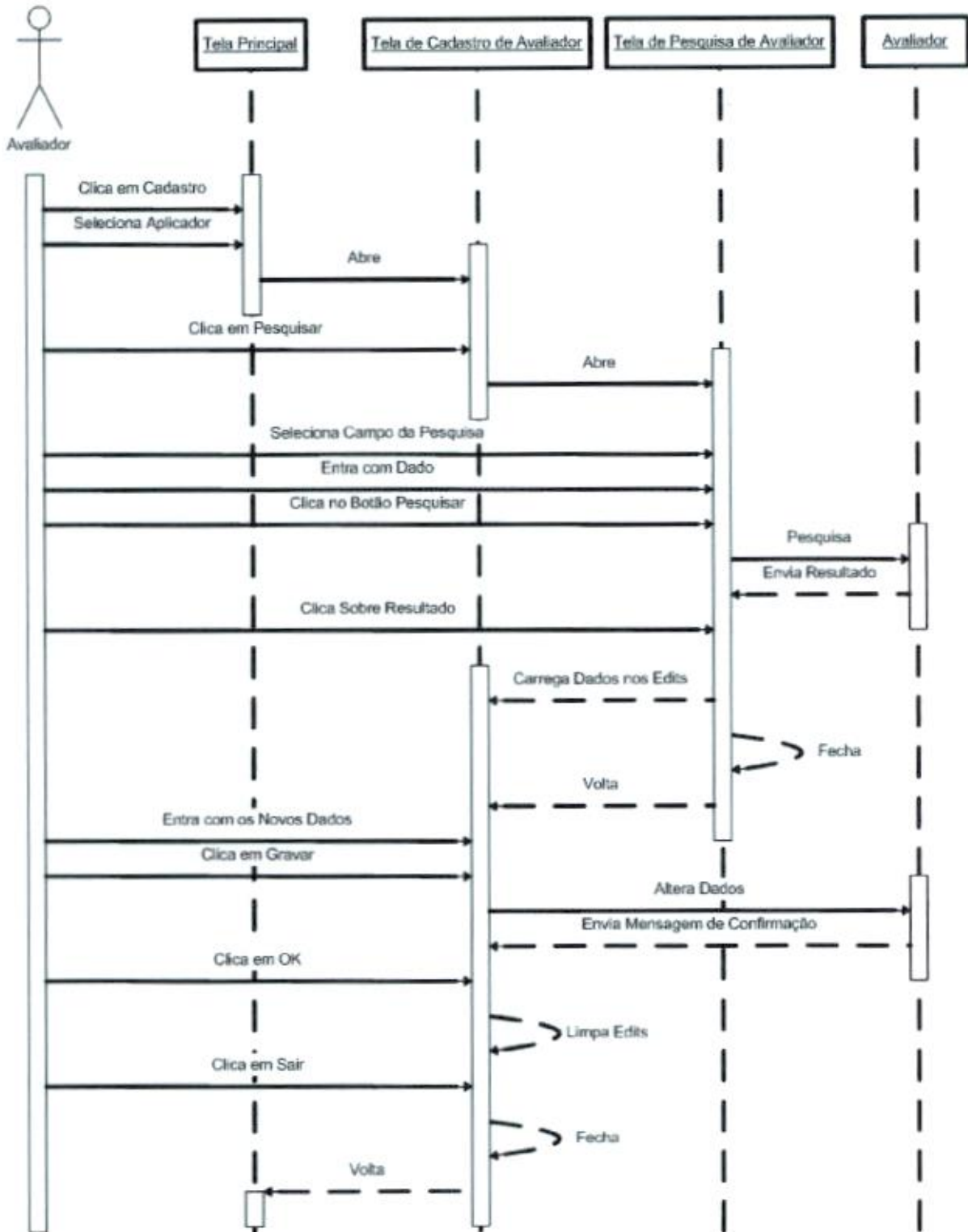


DIAGRAMA DE SEQÜÊNCIA - PROTÓTIPO - EXCLUIR AVALIADOR

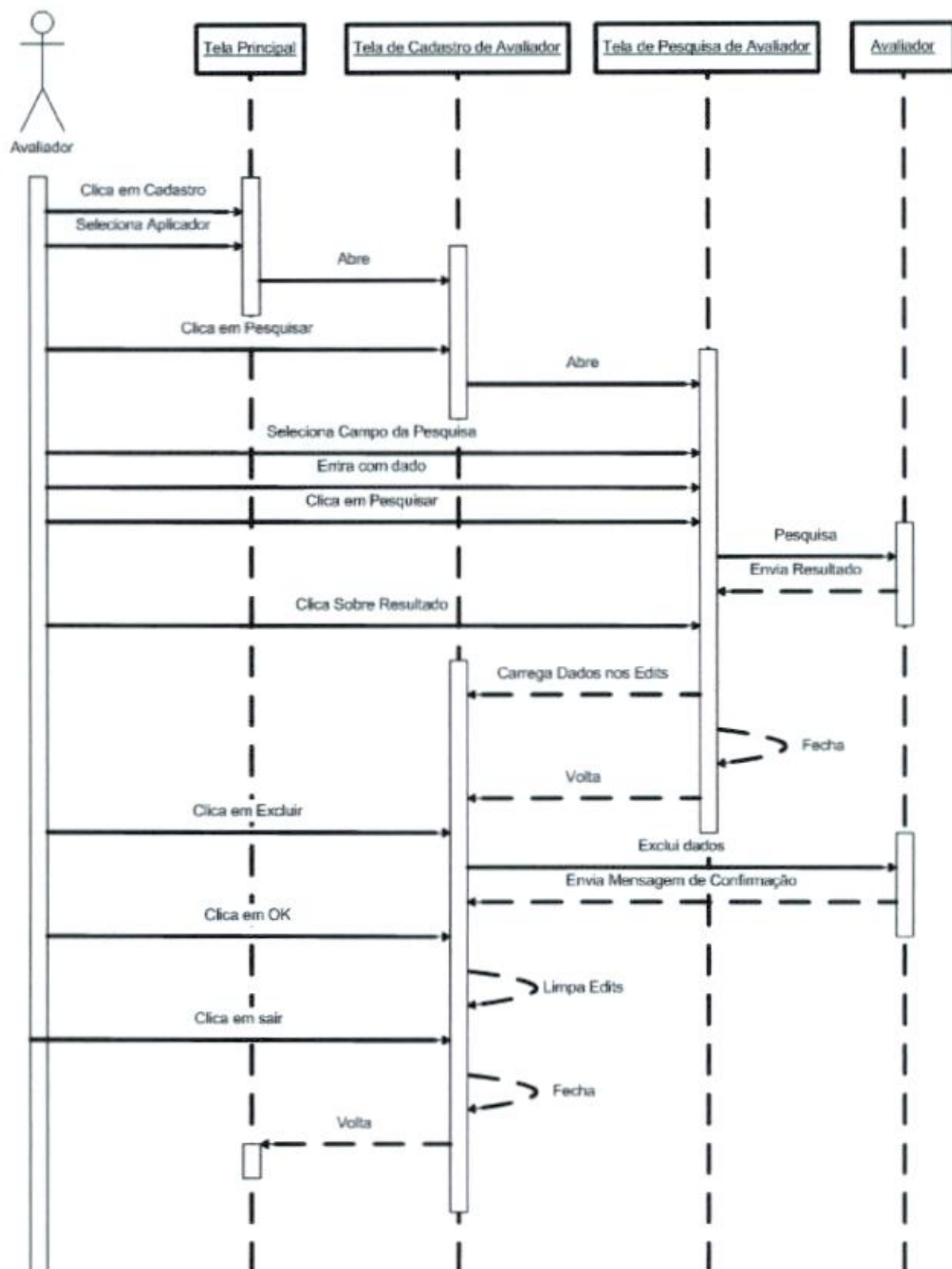


DIAGRAMA DE SEQÜÊNCIA - PROTÓTIPO - CADASTRAR LOCAL

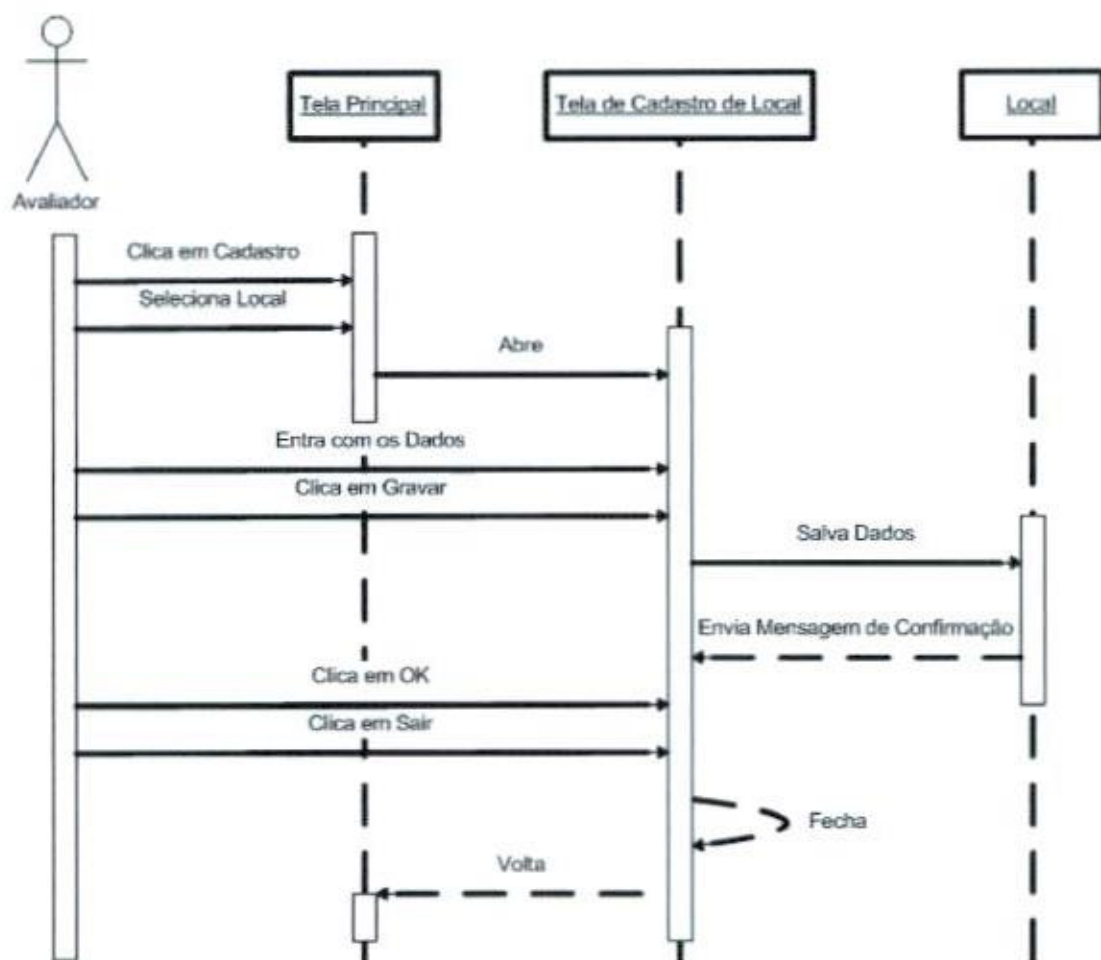


DIAGRAMA DE SEQÜÊNCIA - PROTÓTIPO - ALTERAR DADOS LOCAL

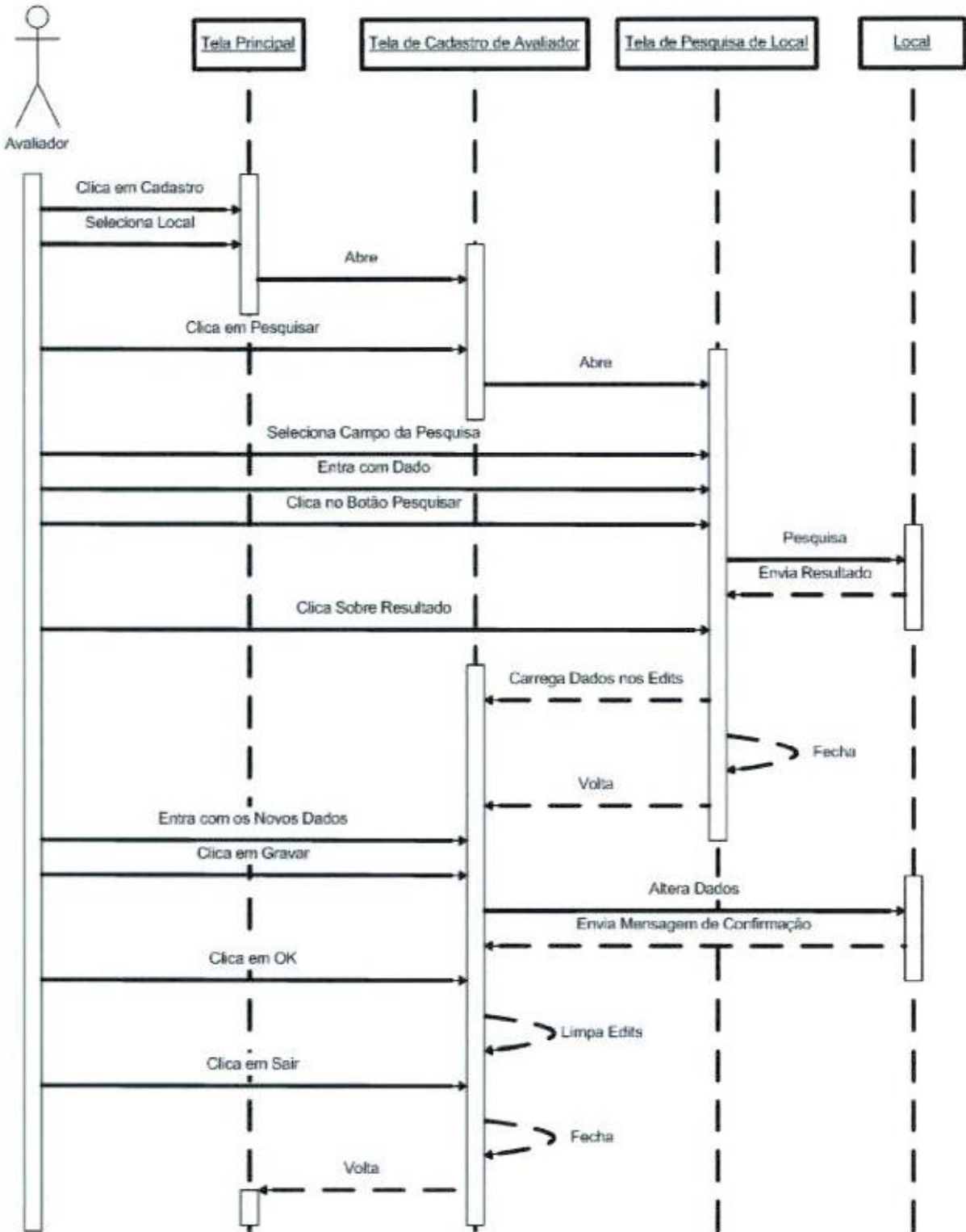


DIAGRAMA DE SEQÜÊNCIA - PROTÓTIPO - EXCLUIR LOCAL

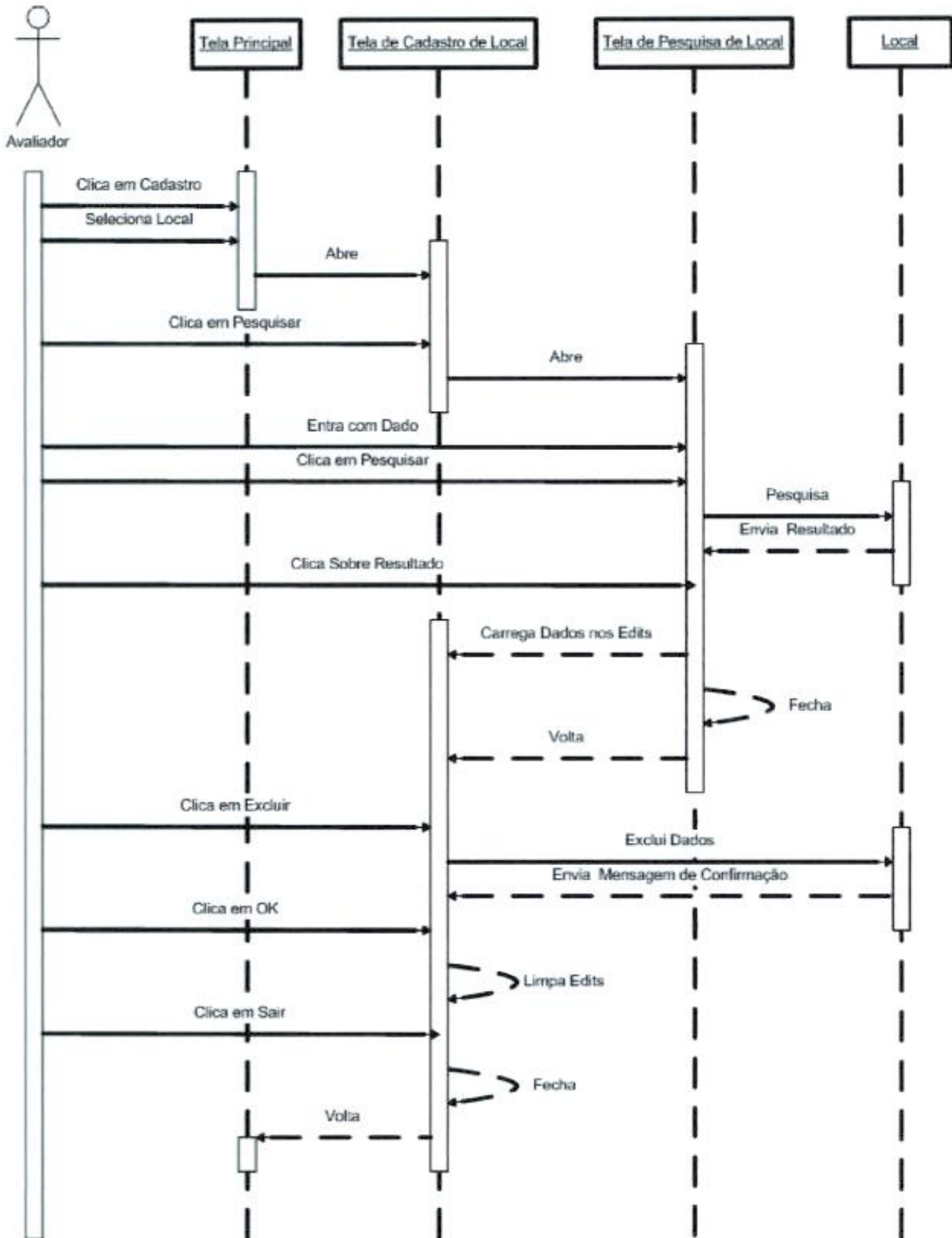


DIAGRAMA DE SEQÜÊNCIA - PROTÓTIPO - LOGAR NO SISTEMA

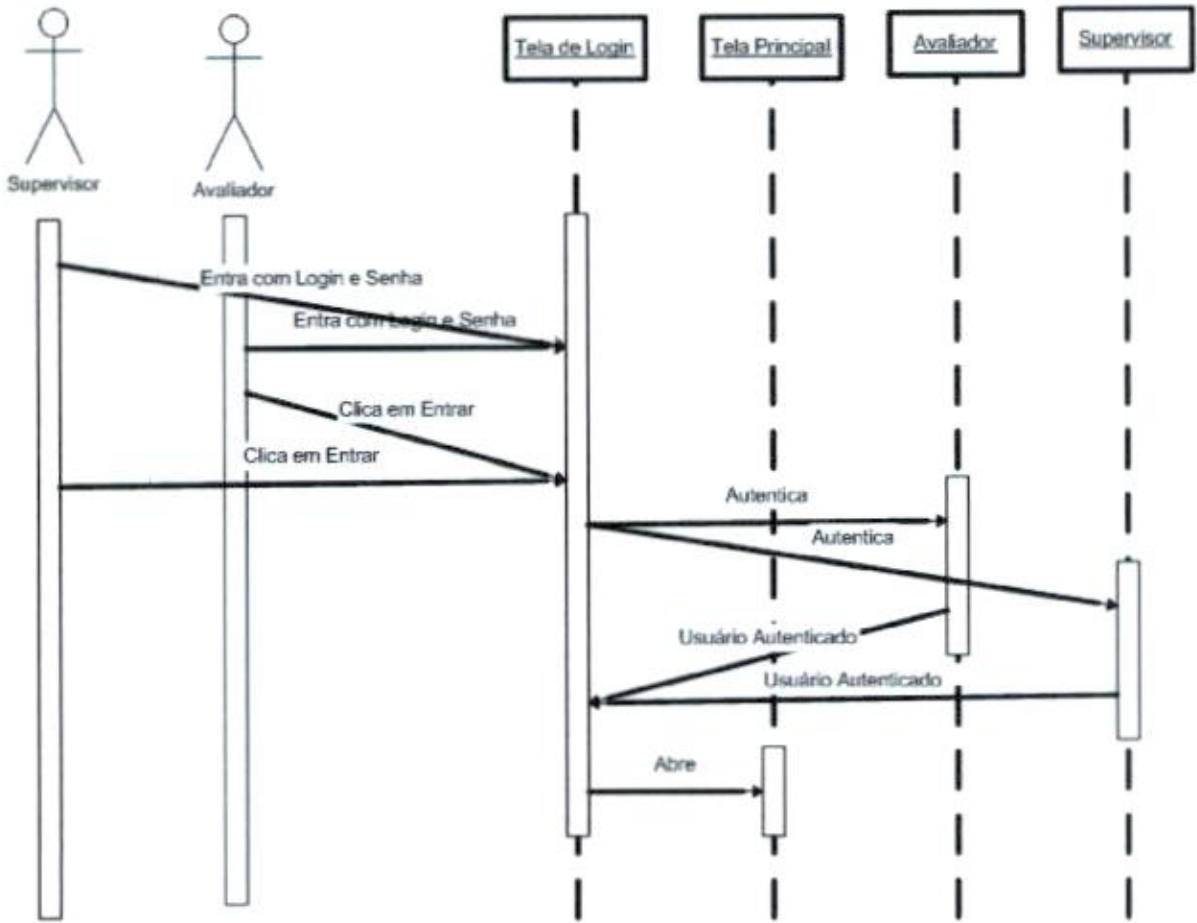


DIAGRAMA DE SEQÜÊNCIA - PROTÓTIPO - SUBSTITUIÇÃO DE USUÁRIO

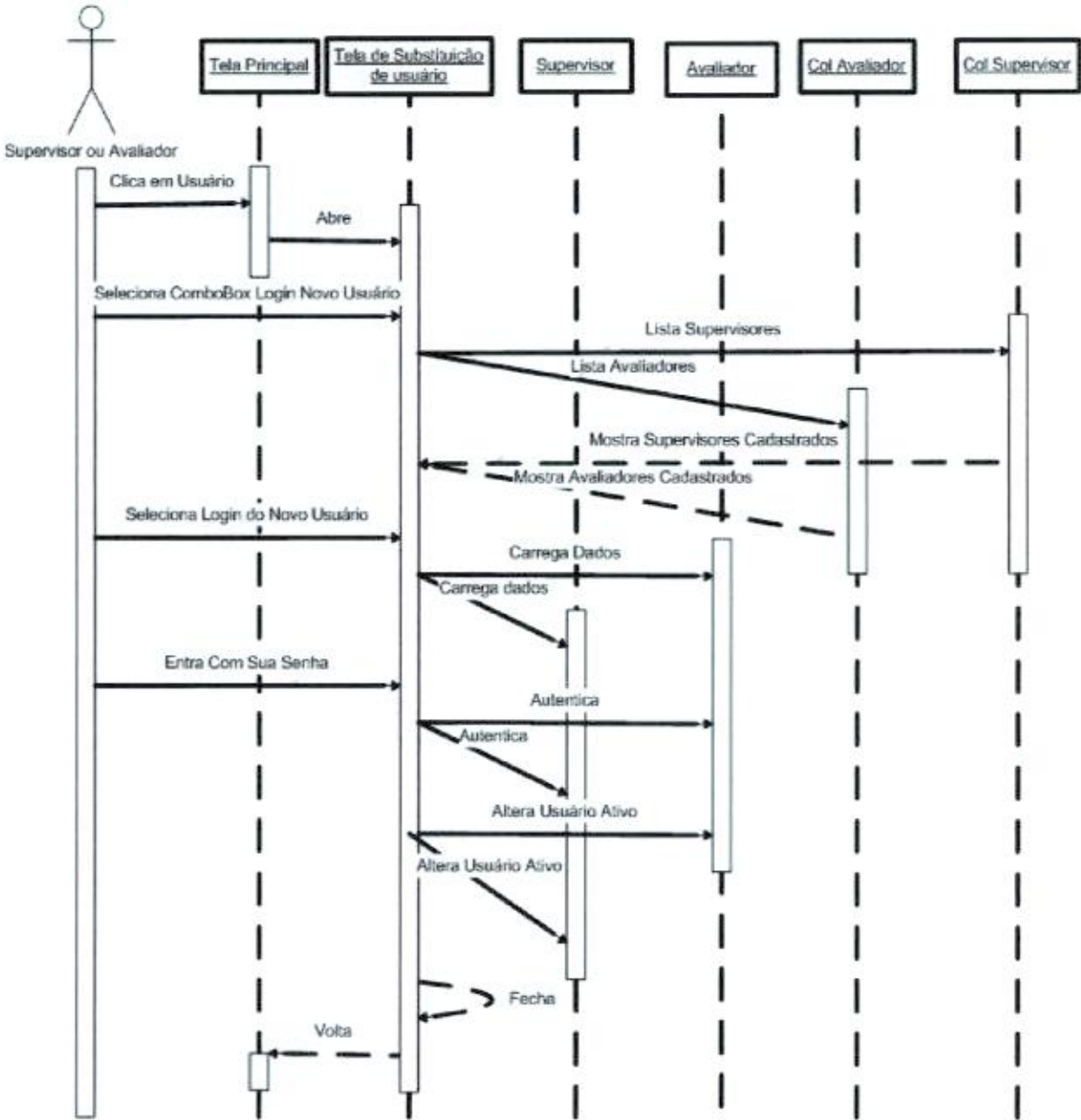


DIAGRAMA DE SEQÜÊNCIA - PROTÓTIPO - TESTE 2.1

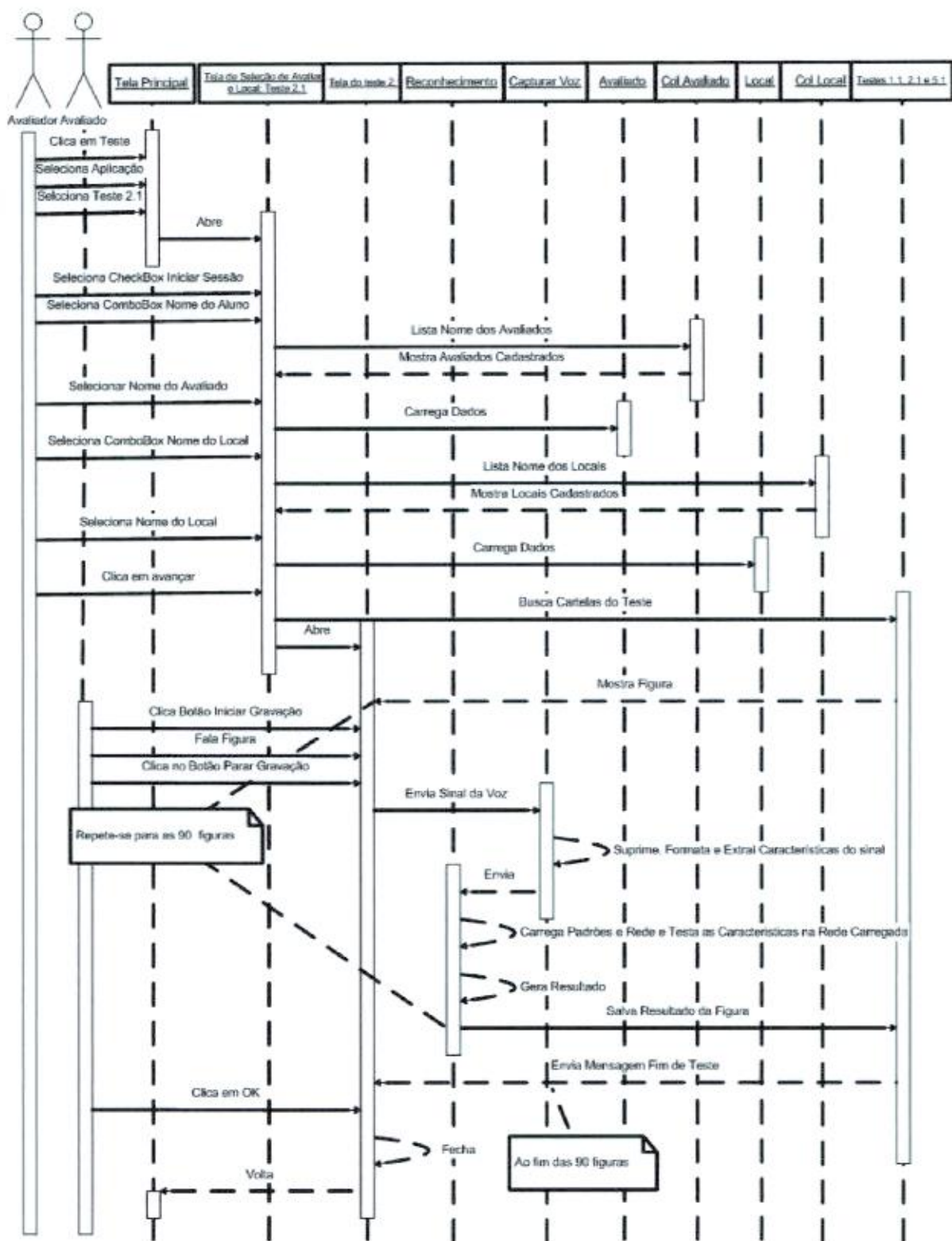


DIAGRAMA DE SEQÜÊNCIA - PROTÓTIPO - TESTE 2.1 - CANCELAR TESTE.

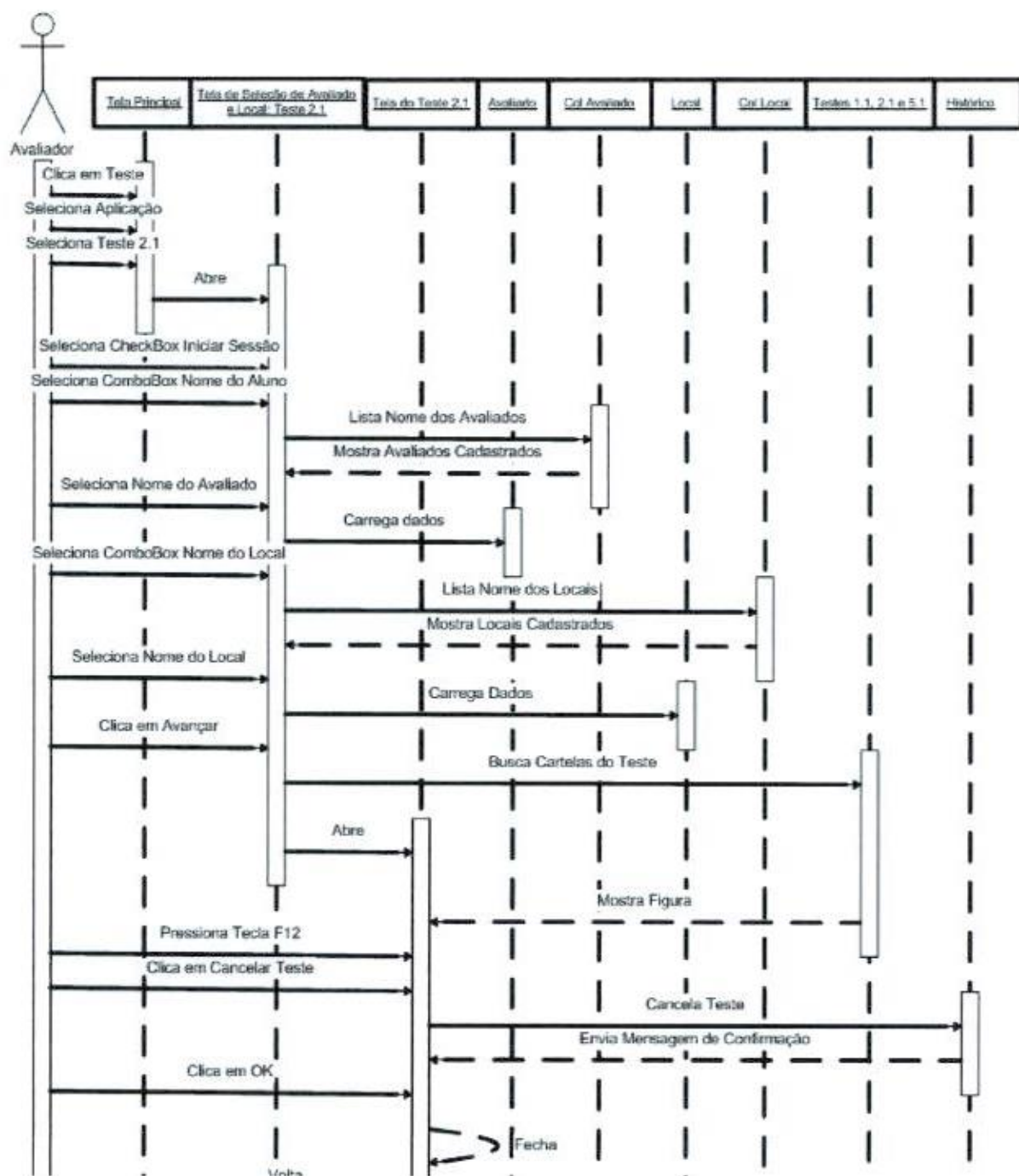


DIAGRAMA DE SEQÜÊNCIA - PROTÓTIPO - TESTE 2.1 - INTERROMPER E APAGAR DADOS DE TODOS OS TESTE REALIZADOS ATÉ O MOMENTO.

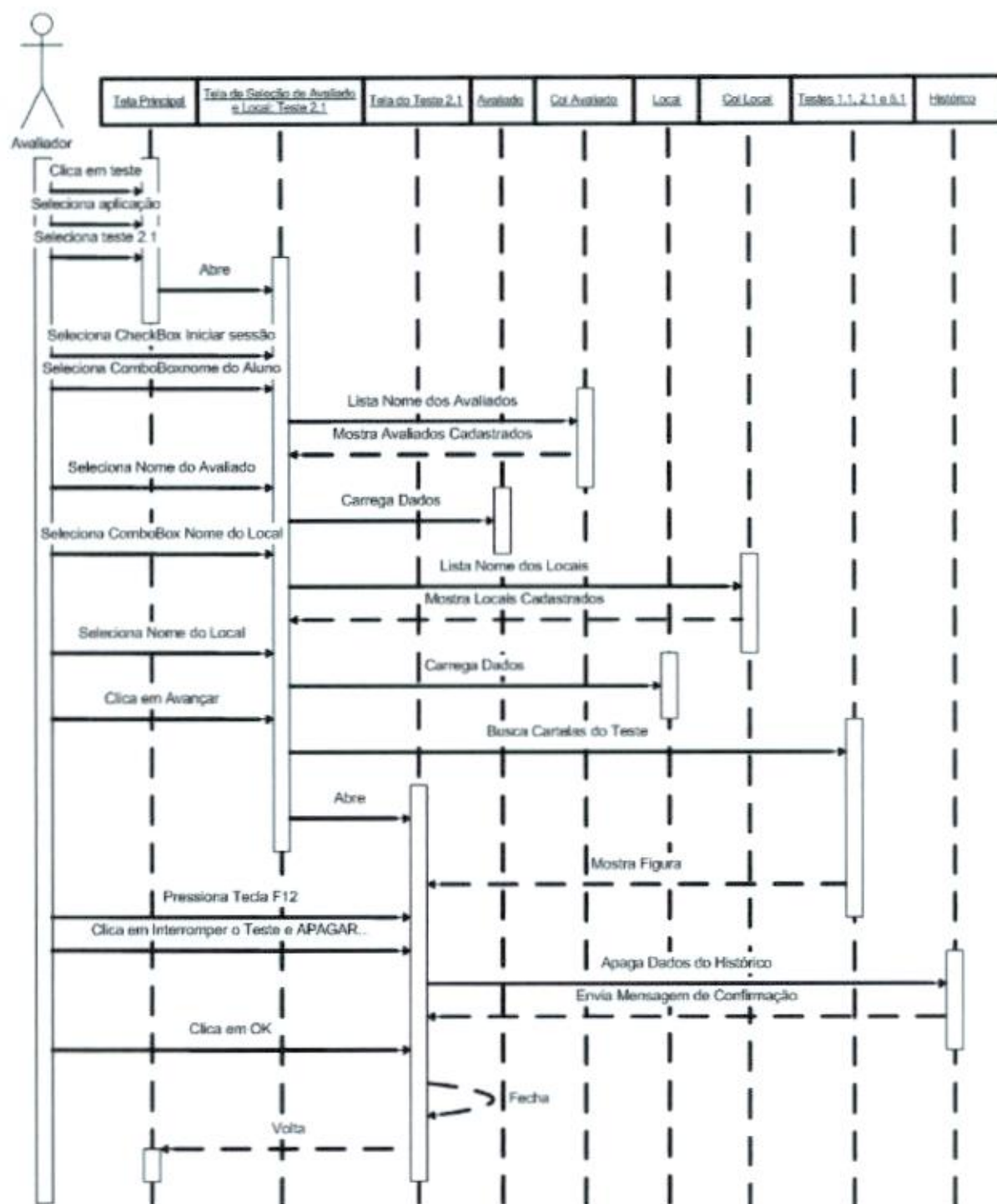


DIAGRAMA DE SEQÜÊNCIA - PROTÓTIPO - TESTE 2.1 - INTERROMPER O TESTE E GRAVAR DADOS PARA RETORNAR FUTURAMENTE.

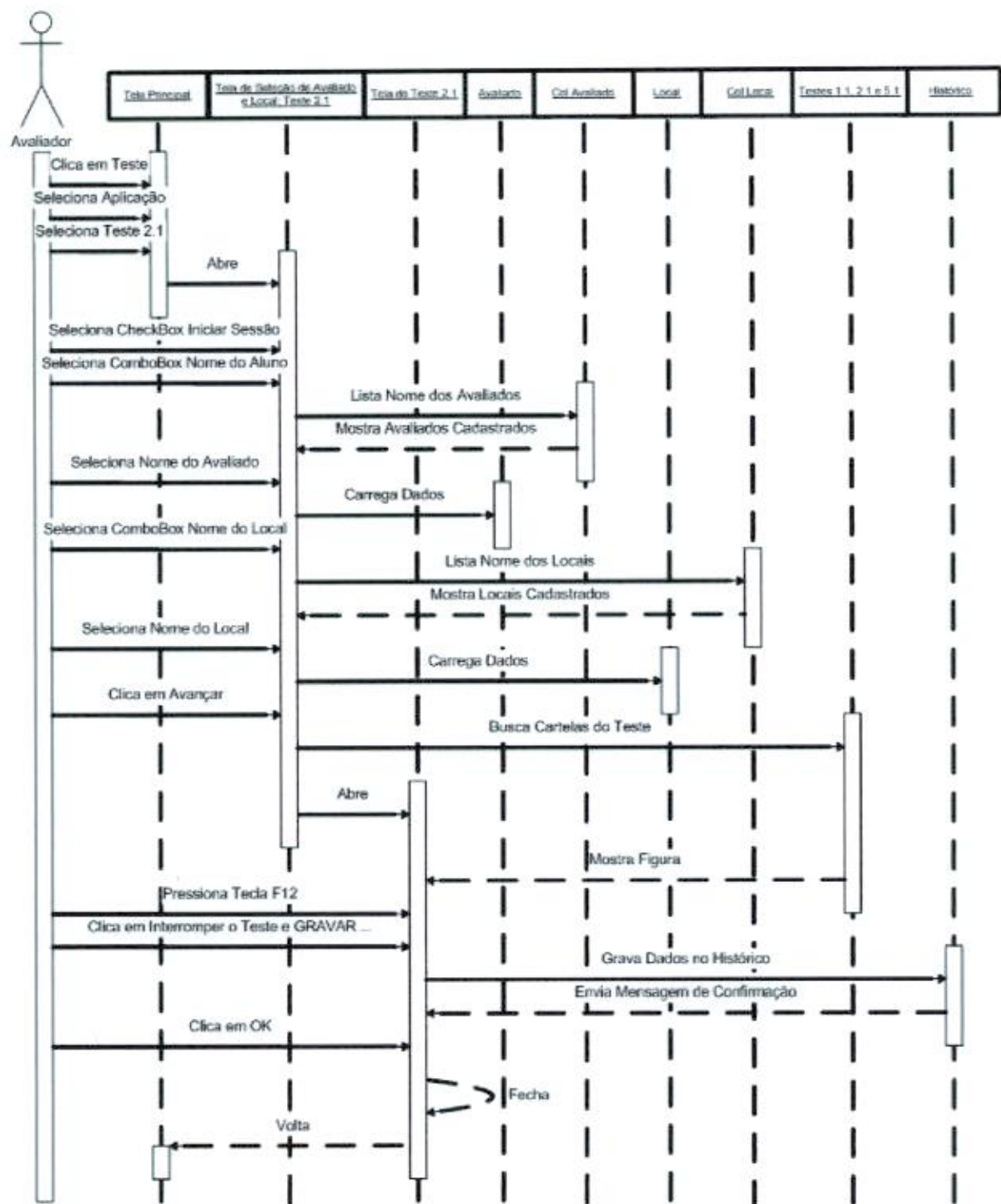


DIAGRAMA DE SEQÜÊNCIA - PROTÓTIPO - TESTE 2.1 - RETOMAR SESSÃO INTERROMPIDA

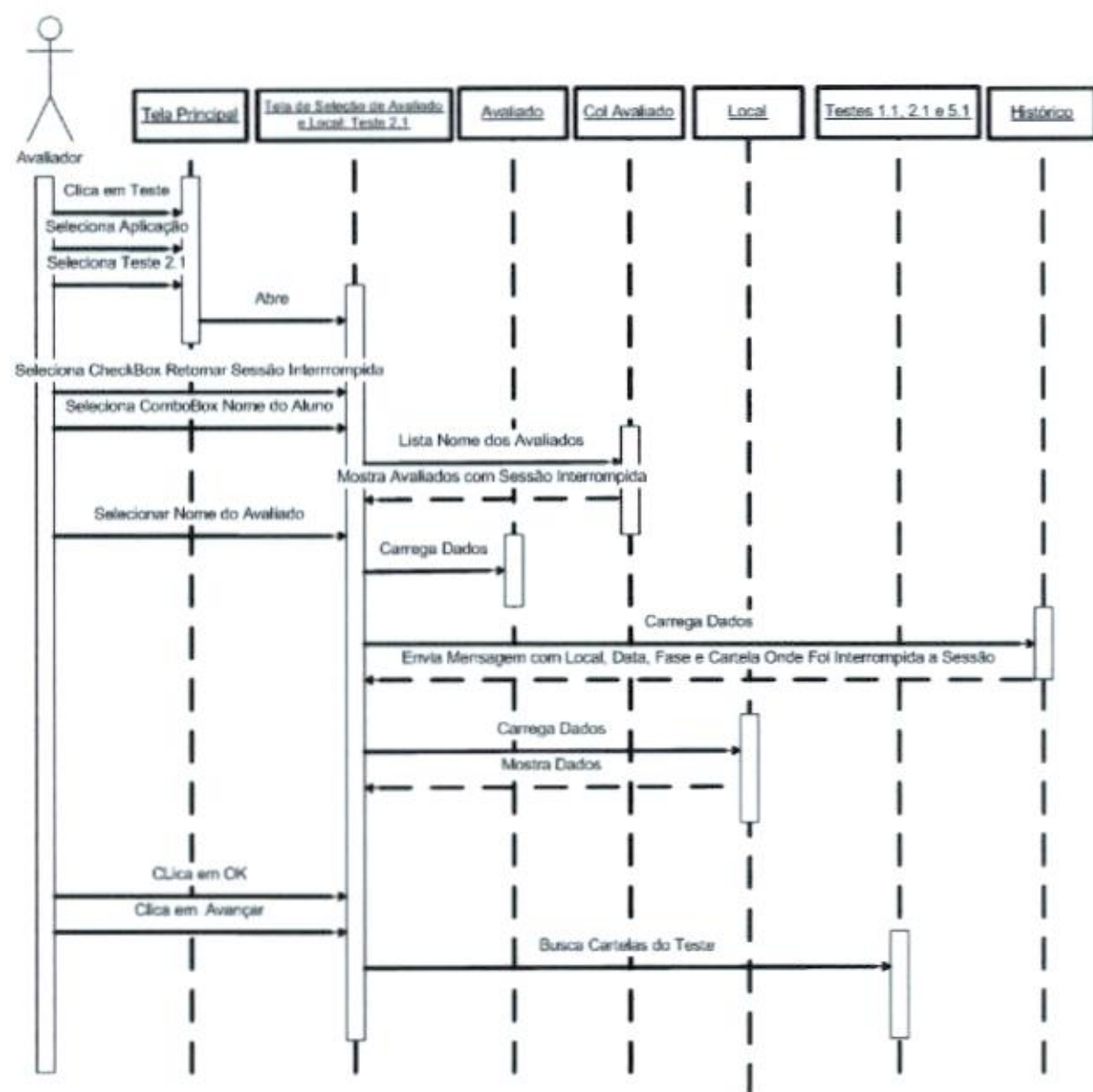


DIAGRAMA DE COLABORAÇÃO - PROTÓTIPO - CADASTRAR AVALIADO.



DIAGRAMA DE COLABORAÇÃO - PROTÓTIPO - ALTERAR DADOS DO AVALIADO.

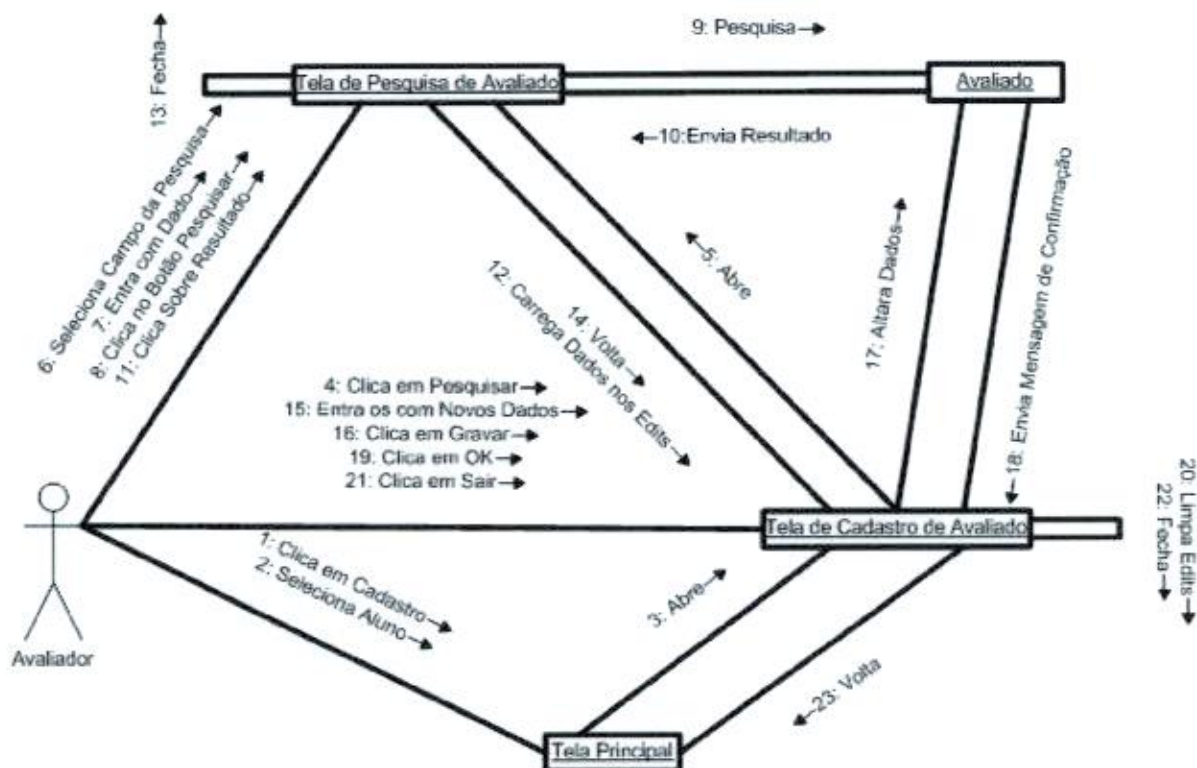


DIAGRAMA DE COLABORAÇÃO - PROTÓTIPO - EXCLUIR AVALIADO.

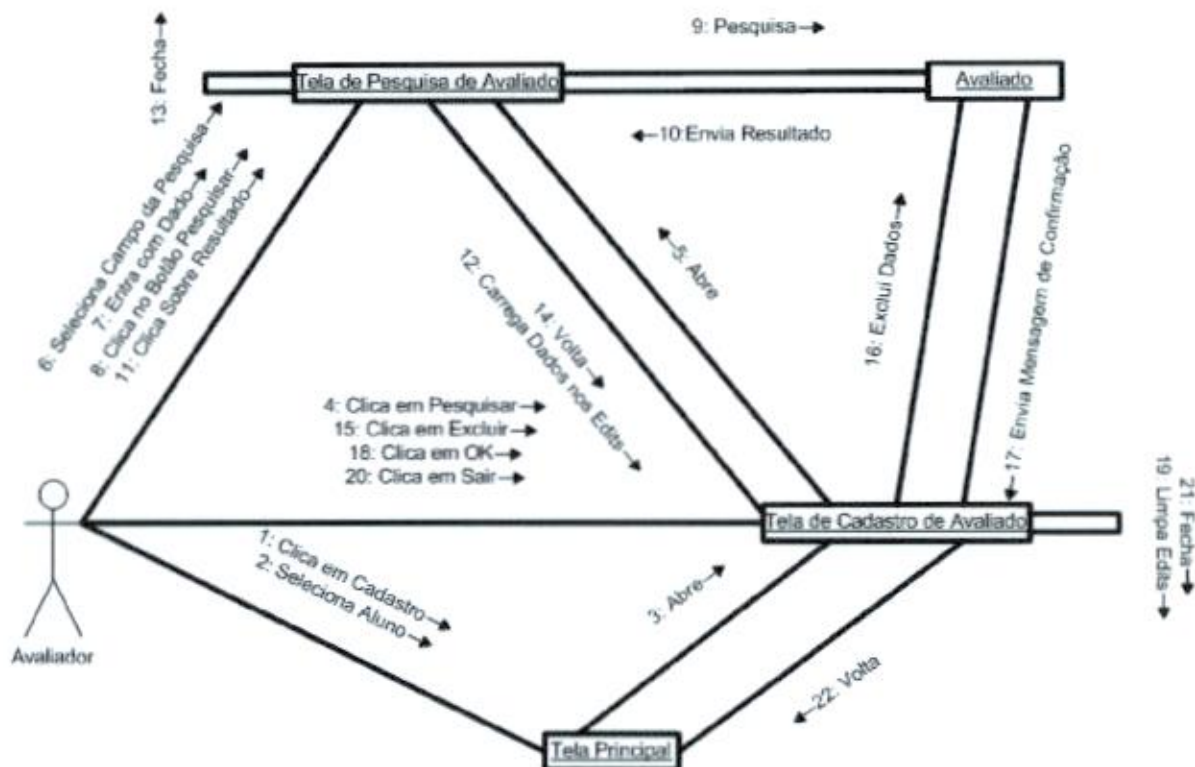


DIAGRAMA DE COLABORAÇÃO - PROTÓTIPO - CADASTRAR AVALIADOR.

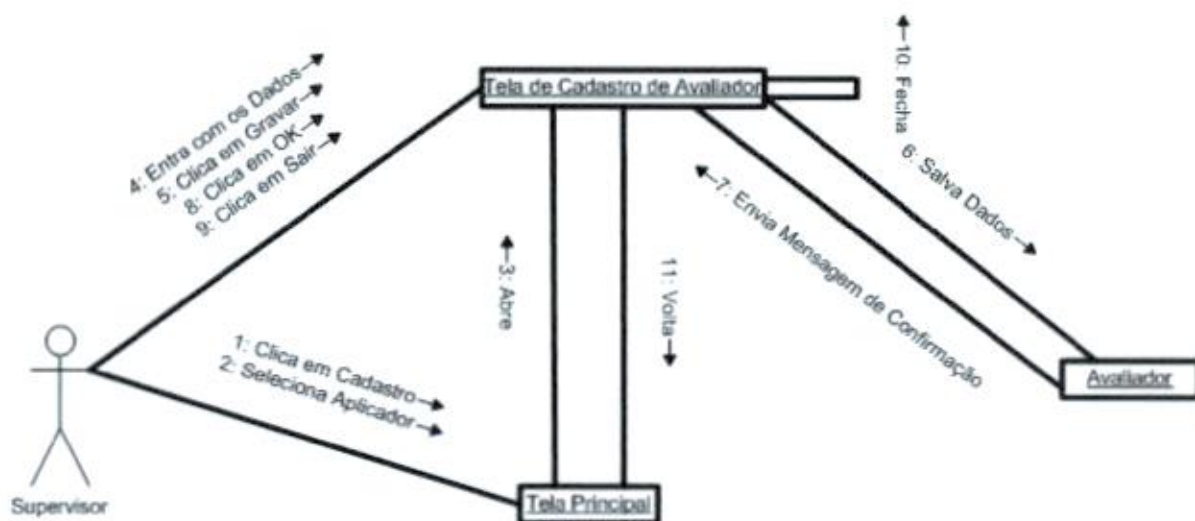


DIAGRAMA DE COLABORAÇÃO - PROTÓTIPO - ALTERAR DADOS DO AVALIADOR.

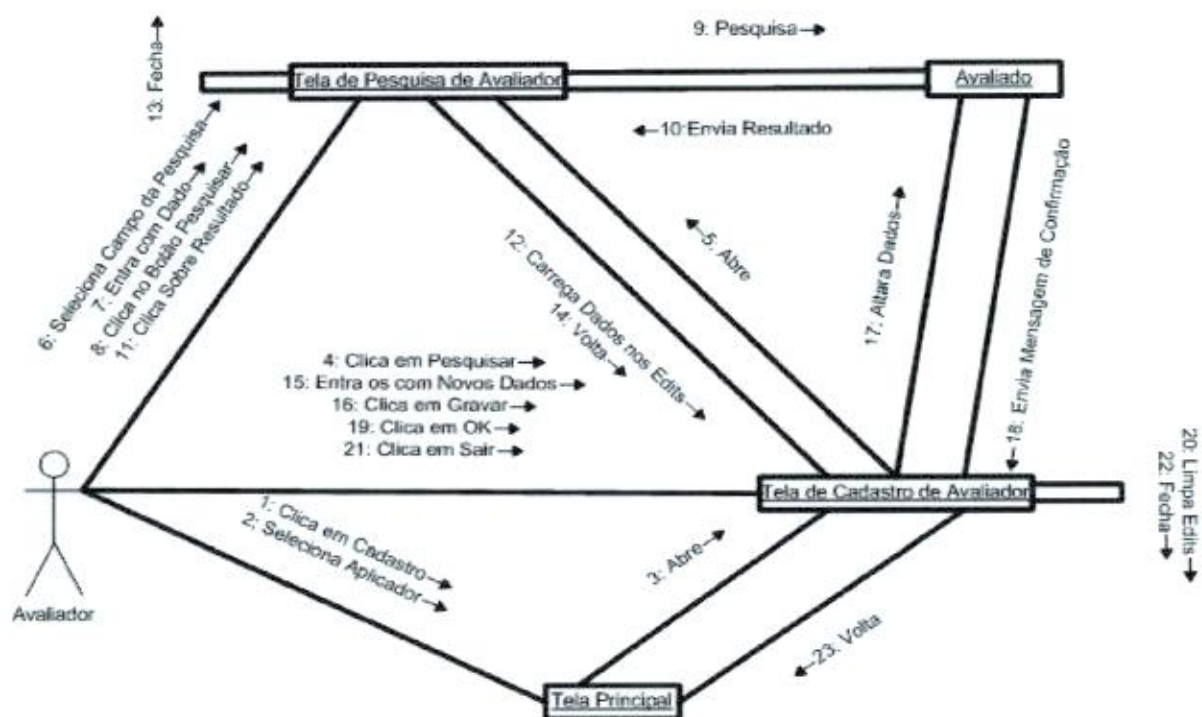


DIAGRAMA DE COLABORAÇÃO - PROTÓTIPO - EXCLUIR AVALIADOR.

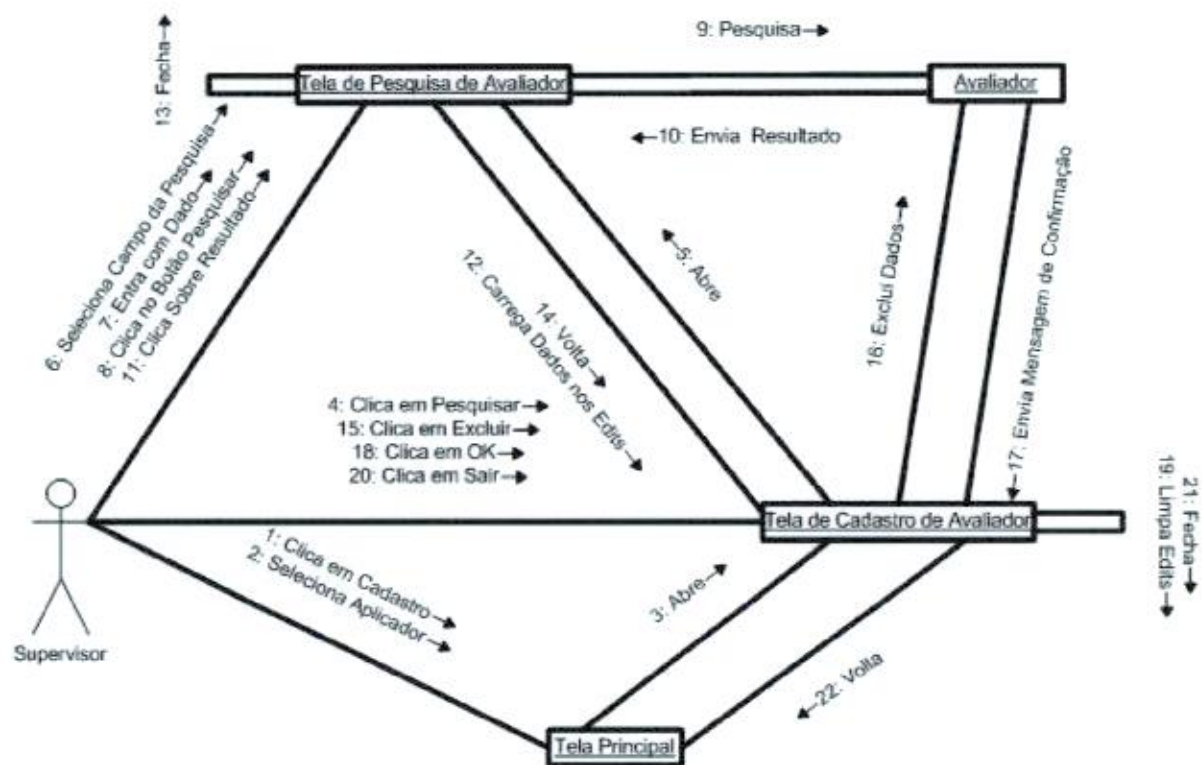


DIAGRAMA DE COLABORAÇÃO - PROTÓTIPO - CADASTRAR LOCAL.

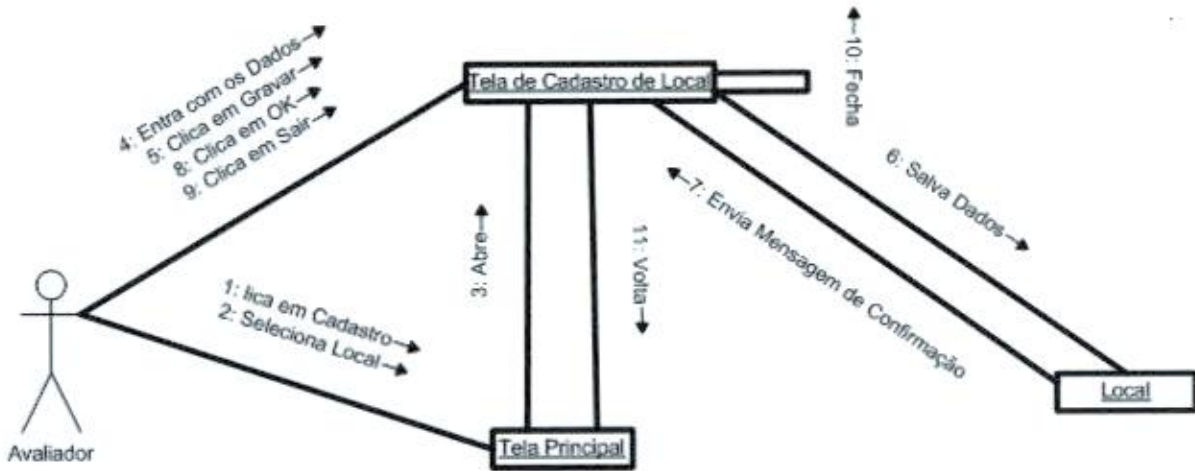


DIAGRAMA DE COLABORAÇÃO - PROTÓTIPO - ALTERAR DADOS DO LOCAL

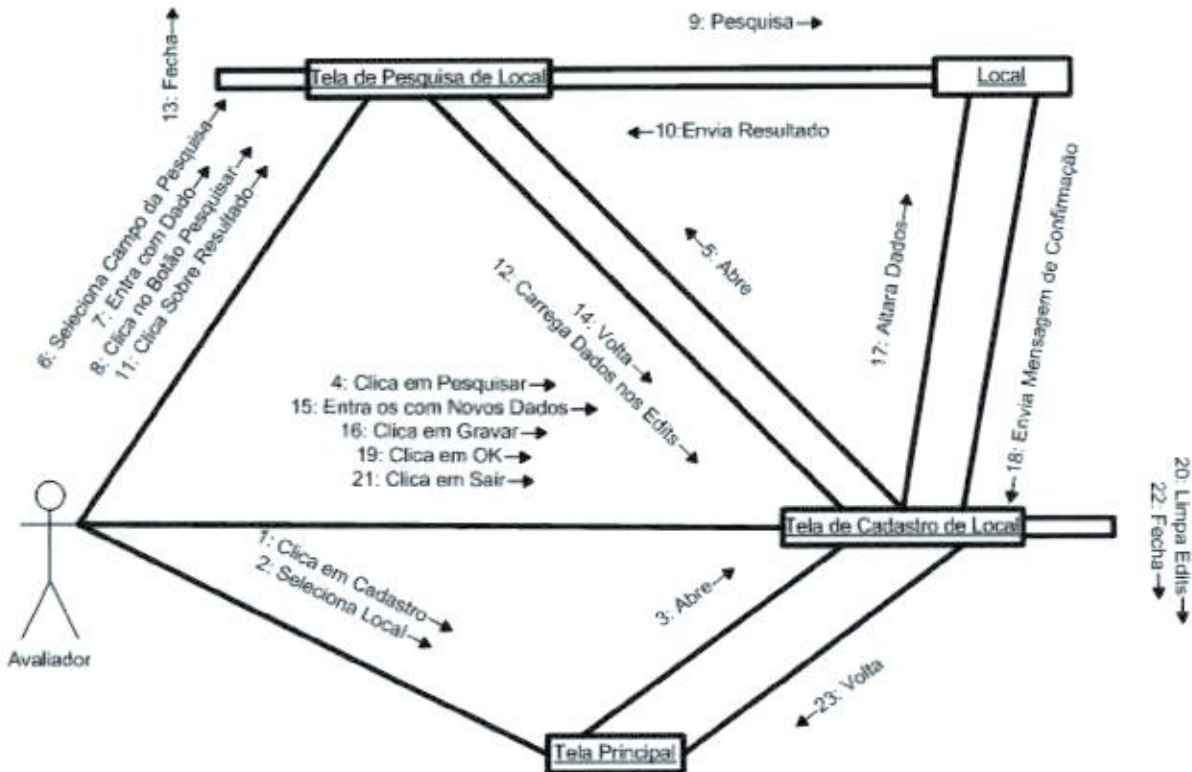


DIAGRAMA DE COLABORAÇÃO - EXCLUIR LOCAL.

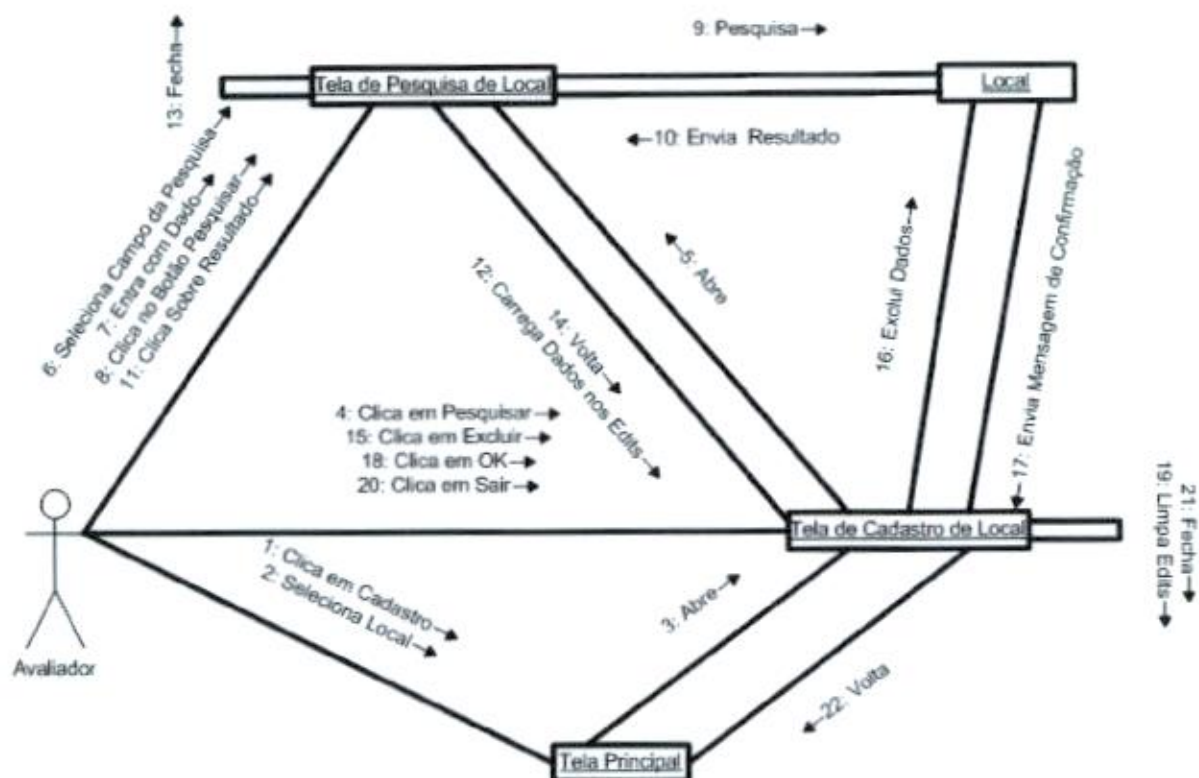


DIAGRAMA DE COLABORAÇÃO - PROTÓTIPO - LOGAR NO SISTEMA

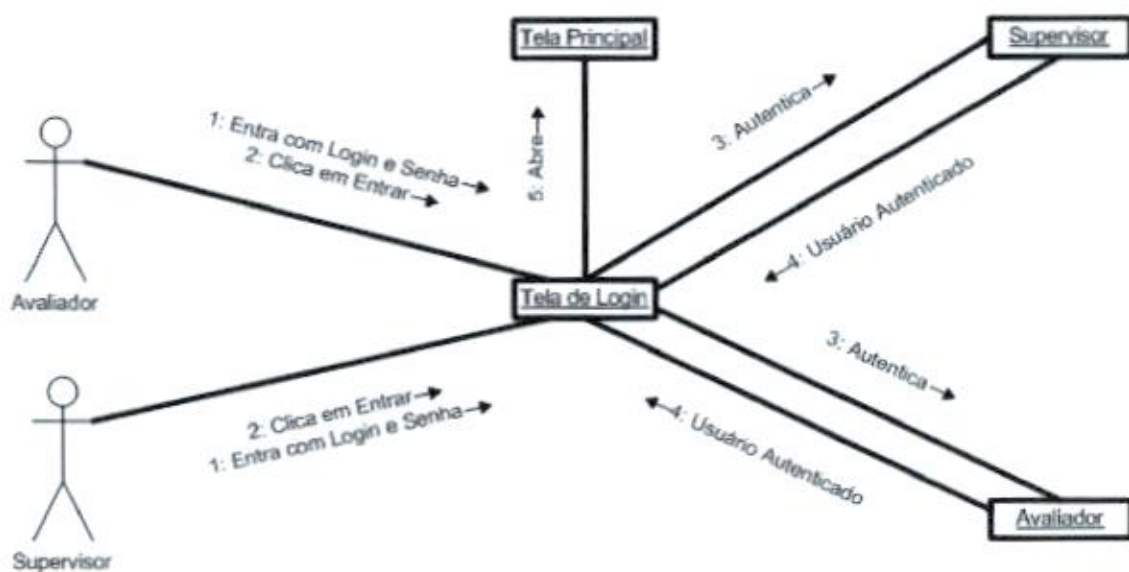


DIAGRAMA DE COLABORAÇÃO - PROTÓTIPO - SUBSTITUIÇÃO DE USUÁRIO.

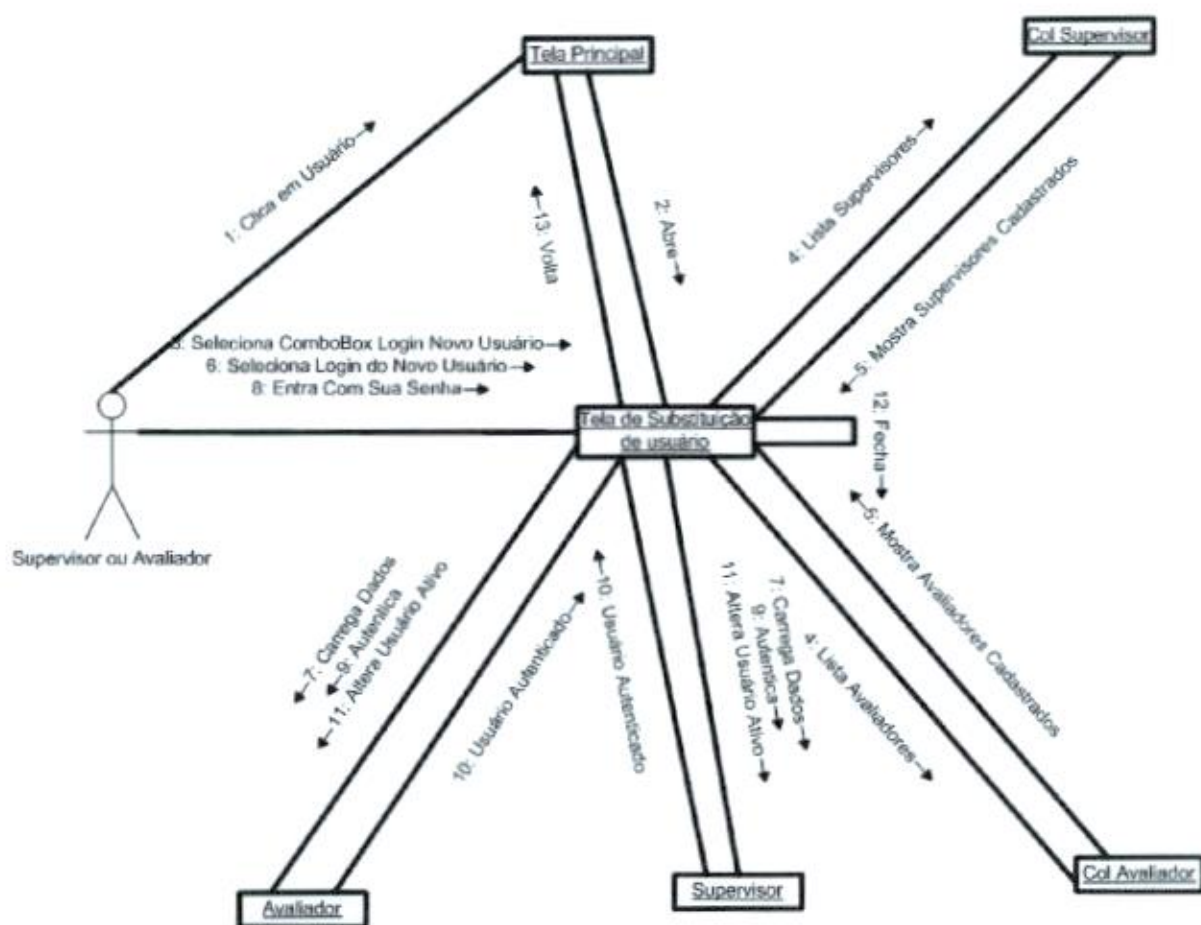


DIAGRAMA DE COLABORAÇÃO - PROTÓTIPO - TESTE 2.1

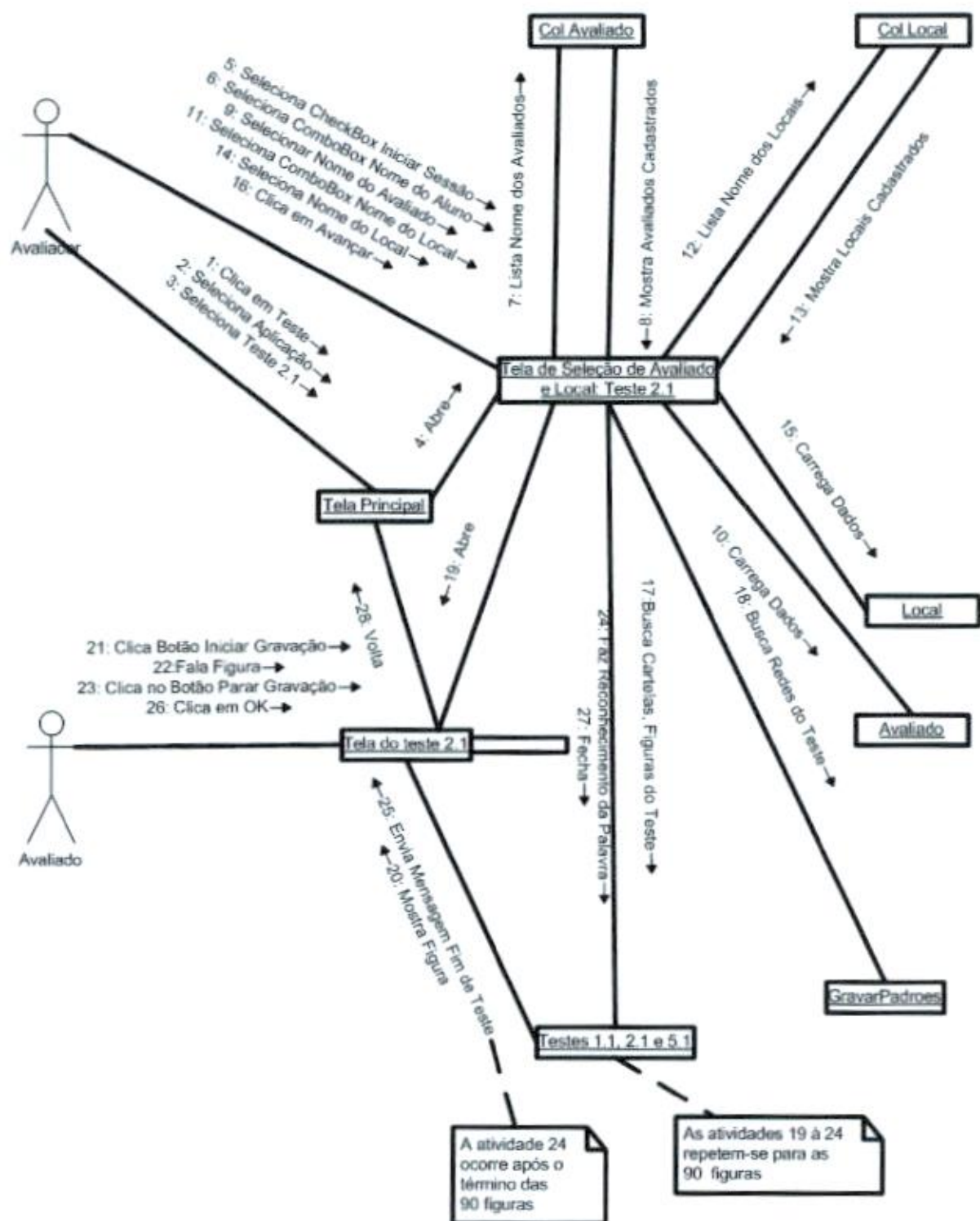


DIAGRAMA DE COLABORAÇÃO - PROTÓTIPO - TESTE 2.1 - CANCELAR TESTE.

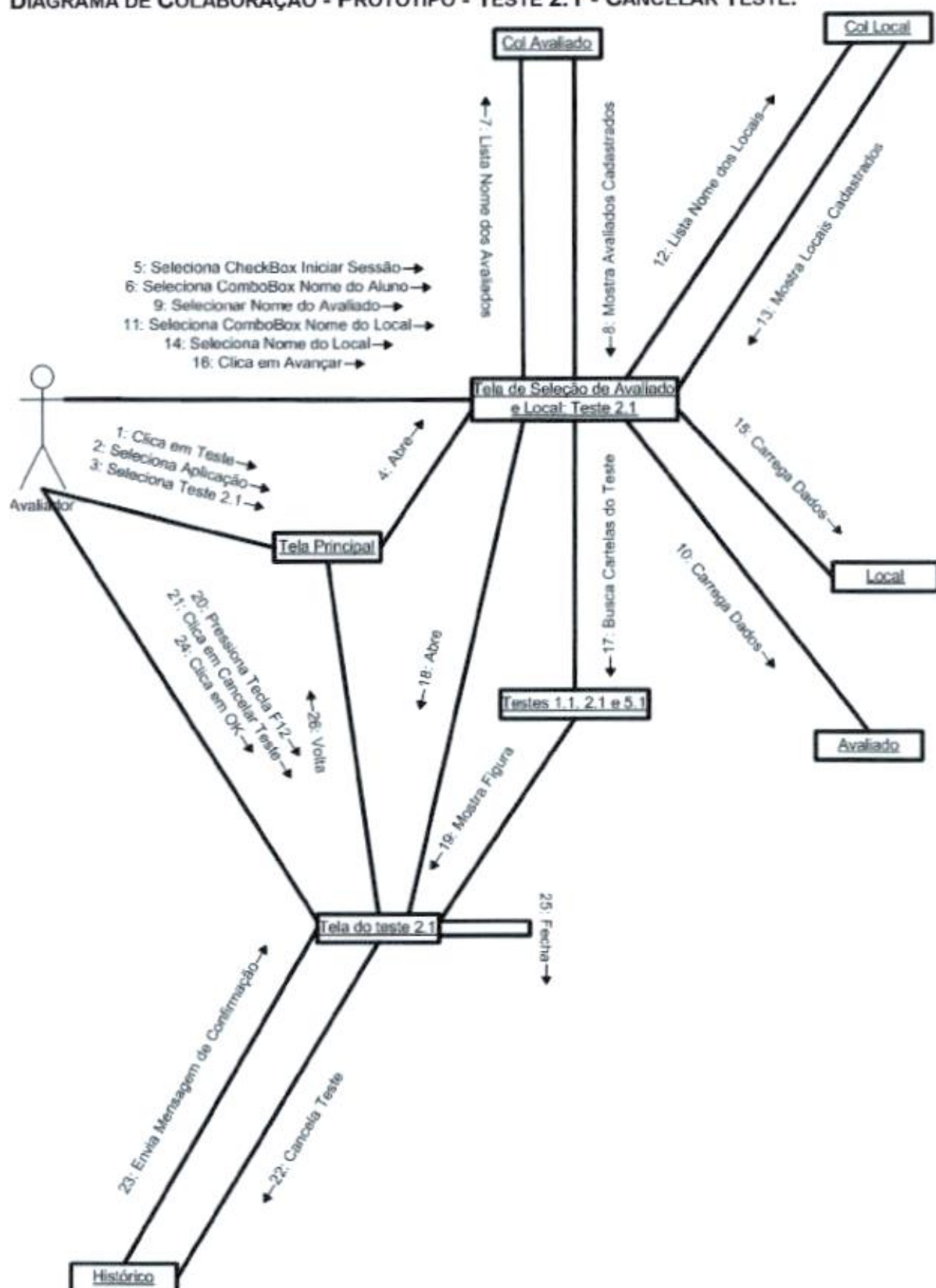


DIAGRAMA DE COLABORAÇÃO - PROTÓTIPO - TESTE 2.1 - INTERROMPER E APAGAR DADOS DE TODOS OS TESTE REALIZADOS ATÉ O MOMENTO.

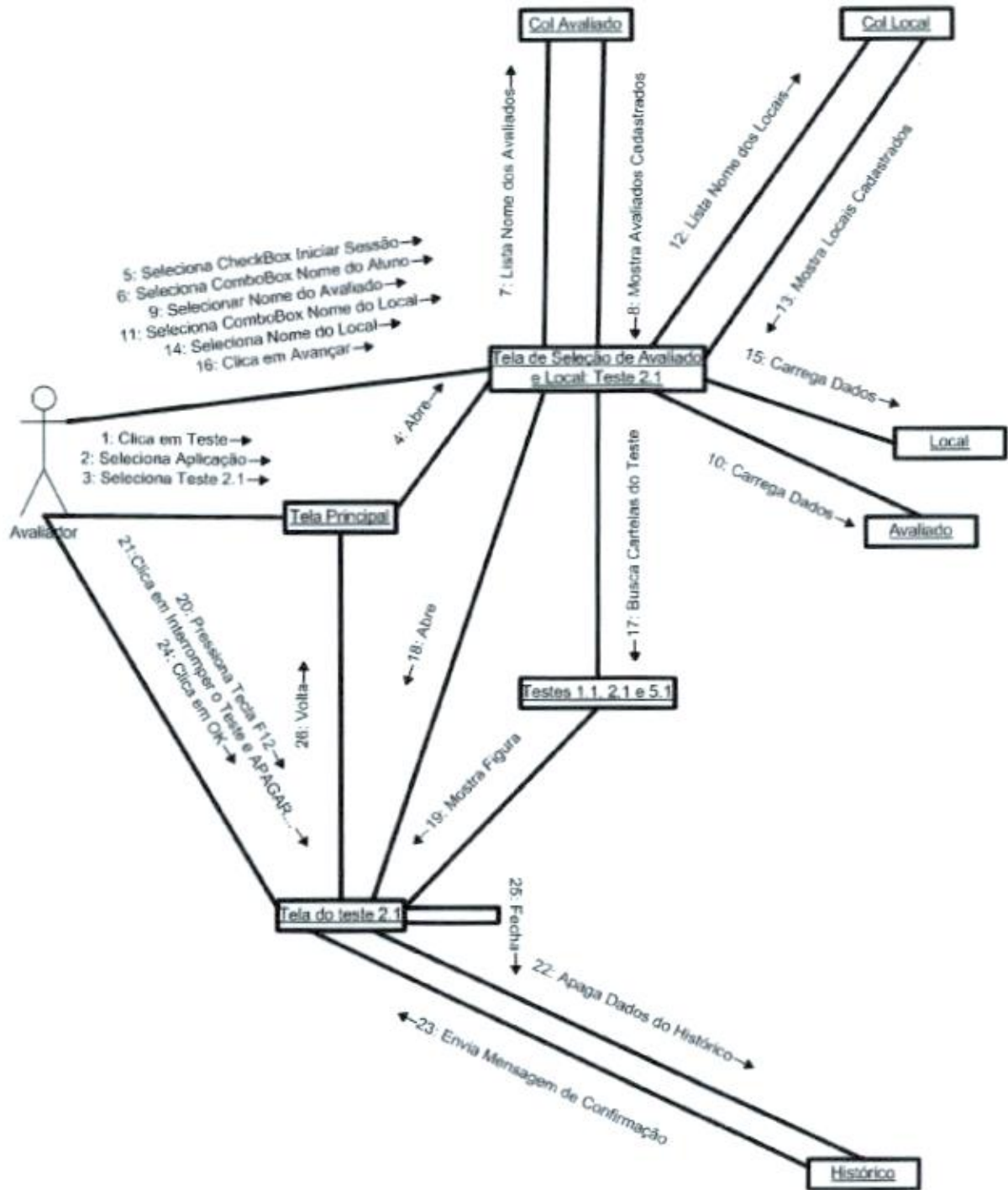


DIAGRAMA DE COLABORAÇÃO - PROTÓTIPO - TESTE 2.1 - INTERROMPER O TESTE E GRAVAR DADOS PARA RETORNAR FUTURAMENTE.

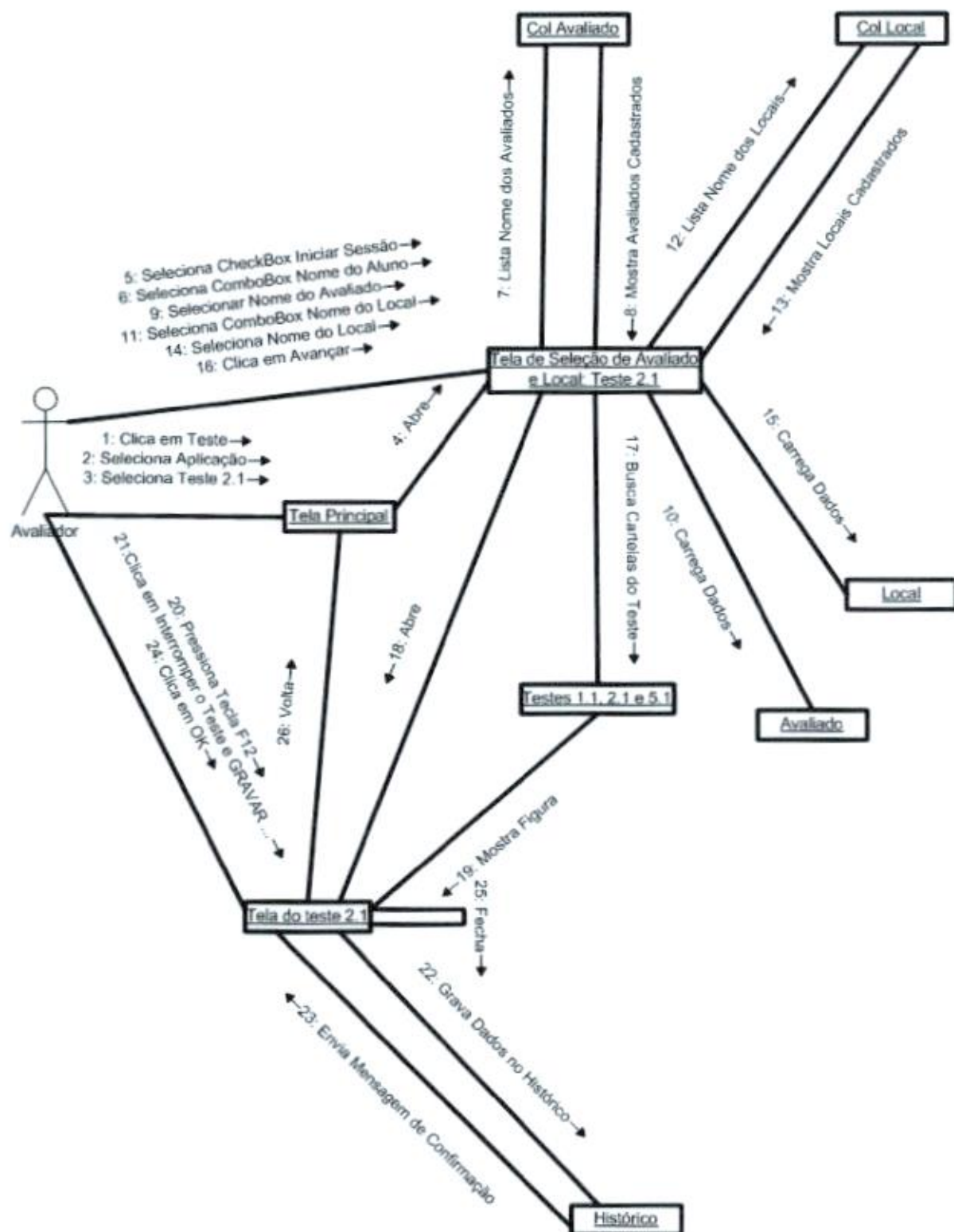
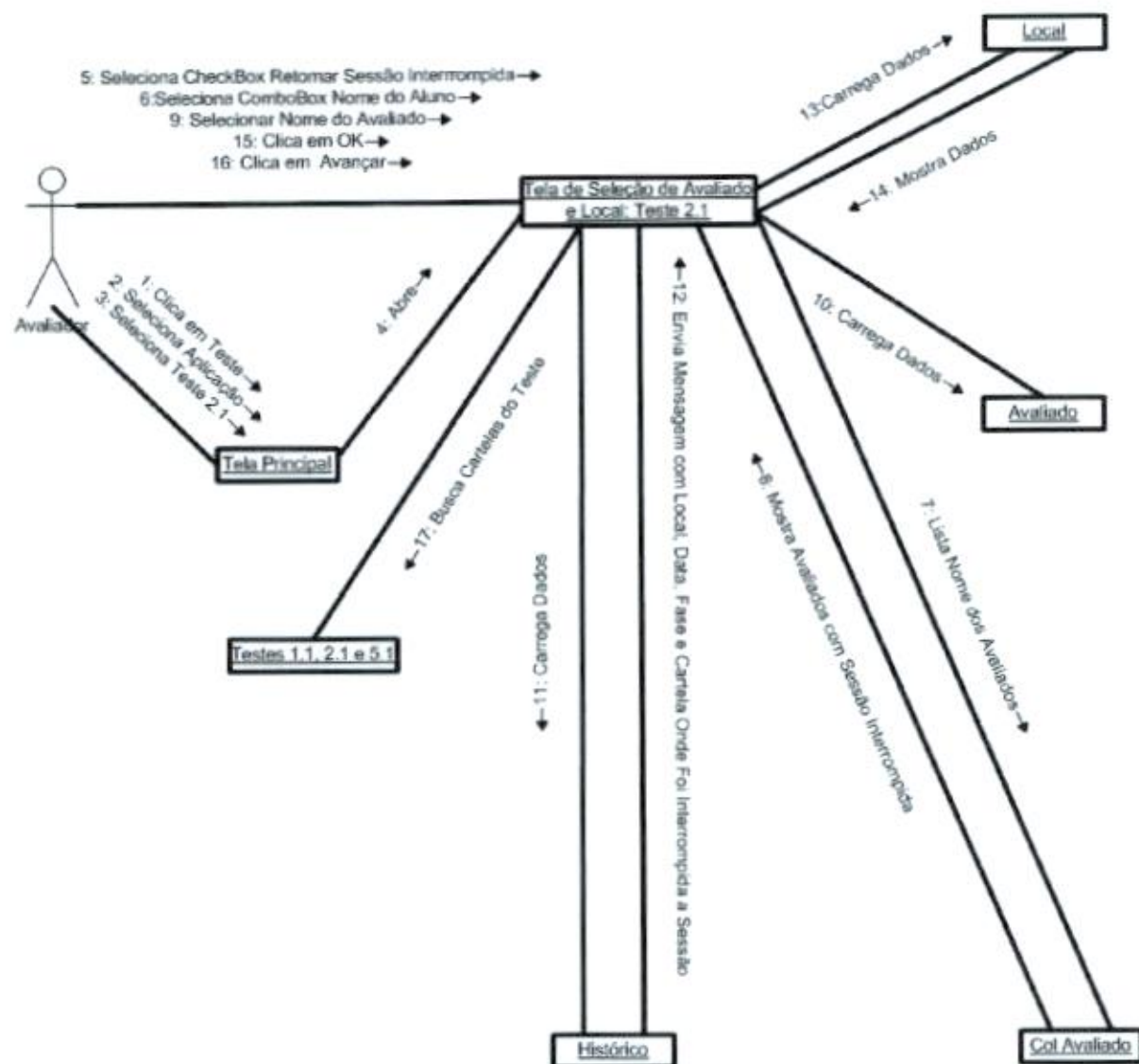


DIAGRAMA DE COLABORAÇÃO - PROTÓTIPO - TESTE 2.1 - RETOMAR SESSÃO INTERROMPIDA



DIAGRAMAS DE ESTADOS - PROTÓTIPO - CADASTROS DE AVALIADOR, AVALIADO E LOCAL.

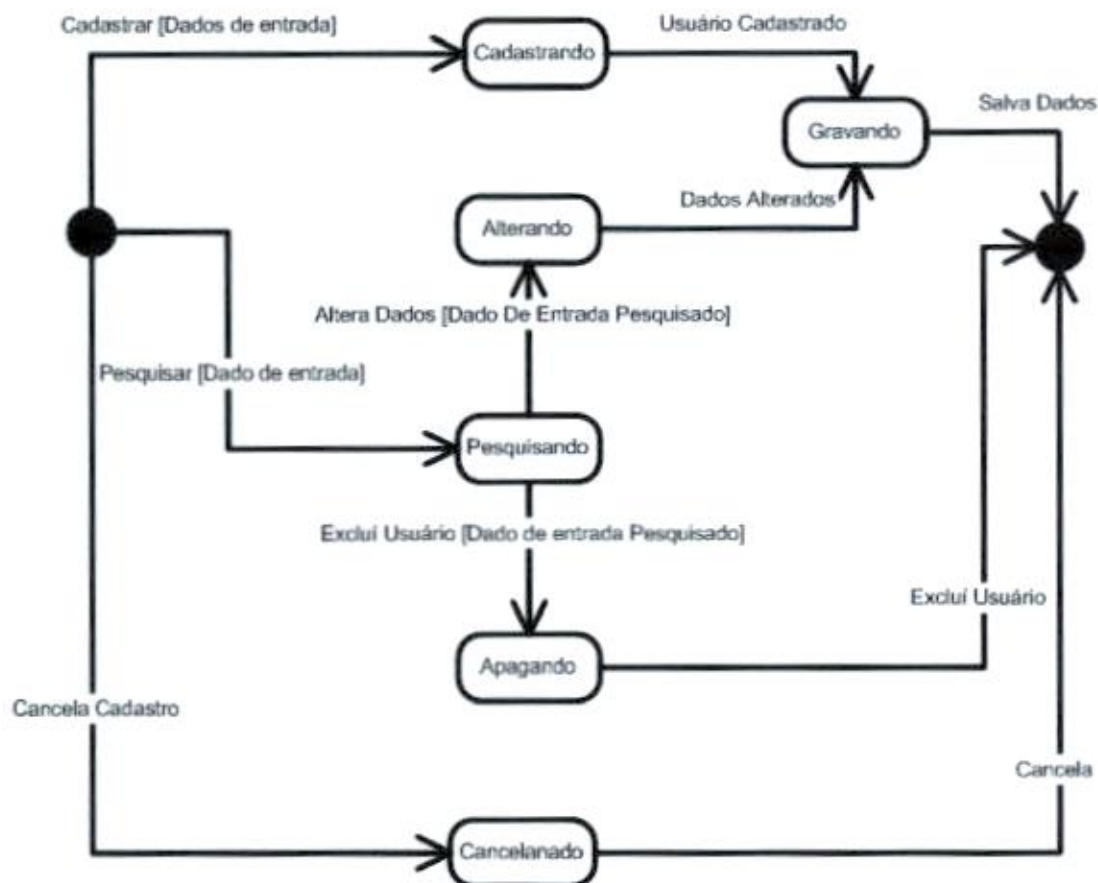
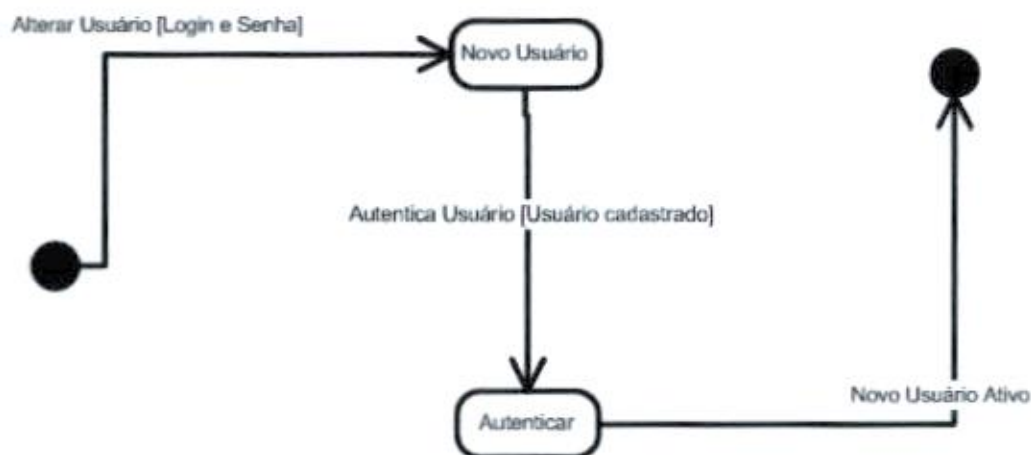
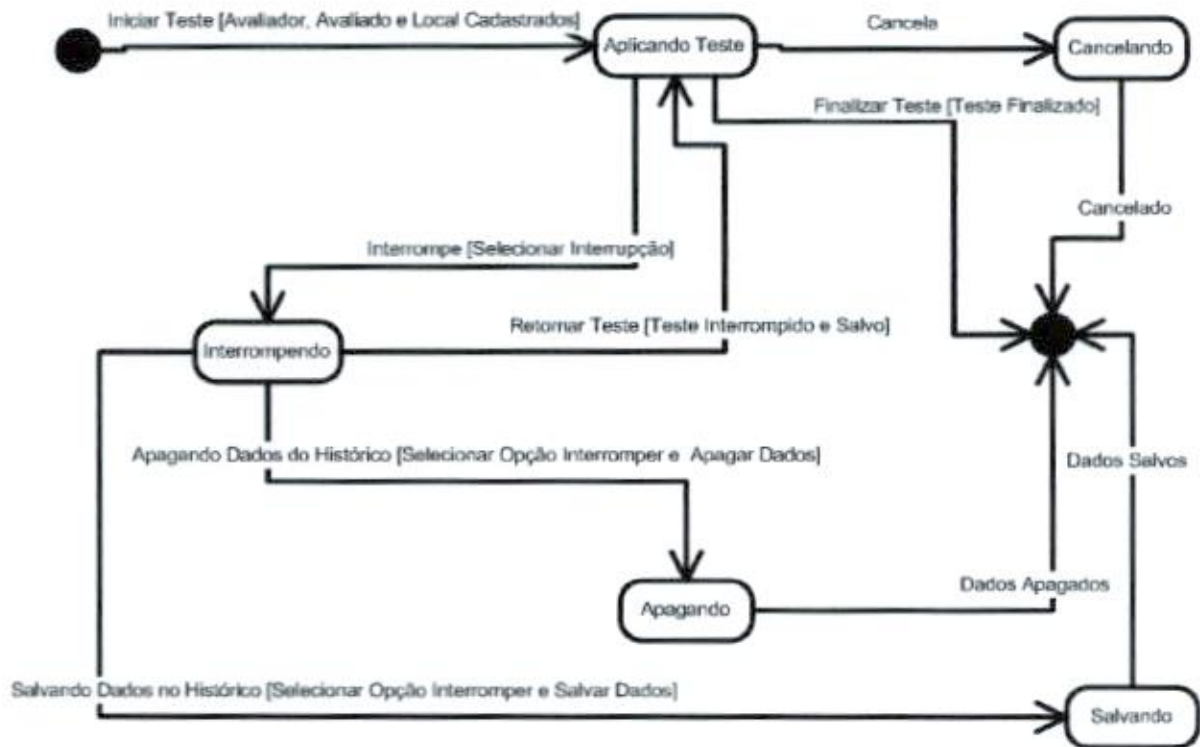


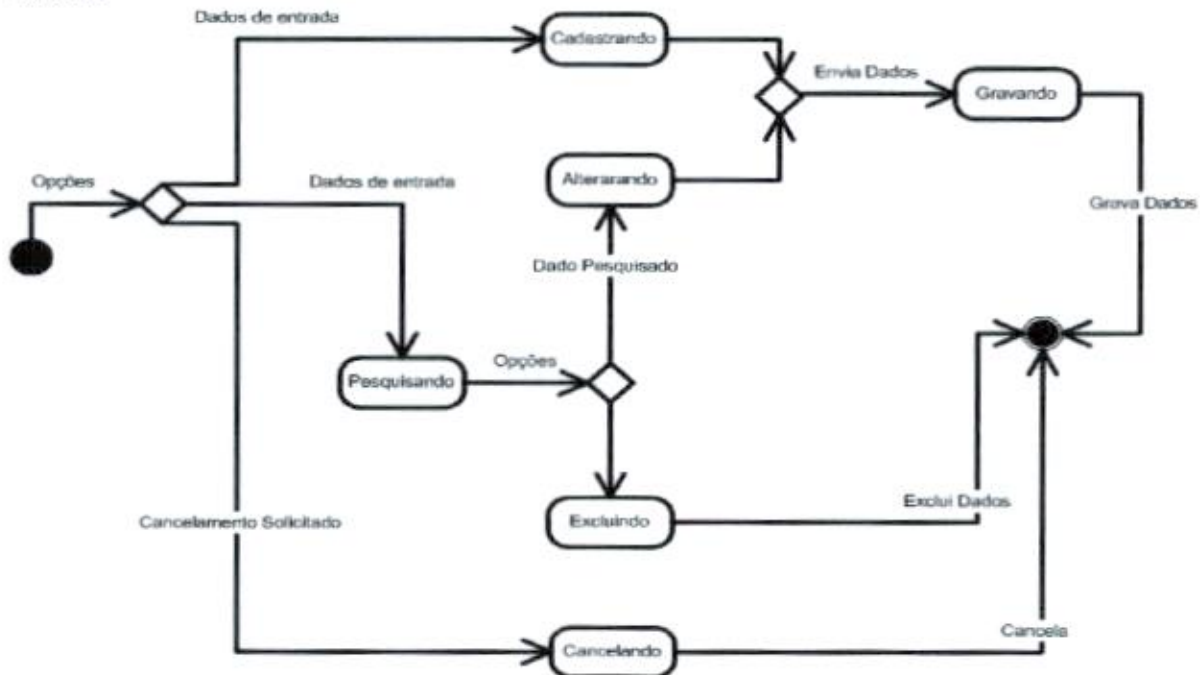
DIAGRAMA DE ESTADOS - SUBSTITUIÇÃO DE USUÁRIO.



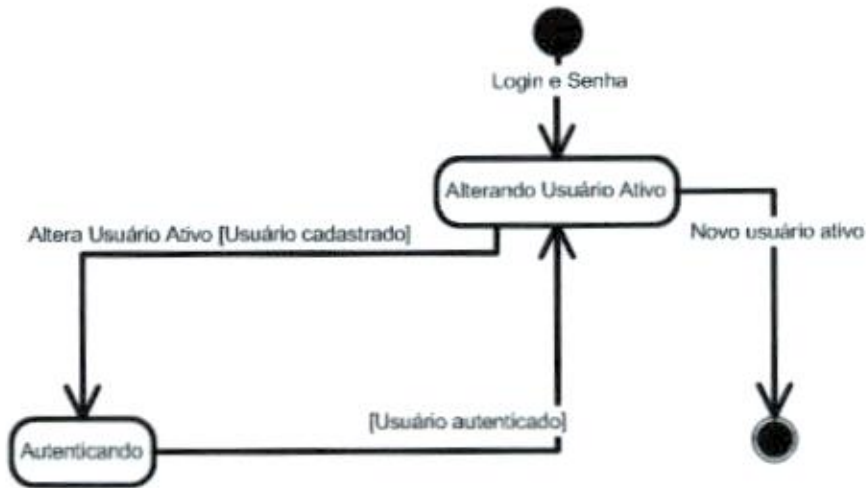
Diagramas de Estados - Protótipo - Teste 2.1



DIAGRAMAS DE ATIVIDADES - PROTÓTIPO - CADASTROS DE AVALIADOR, AVALIADO E LOCAL.



DIAGRAMAS DE ATIVIDADES - PROTÓTIPO - SUBSTITUIÇÃO DE USUÁRIO.



DIAGRAMAS DE ATIVIDADES - PROTÓTIPO - TESTE 2.1

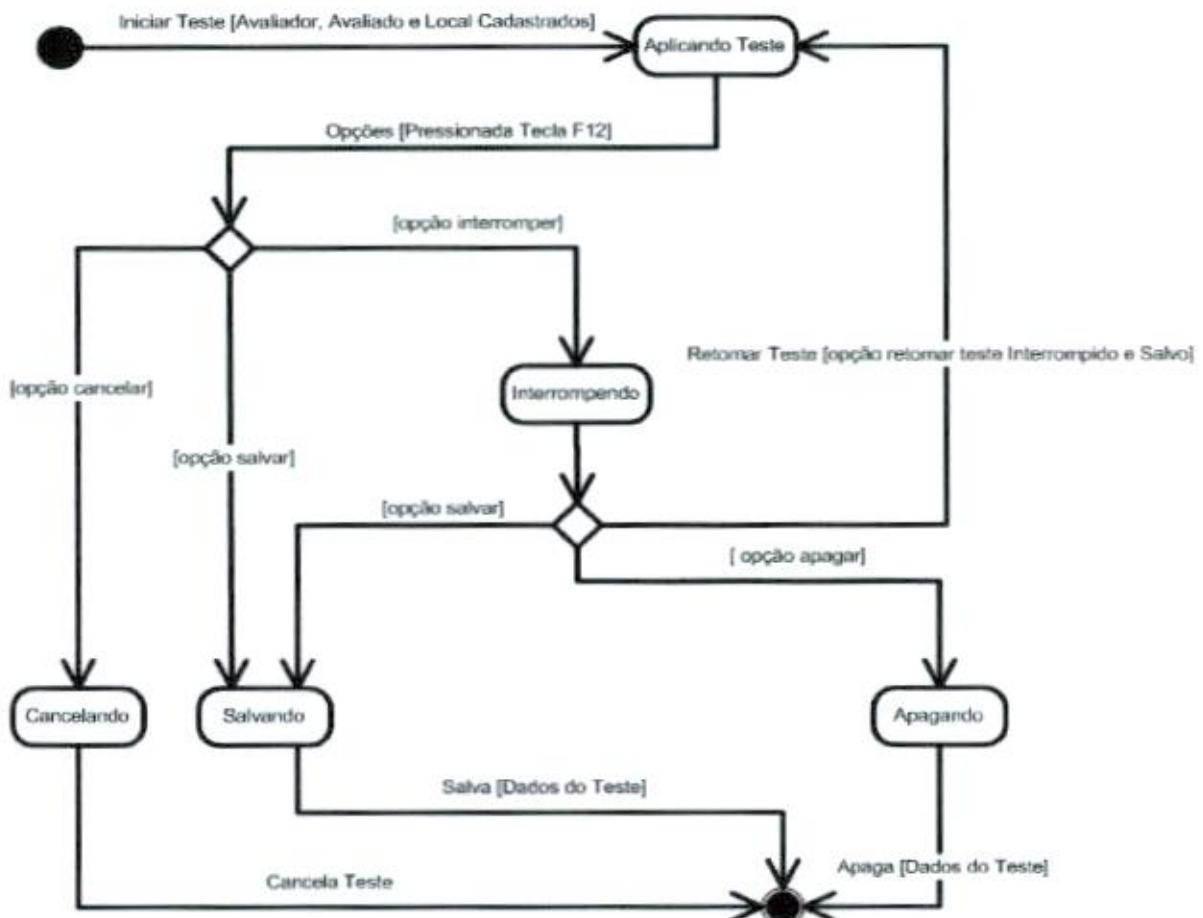


DIAGRAMA ENTIDADE RELACIONAMENTO

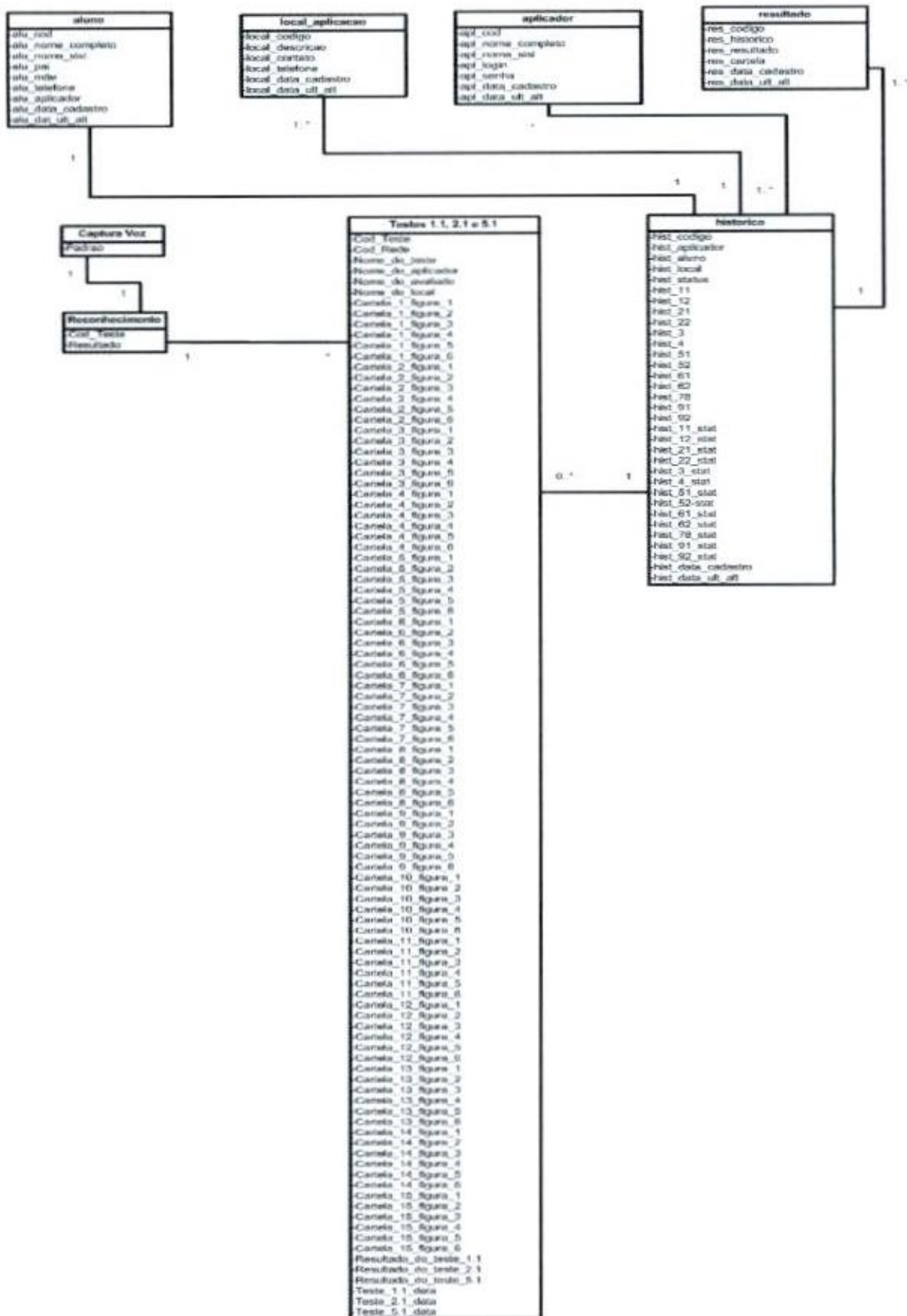
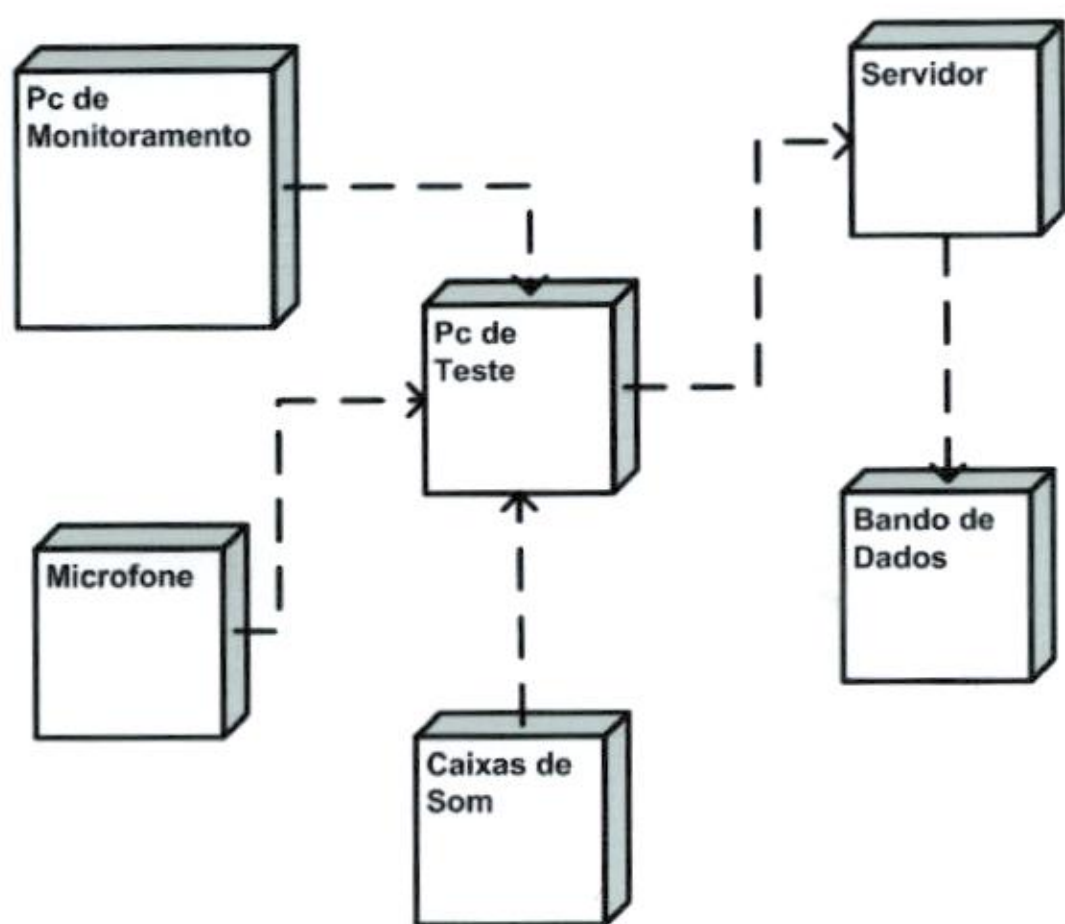


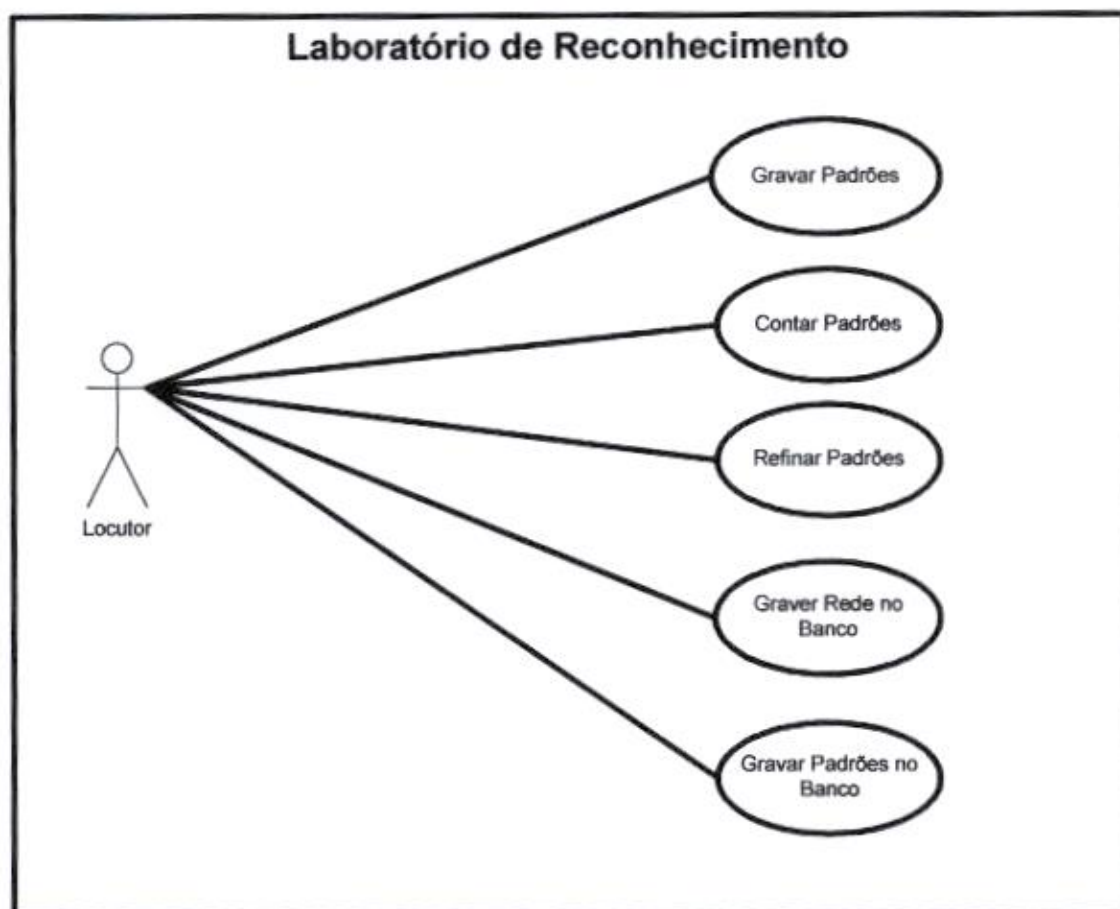
DIAGRAMA DE IMPLANTAÇÃO – PROTÓTIPO



15.2 ANEXO II

MODELAGEM DO LABORATÓRIO DE RECONHECIMENTO

DIAGRAMA DE CASOS DE USO – LABORATÓRIO



DESCRIÇÃO RESUMIDA LABORATÓRIO

CASOS DE USO	QUEM INICIA	DESCRIÇÃO
Gravar Padrões	Locutor	Neste caso de uso o Locutor grava os padrões (palavras).
Contar Padrões	Locutor	Neste caso de uso o Locutor conta o número de padrões gravados .
Refinar Padrões	Locutor	Neste caso de uso o Locutor realiza um refinamento nos padrões, a fim de eliminar padrões ditos "ruins".
Gravar Rede no Banco	Locutor	Neste caso de uso o Locutor grava a rede de uma determinada cartela no banco de dados.
Gravar Padrões no Banco	Locutor	Neste caso de uso o Locutor grava o padrão de uma determinada palavra no banco de dados.

FLUXOS DE EVENTOS LABORATÓRIO

CASO DE USO: GRAVAR PADRÕES.

PRÉ-CONDIÇÕES: Microfone e placa de som instalados e devidamente configurados.

FLUXO PRINCIPAL.

Este caso de uso se inicia quando o locutor seleciona na tela principal o "ComboBox" cartela, seleciona uma das quinze cartelas e clica no botão capturar, que abre a Tela de Captura de Padrões onde o locutor clica no botão gravar e fala

uma das palavras da cartela selecionada. Após isto, ele tem as seguintes alternativas:

PALAVRA RECONHECIDA CERTA, (A-1): executa o sub-fluxo "Palavra reconhecida".

PALAVRA RECONHECIDA ERRADA (A-2): executa o sub-fluxo "Palavra não reconhecida".

DESFAZER (A-3): executa o sub-fluxo "Desfazer confirmação".

FECHAR (A-4): executa o sub-fluxo "Fechando a tela de captura de padrões".

CASO DE USO: CONTAR PADRÕES.

Pré-Condições: Arquivo com os padrões existente.

FLUXO PRINCIPAL.

Este caso de uso se inicia quando o locutor seleciona na tela principal, o botão contador, que abrirá tela de Contagem de Padrões, aonde ele terá as seguintes alternativas:

ABRIR (A-5): executa o sub-fluxo "Abrir um arquivo para contagem".

SALVAR (A-6): executa o sub-fluxo "Salvar aleatoriamente".

FECHAR (A-7): executa o sub-fluxo "Fechando a tela de contagem".

CASO DE USO: REFINAR PADRÕES.

Pré-Condições: Arquivo com os padrões existente.

FLUXO PRINCIPAL.

Este caso de uso se inicia quando o locutor seleciona na tela principal, o botão refinador, que abrirá uma tela aonde ele terá as seguintes alternativas: ABRIR um arquivo para refinamento ou FECHAR a tela. Caso opção escolhida seja:

ABRIR (A-8): executa o sub-fluxo "Abrir um arquivo para refinamento".

SALVAR (A-9): executa o sub-fluxo "Salvar refinamento".

FECHAR (A-10): executa o sub-fluxo "Fechando a tela de refinamento".

CASO DE USO: GRAVAR REDE NO BANCO.

Pré-Condições: Computador devidamente configurado para acessar ao banco de dados remoto.

FLUXO PRINCIPAL.

Este caso de uso se inicia quando o locutor seleciona na tela principal, o botão gravar rede no banco, onde abre-se uma tela onde o locutor seleciona no "ComboBox" qual das 15 redes que será salva e clica no botão executar, (A-11) que executa o sub-fluxo "Gravando rede no banco".

CASO DE USO: GRAVAR PADRÕES NO BANCO.

Pré-Condições: Computador devidamente configurado para acessar ao banco de dados remoto.

FLUXO PRINCIPAL.

Este caso de uso se inicia quando o locutor seleciona na tela principal, o botão gravar padrões no banco, e abre-se tela de Gravação da Rede no Banco, onde o locutor seleciona o menu iniciar, o diretório e o nome do arquivo, feito isto, ele terá as seguintes alternativas:

EXECUTAR (A-12): executa o sub-fluxo "Gravando padrões".

FECHAR (A-13): executa o sub-fluxo "Fechando a tela de gravar no banco".

FLUXOS ALTERNATIVOS.

A-1: Palavra reconhecida.

O locutor clica no botão certo e o laboratório grava as características da palavra.

A-2: Palavra reconhecida.

O locutor clica no botão errado, abre-se uma tela aonde (E-1), e o laboratório grava as características da palavra.

A-3: Desfazer confirmação

O locutor clica no botão desfazer Certo/Errado e o laboratório apaga o ultimo padrão salvo no arquivo.

A-4: Fechando a tela de Captura de Padrões.

Após abrir a tela de Captura de Padrões, o usuário poderá a qualquer momento encerrar o caso de uso clicando no botão FECHAR.

A-5: Abrir um arquivo para contagem.

O locutor seleciona o diretório e o arquivo para contagem e a aplicação mostra o número total de padrões(palavras) salvos e ainda o número de vezes que cada palavra de uma determinada cartela aparece nesse arquivo.

A-6: Salvar aleatoriamente.

O locutor seleciona o menu salvar, o diretório e o nome do arquivo e salva o arquivo de forma aleatória, afim de evitar muito padrões da mesma palavra em seqüência.

A-7: Fechando a tela de contagem.

Após abrir a tela de contagem de padrões, o usuário poderá a qualquer momento encerrar o caso de uso selecionando no menu a opção FECHAR.

A-8: Abrir um arquivo para refinamento.

O locutor seleciona o diretório e o arquivo para refinamento e a aplicação mostra o arquivo normal no "memo" chamado arquivo original, e o arquivo refinado no segundo "memo" chamado arquivo refinado.

A-9: Salvar refinamento,

O locutor seleciona o menu salvar, o diretório e o nome do arquivo e salva o arquivo do refinamento.

A-10: Fechando a Tela de Refinamento.

Após abrir a tela de refinamento de padrões, o usuário poderá a qualquer momento encerrar o caso de uso selecionando no menu a opção FECHAR.

A-11: Abrir um arquivo de padrões para gravar.

Onde o locutor seleciona o diretório e o arquivo com de padrões(palavras) a gravar.

A-12: Gravando padrões.

A rede é salva no banco de dados remoto.

A-13: Fechando a tela de gravar no banco.

Após abrir a tela de gravação de padrões, o usuário poderá a qualquer momento encerrar o caso de uso selecionando no menu a opção FECHAR.

FLUXOS DE EXCEÇÃO.

E-1: O locutor seleciona no "ComboBox" a palavra que falou realmente. Abrir um arquivo de padrões para gravar.

CENÁRIOS LABORATÓRIO

CASO DE USO: GRAVAR PADRÕES.

CENÁRIO PRINCIPAL.

O locutor João da Silva seleciona na tela principal a opção cartela 1 no "ComboBox" cartela, e clica no botão captura que abre a tela de captura de padrões onde João fala no microfone a palavra bode e o laboratório mostra que a palavra com maior grau de pertinência e a palavra bode, após isto, João clica no botão certo e as características da palavra são salvas no arquivo.

CENÁRIOS SECUNDÁRIOS.

O laboratório mostra a palavra bote como a palavra com maior grau de pertinência, então João clica no botão errado, abre-se a tela de correção de padrões, e João seleciona no "ComboBox" a palavra bode e clica no botão OK e o laboratório salva a palavra no arquivo.

O laboratório mostra a palavra bote como a palavra com maior grau de pertinência, então João clica no botão errado, abre-se a tela de correção de padrões, e João seleciona no "ComboBox" a palavra bote e percebe que gravou um padrão errado, então clica no botão Desfazer Certo/Errado e o laboratório desfaz a gravação do padrão.

O locutor João clica no botão fechar que volta para a tela principal do laboratório.

O laboratório informa ocorrência de erro pois o arquivo de padrões não existe, então João cria o arquivo e então executa novamente a gravação.

CASO DE USO: CONTAR PADRÕES.

CENÁRIO PRINCIPAL.

O locutor João da Silva clica na tela principal no botão contador que abre a tela de contagem de padrões aonde o mesmo seleciona o menu abrir e localiza o diretório e o arquivo com os padrões e, após isto, é mostrado na mesma tela o numero total de padrões e o número de vezes que cada palavra rede aparece.

O locutor João seleciona o menu salvar, seleciona o diretório e o nome do arquivo e salva o mesmo com os padrões distribuídos de forma aleatória.

O locutor João seleciona o menu fechar que volta para a tela principal do laboratório.

CASO DE USO: REFINAR PADRÕES.

CENÁRIO PRINCIPAL.

O locutor João da Silva clica na tela principal no botão refinador que abre a tela de refinamento de padrões onde ele clica no menu abrir, seleciona o diretório e o arquivo a serem refinados e, após isto, o laboratório mostra no "memo" Arquivo original, o arquivo original propriamente dito e o número de linhas que o mesmo contém, e nome "memo" arquivo refinado, o arquivo refinado com o seu respectivo número de linhas.

O locutor João seleciona o menu salvar, seleciona o diretório e o nome do arquivo e salva o mesmo.

O locutor João seleciona o menu fechar que volta para a tela principal do laboratório.

CASO DE USO: GRAVAR REDE NO BANCO.

CENÁRIO PRINCIPAL.

O locutor João da Silva clica na tela principal no botão Gravar Rede no Banco que abra a tela de gravação de rede no banco onde ele no "ComboBox" a rede a ser salva e clica no botão executar, que salva a rede no banco de dados remoto.

CENÁRIOS SECUNDÁRIOS.

O laboratório informa ocorrência de erro pois não é possível acessar o banco de dados remoto, então João verifica o que há de errado e após solucionar o problema, executa a tarefa novamente.

CASO DE USO: GRAVAR PADRÕES NO BANCO.

CENÁRIO PRINCIPAL.

O locutor João da Silva clica na tela principal no botão Gravar Padrões no Banco que abra a tela de gravação de padrões no banco onde ele seleciona o menu abrir, seleciona o diretório e o arquivo, que mostra em um "StringGrid" o número total de padrões.

O locutor João seleciona o menu executar, que salva os padrões no banco de dados remoto.

O locutor João seleciona o menu fechar que volta para a tela principal do laboratório.

CENÁRIOS SECUNDÁRIOS.

O laboratório informa ocorrência de erro pois não é possível acessar o banco de dados remoto, então João verifica o que há de errado e após solucionar o problema, executa a tarefa novamente.

DIAGRAMA DE TELAS LABORATÓRIO DE RECONHECIMENTO



DIAGRAMA DE SEQÜÊNCIA - LABORATÓRIO - CAPTURAR - RECONHECIMENTO CORRETO

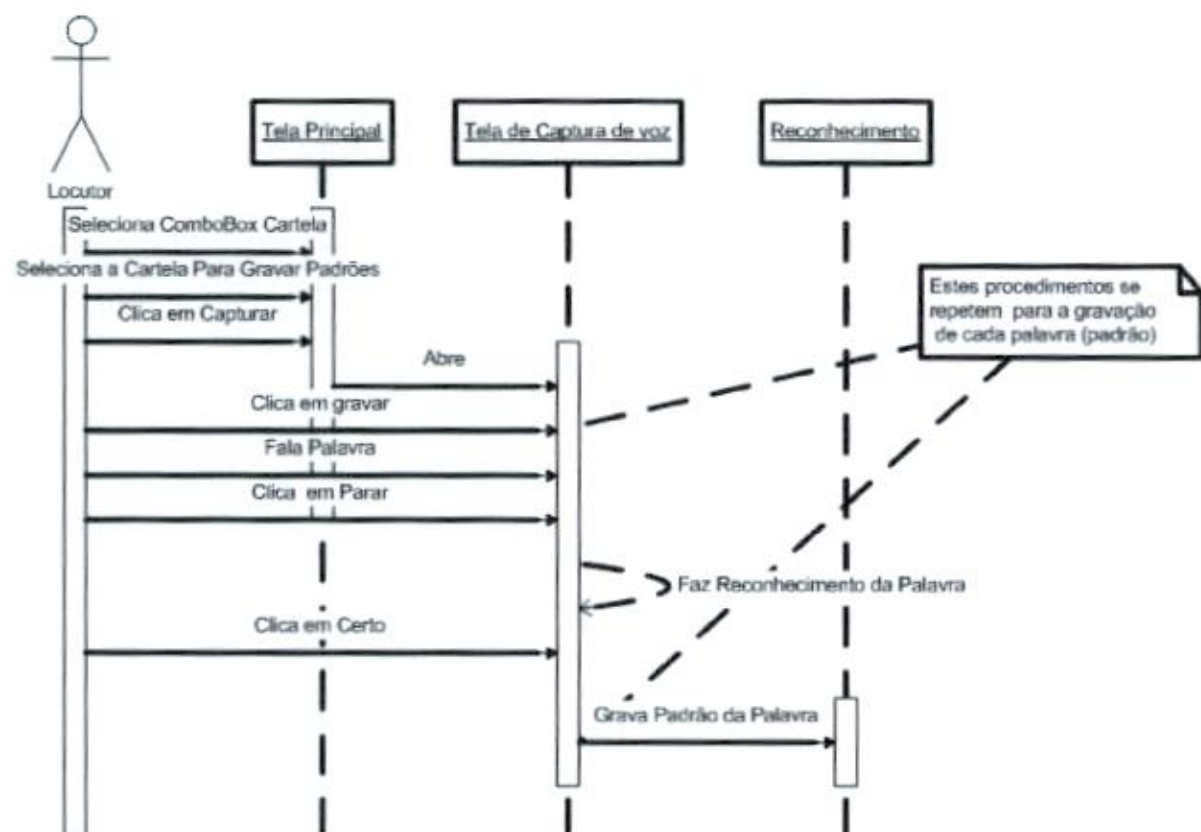


DIAGRAMA DE SEQÜÊNCIA - LABORATÓRIO - CAPTURAR - RECONHECIMENTO ERRÔNEO

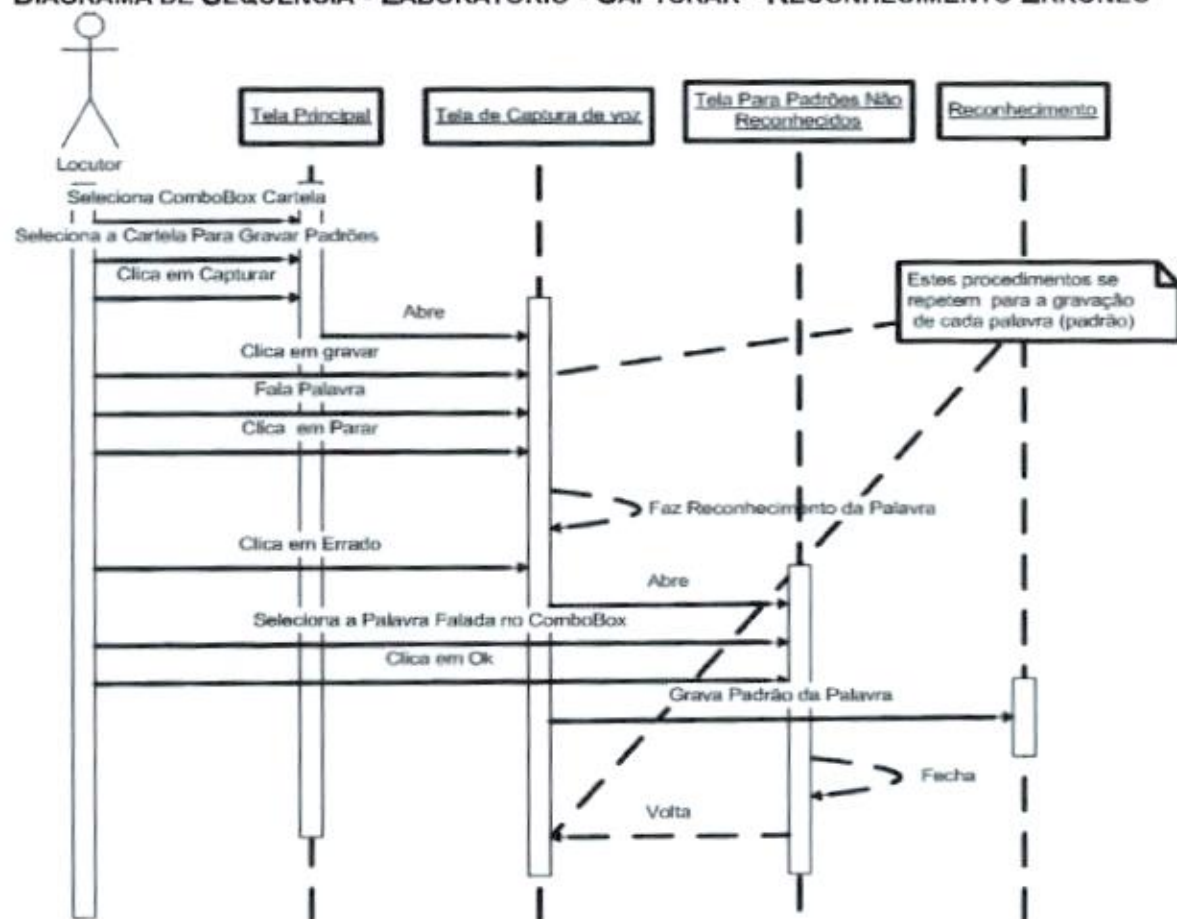


DIAGRAMA DE SEQÜÊNCIA - LABORATÓRIO - GRAVAR PADRÕES NO BANCO

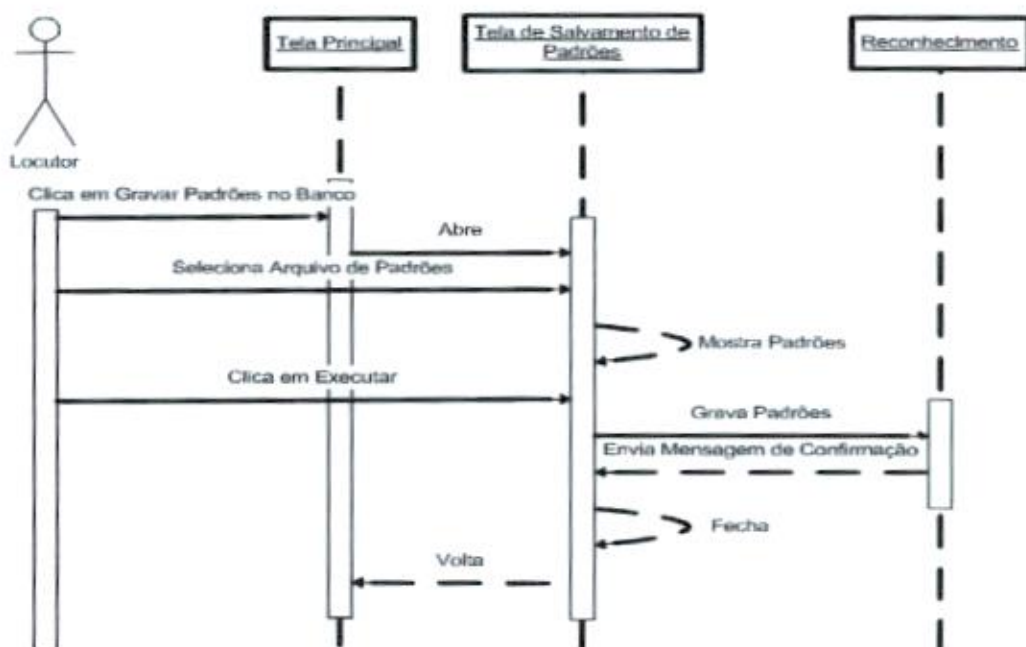


DIAGRAMA DE SEQÜÊNCIA - LABORATÓRIO - GRAVAR REDE NO BANCO



DIAGRAMA DE SEQÜÊNCIA - LABORATÓRIO - REFINADOR

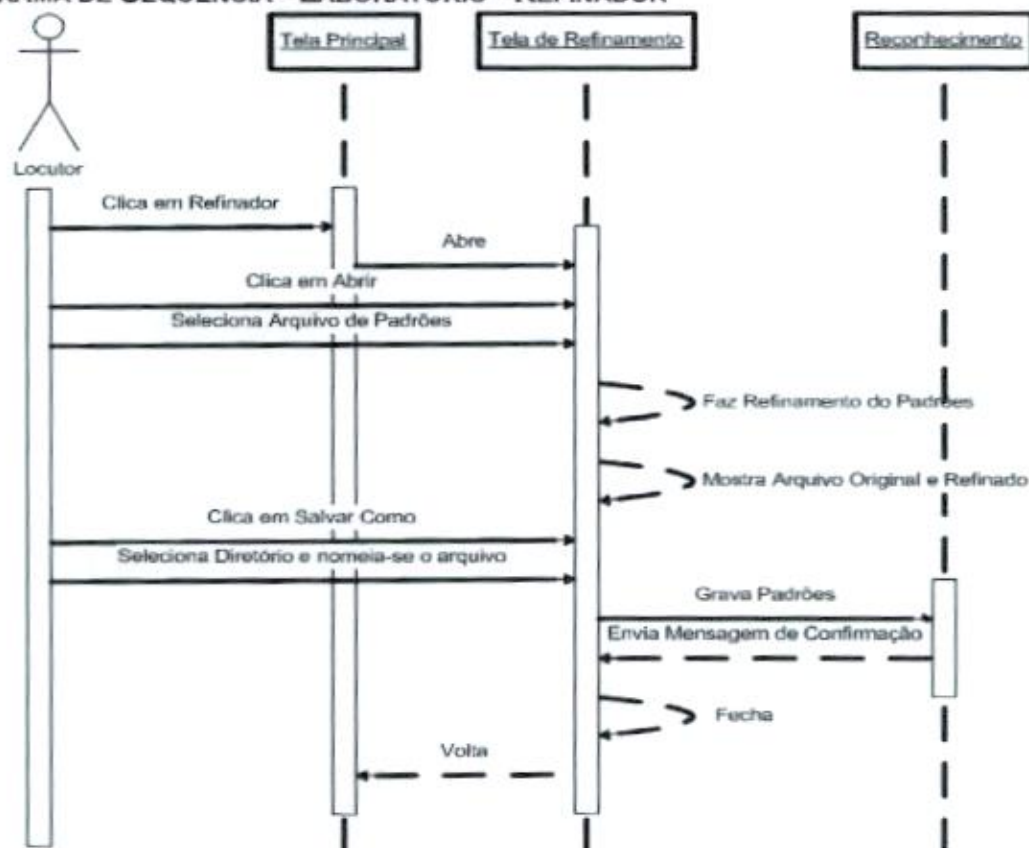


DIAGRAMA DE SEQÜÊNCIA - LABORATÓRIO – CONTADOR

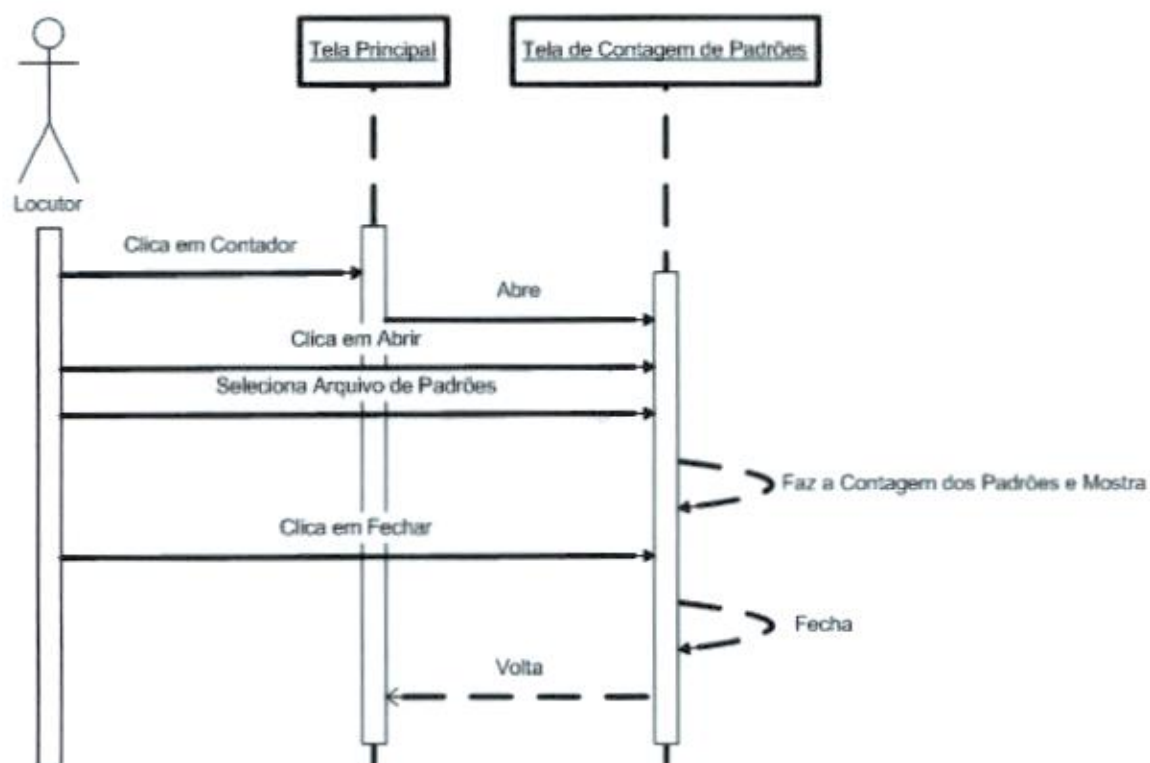


DIAGRAMA DE COLABORAÇÃO - LABORATÓRIO - CAPTURAR - RECONHECIMENTO CORRETO

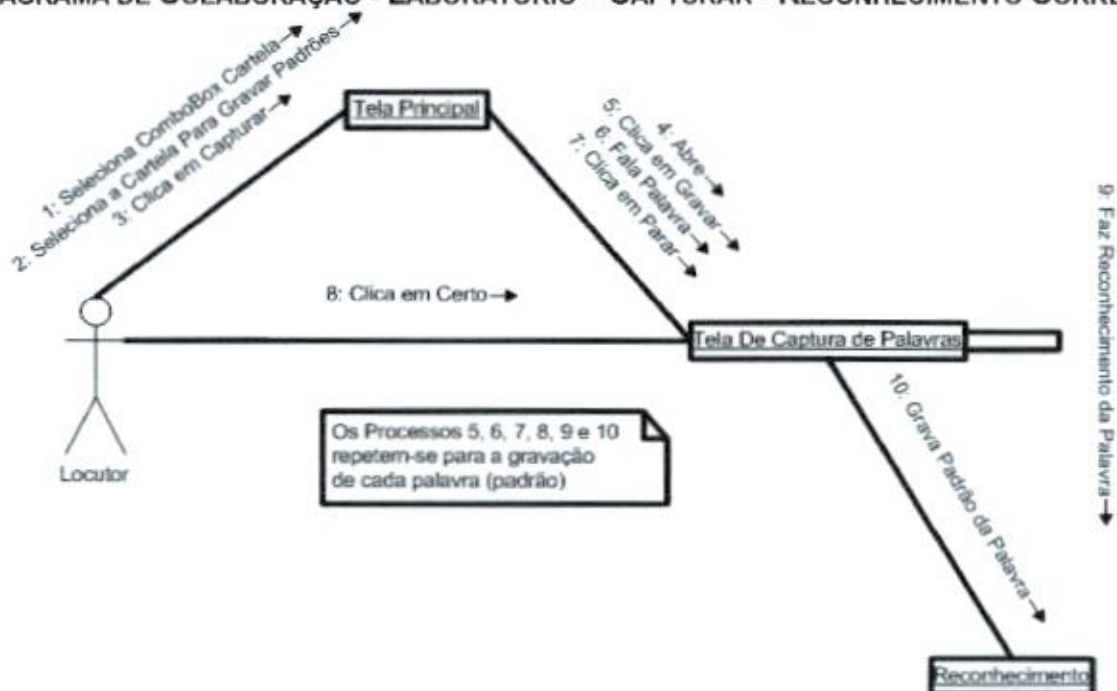


DIAGRAMA DE COLABORAÇÃO - LABORATÓRIO - CAPTURAR - RECONHECIMENTO ERRÔNEO

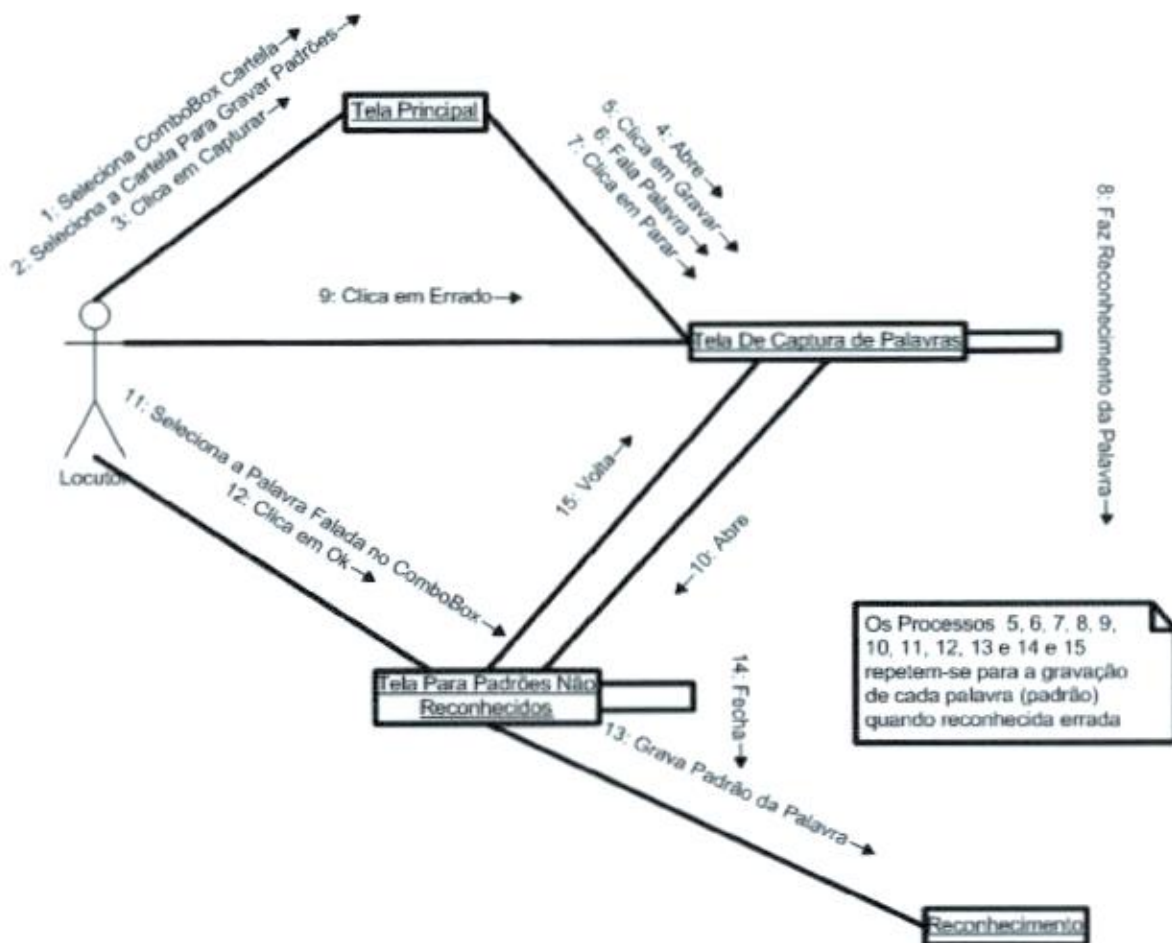


DIAGRAMA DE COLABORAÇÃO - LABORATÓRIO - GRAVAR PADRÕES NO BANCO

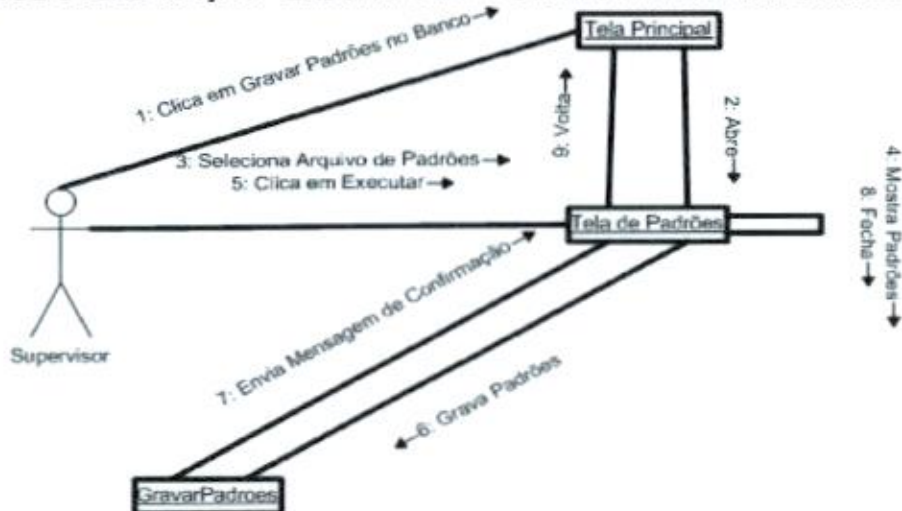


DIAGRAMA DE COLABORAÇÃO - LABORATÓRIO - GRAVAR REDE NO BANCO

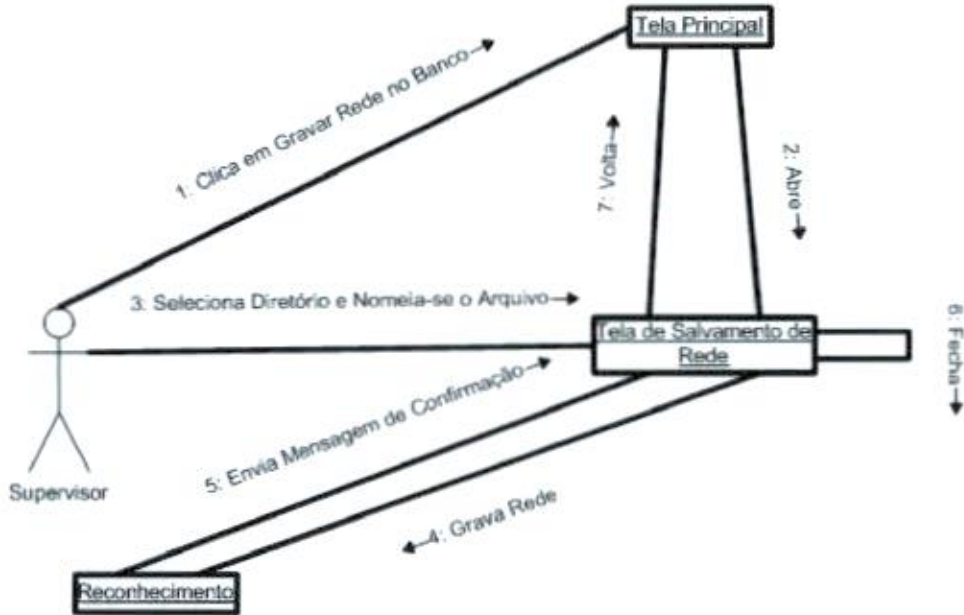


DIAGRAMA DE COLABORAÇÃO - LABORATÓRIO - REFINADOR



DIAGRAMA DE COLABORAÇÃO - LABORATÓRIO - CONTADOR

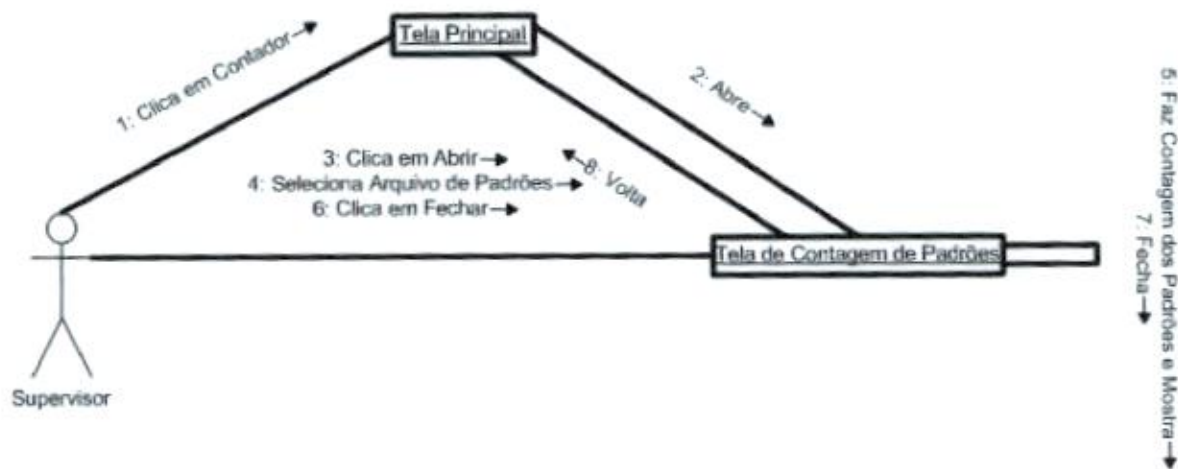


DIAGRAMA DE ESTADOS - LABORATÓRIO - RECONHECIMENTO.

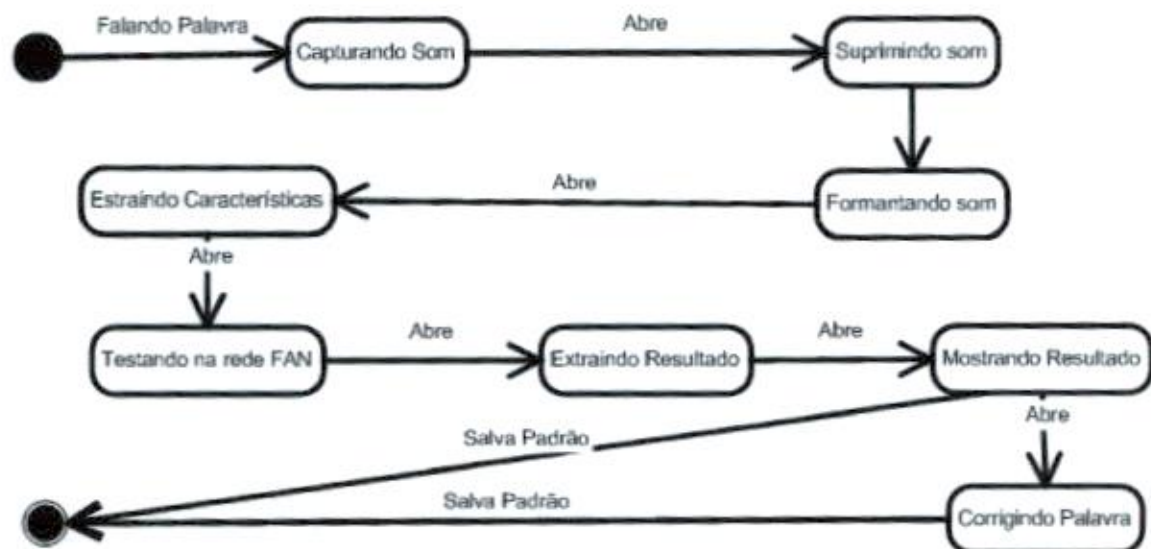


DIAGRAMA DE ESTADOS - LABORATÓRIO - REFINANDO PADRÕES.



DIAGRAMA DE ESTADOS - LABORATÓRIO - CONTANDO PADRÕES.

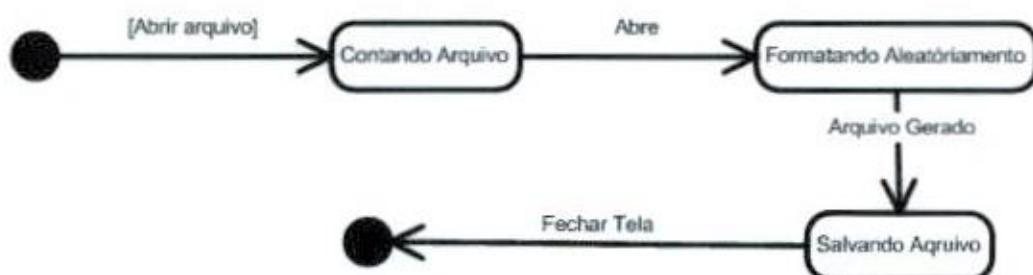


DIAGRAMA DE ATIVIDADES - LABORATÓRIO - RECONHECIMENTO.

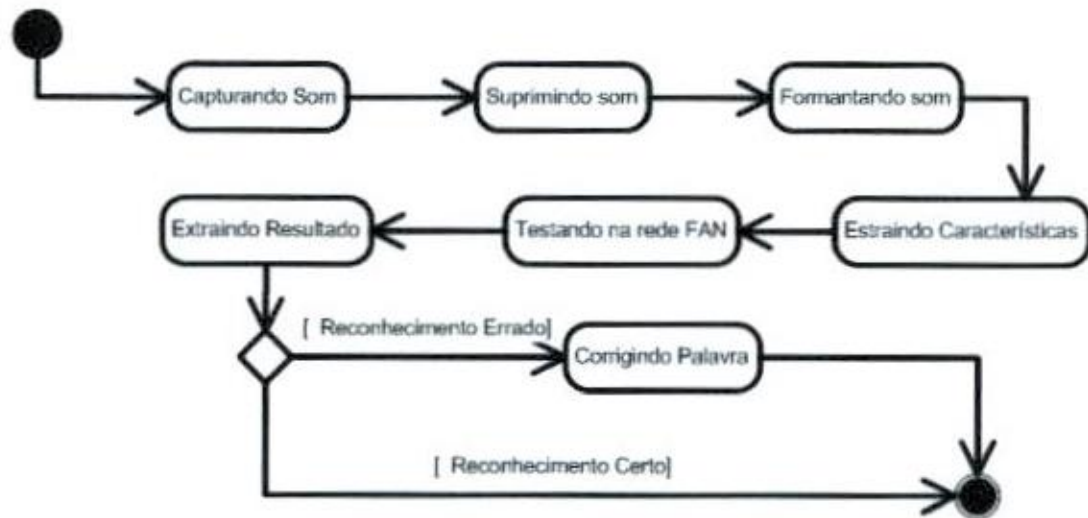


DIAGRAMA DE ESTADOS - LABORATÓRIO - REFINANDO PADRÕES.



DIAGRAMA DE ATIVIDADES - LABORATÓRIO - CONTANDO PADRÕES.



15.3 ANEXO III
DICIONÁRIO DE DADOS

DICIONÁRIO DE DADOS

Tabela aluno

CAMPO	TIPO	TAMANHO	OBS
alu_codigo	int		Não nulo Primary Key
alu_nome_completo	char	40	
alu_nome_sist	char	15	
alu_pai	char	40	
alu_mae	char	40	
alu_telefone	char	11	
alu_aplicador	int		
alu_data_cadastro	date		
alu_data_ult_alt	date		

Tabela aplicador

CAMPO	TIPO	TAMANHO	OBS
Apl_codigo	int		Não nulo Primary Key
apl_nome_completo	char	40	
Apl_nome_sist	char	15	
alu_login	char	15	
alu_senha	char	15	
apl_data_cadastro	date		
Apl_data_ult_alt	date		

Tabela local_aplicação

CAMPO	TIPO	TAMANHO	OBS
Local_codigo	int		Não nulo Primary Key
local_descricao	char	40	
local_contato	char	40	
local_telefone	char	11	
Local_data_cadastro	date		
local_data_ult_alt	date		

Tabela resultado

CAMPO	TIPO	TAMANHO	OBS
res_codigo	int		Não nulo Primary Key
res_historico	int	40	
res_resultado	boolean		
res_cartela	int	11	
Res_data_cadastro	date		
res_data_ult_alt	date		

Tabela Histórico

CAMPO	TIPO	TAMANHO	OBS
his_codigo	int		Não nulo Primary key
his_aplicador	Int		
his_aluno	Int		
his_local	Int		
his_status	int		
his_11	text		
his_12	text		
his_21	text		

his_22	text		
his_3	text		
His_4	text		
his_51	text		
his_52	text		
his_61	text		
his_62	text		
his_78	text		
his_91	text		
his_92	text		
his_11_stat	char(3)		
his_12_stat	char(3)		
his_21_stat	char(3)		
his_22_stat	char(3)		
his_3_stat	char(3)		
his_4_stat	char(3)		
his_51_stat	char(3)		
his_52_stat	char(3)		
his_61_stat	char(3)		
his_62_stat	char(3)		
his_78_stat	char(3)		
his_91_stat	char(3)		
his_92_stat	char(3)		
his_data_cadastro	date		
his_data_ult_alt	date		

Tabela carac_1

CAMPO	TIPO	TAMANHO	OBS
pl_codigo	int		
carac_1	double precision		
carac_2	double precision		
carac_3	double precision		
carac_4	double precision		

carac_5	double precision		
carac_6	double precision		
carac_7	double precision		

Tabela carac_2

CAMPO	TIPO	TAMANHO	OBS
pl_codigo	int		
carac_1	double precision		
carac_2	double precision		
carac_3	double precision		
carac_4	double precision		
carac_5	double precision		
carac_6	double precision		
carac_7	double precision		

Tabela carac_3

CAMPO	TIPO	TAMANHO	OBS
pl_codigo	int		
carac_1	double precision		
carac_2	double precision		
carac_3	double precision		
carac_4	double precision		
carac_5	double precision		
carac_6	double precision		
carac_7	double precision		

Tabela carac_4

CAMPO	TIPO	TAMANHO	OBS
pl_codigo	int		
carac_1	double precision		
carac_2	double precision		
carac_3	double precision		
carac_4	double precision		
carac_5	double precision		
carac_6	double precision		
carac_7	double precision		

Tabela carac_5

CAMPO	TIPO	TAMANHO	OBS
pl_codigo	int		
carac_1	double precision		
carac_2	double precision		
carac_3	double precision		
carac_4	double precision		
carac_5	double precision		
carac_6	double precision		
carac_7	double precision		

Tabela carac_6

CAMPO	TIPO	TAMANHO	OBS
pl_codigo	int		
carac_1	double precision		
carac_2	double precision		
carac_3	double precision		
carac_4	double precision		
carac_5	double precision		
carac_6	double precision		
carac_7	double precision		

Tabela carac_7

CAMPO	TIPO	TAMANHO	OBS
pl_codigo	int		
carac_1	double precision		
carac_2	double precision		
carac_3	double precision		
carac_4	double precision		
carac_5	double precision		
carac_6	double precision		
carac_7	double precision		

Tabela carac_8

CAMPO	TIPO	TAMANHO	OBS
pl_codigo	int		
carac_1	double precision		
carac_2	double precision		
carac_3	double precision		
carac_4	double precision		
carac_5	double precision		
carac_6	double precision		
carac_7	double precision		

Tabela carac_9

CAMPO	TIPO	TAMANHO	OBS
pl_codigo	int		
carac_1	double precision		
carac_2	double precision		
carac_3	double precision		
carac_4	double precision		
carac_5	double precision		
carac_6	double precision		
carac_7	double precision		

Tabela carac_10

CAMPO	TIPO	TAMANHO	OBS
PI_codigo	int		
carac_1	double precision		
carac_2	double precision		
carac_3	double precision		
carac_4	double precision		
carac_5	double precision		
carac_6	double precision		
carac_7	double precision		

Tabela carac_11

CAMPO	TIPO	TAMANHO	OBS
PI_codigo	int		
carac_1	double precision		
carac_2	double precision		
carac_3	double precision		
carac_4	double precision		
carac_5	double precision		
carac_6	double precision		
carac_7	double precision		

Tabela carac_12

CAMPO	TIPO	TAMANHO	OBS
PI_codigo	int		
carac_1	double precision		
carac_2	double precision		
carac_3	double precision		
carac_4	double precision		
carac_5	double precision		
carac_6	double precision		
carac_7	double precision		

Tabela carac_13

CAMPO	TIPO	TAMANHO	OBS
PI_codigo	int		
carac_1	double precision		
carac_2	double precision		
carac_3	double precision		
carac_4	double precision		
carac_5	double precision		
carac_6	double precision		
carac_7	double precision		

Tabela carac_14

CAMPO	TIPO	TAMANHO	OBS
PI_codigo	int		
carac_1	double precision		
carac_2	double precision		
carac_3	double precision		
carac_4	double precision		
carac_5	double precision		
carac_6	double precision		
carac_7	double precision		

Tabela carac_15

CAMPO	TIPO	TAMANHO	OBS
pl_codigo	int		
carac_1	double precision		
carac_2	double precision		
carac_3	double precision		
carac_4	double precision		
carac_5	double precision		
carac_6	double precision		
carac_7	double precision		

Tabela Testes 1.1, 2.1 e 5.1

CAMPO	TIPO	TAMANHO	OBS
Cod_teste	Int		Não nulo Primary Key
Cod_rede	Int		Não nulo Primay key
nome_do_teste	Char	3	
nome_do_aplicador	Char	15	
nome_do_avaliado	Char	15	
Nome_do_local	Char	40	
Cartela_1_figura_1	Char	3	
Cartela_1_figura_2	Char	3	
Cartela_1_figura_3	Char	3	
Cartela_1_figura_4	Char	3	
Cartela_1_figura_5	Char	3	
Cartela_1_figura_6	Char	3	
Cartela_2_figura_1	Char	3	
Cartela_2_figura_2	Char	3	
Cartela_2_figura_3	Char	3	
Cartela_2_figura_4	Char	3	
Cartela_2_figura_5	Char	3	
Cartela_2_figura_6	Char	3	
Cartela_3_figura_1	Char	3	
Cartela_3_figura_2	Char	3	
Cartela_3_figura_3	Char	3	
Cartela_3_figura_4	Char	3	
Cartela_3_figura_5	Char	3	
Cartela_3_figura_6	Char	3	
Cartela_4_figura_1	Char	3	
Cartela_4_figura_2	Char	3	
Cartela_4_figura_3	Char	3	
Cartela_4_figura_4	Char	3	
Cartela_4_figura_5	Char	3	
Cartela_4_figura_6	Char	3	
Cartela_5_figura_1	Char	3	

Cartela_5_figura_2	Char	3	
Cartela_5_figura_3	Char	3	
Cartela_5_figura_4	Char	3	
Cartela_5_figura_5	Char	3	
Cartela_5_figura_6	Char	3	
Cartela_6_figura_1	Char	3	
Cartela_6_figura_2	Char	3	
Cartela_6_figura_3	Char	3	
Cartela_6_figura_4	Char	3	
Cartela_6_figura_5	Char	3	
Cartela_6_figura_6	Char	3	
Cartela_7_figura_1	Char	3	
Cartela_7_figura_2	Char	3	
Cartela_7_figura_3	Char	3	
Cartela_7_figura_4	Char	3	
Cartela_7_figura_5	Char	3	
Cartela_7_figura_6	Char	3	
Cartela_8_figura_1	Char	3	
Cartela_8_figura_2	Char	3	
Cartela_8_figura_3	Char	3	
Cartela_8_figura_4	Char	3	
Cartela_8_figura_5	Char	3	
Cartela_8_figura_6	Char	3	
Cartela_9_figura_1	Char	3	
Cartela_9_figura_2	Char	3	
Cartela_9_figura_3	Char	3	
Cartela_9_figura_4	Char	3	
Cartela_9_figura_5	Char	3	
Cartela_9_figura_6	Char	3	
Cartela_10_figura_1	Char	3	
Cartela_10_figura_2	Char	3	
Cartela_10_figura_3	Char	3	
Cartela_10_figura_4	Char	3	
Cartela_10_figura_5	Char	3	
Cartela_10_figura_6	Char	3	
Cartela_11_figura_1	Char	3	

Cartela_11_figura_2	Char	3	
Cartela_11_figura_3	Char	3	
Cartela_11_figura_4	Char	3	
Cartela_11_figura_5	Char	3	
Cartela_11_figura_6	Char	3	
Cartela_12_figura_1	Char	3	
Cartela_12_figura_2	Char	3	
Cartela_12_figura_3	Char	3	
Cartela_12_figura_4	Char	3	
Cartela_12_figura_5	Char	3	
Cartela_12_figura_6	Char	3	
Cartela_13_figura_1	Char	3	
Cartela_13_figura_2	Char	3	
Cartela_13_figura_3	Char	3	
Cartela_13_figura_4	Char	3	
Cartela_13_figura_5	Char	3	
Cartela_13_figura_6	Char	3	
Cartela_14_figura_1	Char	3	
Cartela_14_figura_2	Char	3	
Cartela_14_figura_3	Char	3	
Cartela_14_figura_4	Char	3	
Cartela_14_figura_5	Char	3	
Cartela_14_figura_6	Char	3	
Cartela_15_figura_1	Char	3	
Cartela_15_figura_2	Char	3	
Cartela_15_figura_3	Char	3	
Cartela_15_figura_4	Char	3	
Cartela_15_figura_5	Char	3	
Cartela_15_figura_6	Char	3	
Resultado_do_teste_1.1	Char	50	
Resultado_do_teste_2.1	Char	50	
Resultado_do_teste_5.1	Char	50	
Teste_1.1_data	date		
Teste_2.1_data	Date		
Teste_5.1_data	date		

Tabela Rede_1

CAMPO	TIPO	TAMANHO	OBS
rede_neuronio	int		
rede_carac	int		
rede_range_0	double precision		
rede_range_1	double precision		
rede_range_2	double precision		
rede_range_3	double precision		
rede_range_4	double precision		
rede_range_5	double precision		
rede_range_6	double precision		
rede_range_7	double precision		
rede_range_8	double precision		
rede_range_9	double precision		
rede_range_10	double precision		
rede_range_11	double precision		
rede_range_12	double precision		
rede_range_13	double precision		
rede_range_14	double precision		
rede_range_15	double precision		
rede_range_16	double precision		
rede_range_17	double precision		
rede_range_18	double precision		
rede_range_19	double precision		
rede_range_20	double precision		
rede_range_21	double precision		
rede_range_22	double precision		
rede_range_23	double precision		
rede_range_24	double precision		
rede_range_25	double precision		
rede_range_26	double precision		
rede_range_27	double precision		
rede_range_28	double precision		
rede_range_29	double precision		
rede_range_30	double precision		

rede_range_31	double precision		
rede_range_32	double precision		
rede_range_33	double precision		
rede_range_34	double precision		
rede_range_35	double precision		
rede_range_36	double precision		
rede_range_37	double precision		
rede_range_38	double precision		
rede_range_39	double precision		
rede_range_40	double precision		
rede_range_41	double precision		
rede_range_42	double precision		
rede_range_43	double precision		
rede_range_44	double precision		
rede_range_45	double precision		
rede_range_46	double precision		
rede_range_47	double precision		
rede_range_48	double precision		
rede_range_49	double precision		
rede_range_50	double precision		
rede_range_51	double precision		
rede_range_52	double precision		
rede_range_53	double precision		
rede_range_54	double precision		
rede_range_55	double precision		
rede_range_56	double precision		
rede_range_57	double precision		
rede_range_58	double precision		
rede_range_59	double precision		
rede_range_60	double precision		
rede_range_61	double precision		
rede_range_62	double precision		
rede_range_63	double precision		
rede_range_64	double precision		
rede_range_65	double precision		
rede_range_66	double precision		

rede_range_67	double precision		
rede_range_68	double precision		
rede_range_69	double precision		
rede_range_70	double precision		
rede_range_71	double precision		
rede_range_72	double precision		
rede_range_73	double precision		
rede_range_74	double precision		
rede_range_75	double precision		
rede_range_76	double precision		
rede_range_77	double precision		
rede_range_78	double precision		
rede_range_79	double precision		
rede_range_80	double precision		
rede_range_81	double precision		
rede_range_82	double precision		
rede_range_83	double precision		
rede_range_84	double precision		
rede_range_85	double precision		
rede_range_86	double precision		
rede_range_87	double precision		
rede_range_88	double precision		
rede_range_89	double precision		
rede_range_90	double precision		
rede_range_91	double precision		
rede_range_92	double precision		
rede_range_93	double precision		
rede_range_94	double precision		
rede_range_95	double precision		
rede_range_96	double precision		
rede_range_97	double precision		
rede_range_98	double precision		
rede_range_99	double precision		
rede_range_100	double precision		

Tabela Rede_2

CAMPO	TIPO	TAMANHO	OBS
rede_neuronio	int		
rede_carac	int		
rede_range_0	double precision		
rede_range_1	double precision		
rede_range_2	double precision		
rede_range_3	double precision		
rede_range_4	double precision		
rede_range_5	double precision		
rede_range_6	double precision		
rede_range_7	double precision		
rede_range_8	double precision		
rede_range_9	double precision		
rede_range_10	double precision		
rede_range_11	double precision		
rede_range_12	double precision		
rede_range_13	double precision		
rede_range_14	double precision		
rede_range_15	double precision		
rede_range_16	double precision		
rede_range_17	double precision		
rede_range_18	double precision		
rede_range_19	double precision		
rede_range_20	double precision		
rede_range_21	double precision		
rede_range_22	double precision		
rede_range_23	double precision		
rede_range_24	double precision		
rede_range_25	double precision		
rede_range_26	double precision		
rede_range_27	double precision		
rede_range_28	double precision		
rede_range_29	double precision		
rede_range_30	double precision		

rede_range_31	double precision		
rede_range_32	double precision		
rede_range_33	double precision		
rede_range_34	double precision		
rede_range_35	double precision		
rede_range_36	double precision		
rede_range_37	double precision		
rede_range_38	double precision		
rede_range_39	double precision		
rede_range_40	double precision		
rede_range_41	double precision		
rede_range_42	double precision		
rede_range_43	double precision		
rede_range_44	double precision		
rede_range_45	double precision		
rede_range_46	double precision		
rede_range_47	double precision		
rede_range_48	double precision		
rede_range_49	double precision		
rede_range_50	double precision		
rede_range_51	double precision		
rede_range_52	double precision		
rede_range_53	double precision		
rede_range_54	double precision		
rede_range_55	double precision		
rede_range_56	double precision		
rede_range_57	double precision		
rede_range_58	double precision		
rede_range_59	double precision		
rede_range_60	double precision		
rede_range_61	double precision		
rede_range_62	double precision		
rede_range_63	double precision		
rede_range_64	double precision		
rede_range_65	double precision		
rede_range_66	double precision		

rede_range_67	double precision		
rede_range_68	double precision		
rede_range_69	double precision		
rede_range_70	double precision		
rede_range_71	double precision		
rede_range_72	double precision		
rede_range_73	double precision		
rede_range_74	double precision		
rede_range_75	double precision		
rede_range_76	double precision		
rede_range_77	double precision		
rede_range_78	double precision		
rede_range_79	double precision		
rede_range_80	double precision		
rede_range_81	double precision		
rede_range_82	double precision		
rede_range_83	double precision		
rede_range_84	double precision		
rede_range_85	double precision		
rede_range_86	double precision		
rede_range_87	double precision		
rede_range_88	double precision		
rede_range_89	double precision		
rede_range_90	double precision		
rede_range_91	double precision		
rede_range_92	double precision		
rede_range_93	double precision		
rede_range_94	double precision		
rede_range_95	double precision		
rede_range_96	double precision		
rede_range_97	double precision		
rede_range_98	double precision		
rede_range_99	double precision		
rede_range_100	double precision		

Tabela rede_3

CAMPO	TIPO	TAMANHO	OBS
rede_neuronio	int		
rede_carac	int		
rede_range_0	double precision		
rede_range_1	double precision		
rede_range_2	double precision		
rede_range_3	double precision		
rede_range_4	double precision		
rede_range_5	double precision		
rede_range_6	double precision		
rede_range_7	double precision		
rede_range_8	double precision		
rede_range_9	double precision		
rede_range_10	double precision		
rede_range_11	double precision		
rede_range_12	double precision		
rede_range_13	double precision		
rede_range_14	double precision		
rede_range_15	double precision		
rede_range_16	double precision		
rede_range_17	double precision		
rede_range_18	double precision		
rede_range_19	double precision		
rede_range_20	double precision		
rede_range_21	double precision		
rede_range_22	double precision		
rede_range_23	double precision		
rede_range_24	double precision		
rede_range_25	double precision		
rede_range_26	double precision		
rede_range_27	double precision		
rede_range_28	double precision		
rede_range_29	double precision		
rede_range_30	double precision		

rede_range_31	double precision		
rede_range_32	double precision		
rede_range_33	double precision		
rede_range_34	double precision		
rede_range_35	double precision		
rede_range_36	double precision		
rede_range_37	double precision		
rede_range_38	double precision		
rede_range_39	double precision		
rede_range_40	double precision		
rede_range_41	double precision		
rede_range_42	double precision		
rede_range_43	double precision		
rede_range_44	double precision		
rede_range_45	double precision		
rede_range_46	double precision		
rede_range_47	double precision		
rede_range_48	double precision		
rede_range_49	double precision		
rede_range_50	double precision		
rede_range_51	double precision		
rede_range_52	double precision		
rede_range_53	double precision		
rede_range_54	double precision		
rede_range_55	double precision		
rede_range_56	double precision		
rede_range_57	double precision		
rede_range_58	double precision		
rede_range_59	double precision		
rede_range_60	double precision		
rede_range_61	double precision		
rede_range_62	double precision		
rede_range_63	double precision		
rede_range_64	double precision		
rede_range_65	double precision		
rede_range_66	double precision		

rede_range_67	double precision		
rede_range_68	double precision		
rede_range_69	double precision		
rede_range_70	double precision		
rede_range_71	double precision		
rede_range_72	double precision		
rede_range_73	double precision		
rede_range_74	double precision		
rede_range_75	double precision		
rede_range_76	double precision		
rede_range_77	double precision		
rede_range_78	double precision		
rede_range_79	double precision		
rede_range_80	double precision		
rede_range_81	double precision		
rede_range_82	double precision		
rede_range_83	double precision		
rede_range_84	double precision		
rede_range_85	double precision		
rede_range_86	double precision		
rede_range_87	double precision		
rede_range_88	double precision		
rede_range_89	double precision		
rede_range_90	double precision		
rede_range_91	double precision		
rede_range_92	double precision		
rede_range_93	double precision		
rede_range_94	double precision		
rede_range_95	double precision		
rede_range_96	double precision		
rede_range_97	double precision		
rede_range_98	double precision		
rede_range_99	double precision		
rede_range_100	double precision		

Tabela rede_4

CAMPO	TIPO	TAMANHO	OBS
rede_neuronio	int		
rede_carac	int		
rede_range_0	double precision		
rede_range_1	double precision		
rede_range_2	double precision		
rede_range_3	double precision		
rede_range_4	double precision		
rede_range_5	double precision		
rede_range_6	double precision		
rede_range_7	double precision		
rede_range_8	double precision		
rede_range_9	double precision		
rede_range_10	double precision		
rede_range_11	double precision		
rede_range_12	double precision		
rede_range_13	double precision		
rede_range_14	double precision		
rede_range_15	double precision		
rede_range_16	double precision		
rede_range_17	double precision		
rede_range_18	double precision		
rede_range_19	double precision		
rede_range_20	double precision		
rede_range_21	double precision		
rede_range_22	double precision		
rede_range_23	double precision		
rede_range_24	double precision		
rede_range_25	double precision		
rede_range_26	double precision		
rede_range_27	double precision		
rede_range_28	double precision		
rede_range_29	double precision		
rede_range_30	double precision		

rede_range_31	double precision		
rede_range_32	double precision		
rede_range_33	double precision		
rede_range_34	double precision		
rede_range_35	double precision		
rede_range_36	double precision		
rede_range_37	double precision		
rede_range_38	double precision		
rede_range_39	double precision		
rede_range_40	double precision		
rede_range_41	double precision		
rede_range_42	double precision		
rede_range_43	double precision		
rede_range_44	double precision		
rede_range_45	double precision		
rede_range_46	double precision		
rede_range_47	double precision		
rede_range_48	double precision		
rede_range_49	double precision		
rede_range_50	double precision		
rede_range_51	double precision		
rede_range_52	double precision		
rede_range_53	double precision		
rede_range_54	double precision		
rede_range_55	double precision		
rede_range_56	double precision		
rede_range_57	double precision		
rede_range_58	double precision		
rede_range_59	double precision		
rede_range_60	double precision		
rede_range_61	double precision		
rede_range_62	double precision		
rede_range_63	double precision		
rede_range_64	double precision		
rede_range_65	double precision		
rede_range_66	double precision		

rede_range_67	double precision		
rede_range_68	double precision		
rede_range_69	double precision		
rede_range_70	double precision		
rede_range_71	double precision		
rede_range_72	double precision		
rede_range_73	double precision		
rede_range_74	double precision		
rede_range_75	double precision		
rede_range_76	double precision		
rede_range_77	double precision		
rede_range_78	double precision		
rede_range_79	double precision		
rede_range_80	double precision		
rede_range_81	double precision		
rede_range_82	double precision		
rede_range_83	double precision		
rede_range_84	double precision		
rede_range_85	double precision		
rede_range_86	double precision		
rede_range_87	double precision		
rede_range_88	double precision		
rede_range_89	double precision		
rede_range_90	double precision		
rede_range_91	double precision		
rede_range_92	double precision		
rede_range_93	double precision		
rede_range_94	double precision		
rede_range_95	double precision		
rede_range_96	double precision		
rede_range_97	double precision		
rede_range_98	double precision		
rede_range_99	double precision		
rede_range_100	double precision		

Tabela rede_5

CAMPO	TIPO	TAMANHO	OBS
rede_neuronio	int		
rede_carac	int		
rede_range_0	double precision		
rede_range_1	double precision		
rede_range_2	double precision		
rede_range_3	double precision		
rede_range_4	double precision		
rede_range_5	double precision		
rede_range_6	double precision		
rede_range_7	double precision		
rede_range_8	double precision		
rede_range_9	double precision		
rede_range_10	double precision		
rede_range_11	double precision		
rede_range_12	double precision		
rede_range_13	double precision		
rede_range_14	double precision		
rede_range_15	double precision		
rede_range_16	double precision		
rede_range_17	double precision		
rede_range_18	double precision		
rede_range_19	double precision		
rede_range_20	double precision		
rede_range_21	double precision		
rede_range_22	double precision		
rede_range_23	double precision		
rede_range_24	double precision		
rede_range_25	double precision		
rede_range_26	double precision		
rede_range_27	double precision		
rede_range_28	double precision		
rede_range_29	double precision		
rede_range_30	double precision		

rede_range_31	double precision		
rede_range_32	double precision		
rede_range_33	double precision		
rede_range_34	double precision		
rede_range_35	double precision		
rede_range_36	double precision		
rede_range_37	double precision		
rede_range_38	double precision		
rede_range_39	double precision		
rede_range_40	double precision		
rede_range_41	double precision		
rede_range_42	double precision		
rede_range_43	double precision		
rede_range_44	double precision		
rede_range_45	double precision		
rede_range_46	double precision		
rede_range_47	double precision		
rede_range_48	double precision		
rede_range_49	double precision		
rede_range_50	double precision		
rede_range_51	double precision		
rede_range_52	double precision		
rede_range_53	double precision		
rede_range_54	double precision		
rede_range_55	double precision		
rede_range_56	double precision		
rede_range_57	double precision		
rede_range_58	double precision		
rede_range_59	double precision		
rede_range_60	double precision		
rede_range_61	double precision		
rede_range_62	double precision		
rede_range_63	double precision		
rede_range_64	double precision		
rede_range_65	double precision		
rede_range_66	double precision		

rede_range_67	double precision		
rede_range_68	double precision		
rede_range_69	double precision		
rede_range_70	double precision		
rede_range_71	double precision		
rede_range_72	double precision		
rede_range_73	double precision		
rede_range_74	double precision		
rede_range_75	double precision		
rede_range_76	double precision		
rede_range_77	double precision		
rede_range_78	double precision		
rede_range_79	double precision		
rede_range_80	double precision		
rede_range_81	double precision		
rede_range_82	double precision		
rede_range_83	double precision		
rede_range_84	double precision		
rede_range_85	double precision		
rede_range_86	double precision		
rede_range_87	double precision		
rede_range_88	double precision		
rede_range_89	double precision		
rede_range_90	double precision		
rede_range_91	double precision		
rede_range_92	double precision		
rede_range_93	double precision		
rede_range_94	double precision		
rede_range_95	double precision		
rede_range_96	double precision		
rede_range_97	double precision		
rede_range_98	double precision		
rede_range_99	double precision		
rede_range_100	double precision		

Tabela rede_6

CAMPO	TIPO	TAMANHO	OBS
rede_neuronio	int		
rede_carac	int		
rede_range_0	double precision		
rede_range_1	double precision		
rede_range_2	double precision		
rede_range_3	double precision		
rede_range_4	double precision		
rede_range_5	double precision		
rede_range_6	double precision		
rede_range_7	double precision		
rede_range_8	double precision		
rede_range_9	double precision		
rede_range_10	double precision		
rede_range_11	double precision		
rede_range_12	double precision		
rede_range_13	double precision		
rede_range_14	double precision		
rede_range_15	double precision		
rede_range_16	double precision		
rede_range_17	double precision		
rede_range_18	double precision		
rede_range_19	double precision		
rede_range_20	double precision		
rede_range_21	double precision		
rede_range_22	double precision		
rede_range_23	double precision		
rede_range_24	double precision		
rede_range_25	double precision		
rede_range_26	double precision		
rede_range_27	double precision		
rede_range_28	double precision		
rede_range_29	double precision		
rede_range_30	double precision		

rede_range_31	double precision		
rede_range_32	double precision		
rede_range_33	double precision		
rede_range_34	double precision		
rede_range_35	double precision		
rede_range_36	double precision		
rede_range_37	double precision		
rede_range_38	double precision		
rede_range_39	double precision		
rede_range_40	double precision		
rede_range_41	double precision		
rede_range_42	double precision		
rede_range_43	double precision		
rede_range_44	double precision		
rede_range_45	double precision		
rede_range_46	double precision		
rede_range_47	double precision		
rede_range_48	double precision		
rede_range_49	double precision		
rede_range_50	double precision		
rede_range_51	double precision		
rede_range_52	double precision		
rede_range_53	double precision		
rede_range_54	double precision		
rede_range_55	double precision		
rede_range_56	double precision		
rede_range_57	double precision		
rede_range_58	double precision		
rede_range_59	double precision		
rede_range_60	double precision		
rede_range_61	double precision		
rede_range_62	double precision		
rede_range_63	double precision		
rede_range_64	double precision		
rede_range_65	double precision		
rede_range_66	double precision		

rede_range_67	double precision		
rede_range_68	double precision		
rede_range_69	double precision		
rede_range_70	double precision		
rede_range_71	double precision		
rede_range_72	double precision		
rede_range_73	double precision		
rede_range_74	double precision		
rede_range_75	double precision		
rede_range_76	double precision		
rede_range_77	double precision		
rede_range_78	double precision		
rede_range_79	double precision		
rede_range_80	double precision		
rede_range_81	double precision		
rede_range_82	double precision		
rede_range_83	double precision		
rede_range_84	double precision		
rede_range_85	double precision		
rede_range_86	double precision		
rede_range_87	double precision		
rede_range_88	double precision		
rede_range_89	double precision		
rede_range_90	double precision		
rede_range_91	double precision		
rede_range_92	double precision		
rede_range_93	double precision		
rede_range_94	double precision		
rede_range_95	double precision		
rede_range_96	double precision		
rede_range_97	double precision		
rede_range_98	double precision		
rede_range_99	double precision		
rede_range_100	double precision		

Tabela rede_7

CAMPO	TIPO	TAMANHO	OBS
rede_neuronio	int		
rede_carac	int		
rede_range_0	double precision		
rede_range_1	double precision		
rede_range_2	double precision		
rede_range_3	double precision		
rede_range_4	double precision		
rede_range_5	double precision		
rede_range_6	double precision		
rede_range_7	double precision		
rede_range_8	double precision		
rede_range_9	double precision		
rede_range_10	double precision		
rede_range_11	double precision		
rede_range_12	double precision		
rede_range_13	double precision		
rede_range_14	double precision		
rede_range_15	double precision		
rede_range_16	double precision		
rede_range_17	double precision		
rede_range_18	double precision		
rede_range_19	double precision		
rede_range_20	double precision		
rede_range_21	double precision		
rede_range_22	double precision		
rede_range_23	double precision		
rede_range_24	double precision		
rede_range_25	double precision		
rede_range_26	double precision		
rede_range_27	double precision		
rede_range_28	double precision		
rede_range_29	double precision		
rede_range_30	double precision		

rede_range_31	double precision		
rede_range_32	double precision		
rede_range_33	double precision		
rede_range_34	double precision		
rede_range_35	double precision		
rede_range_36	double precision		
rede_range_37	double precision		
rede_range_38	double precision		
rede_range_39	double precision		
rede_range_40	double precision		
rede_range_41	double precision		
rede_range_42	double precision		
rede_range_43	double precision		
rede_range_44	double precision		
rede_range_45	double precision		
rede_range_46	double precision		
rede_range_47	double precision		
rede_range_48	double precision		
rede_range_49	double precision		
rede_range_50	double precision		
rede_range_51	double precision		
rede_range_52	double precision		
rede_range_53	double precision		
rede_range_54	double precision		
rede_range_55	double precision		
rede_range_56	double precision		
rede_range_57	double precision		
rede_range_58	double precision		
rede_range_59	double precision		
rede_range_60	double precision		
rede_range_61	double precision		
rede_range_62	double precision		
rede_range_63	double precision		
rede_range_64	double precision		
rede_range_65	double precision		
rede_range_66	double precision		

rede_range_67	double precision		
rede_range_68	double precision		
rede_range_69	double precision		
rede_range_70	double precision		
rede_range_71	double precision		
rede_range_72	double precision		
rede_range_73	double precision		
rede_range_74	double precision		
rede_range_75	double precision		
rede_range_76	double precision		
rede_range_77	double precision		
rede_range_78	double precision		
rede_range_79	double precision		
rede_range_80	double precision		
rede_range_81	double precision		
rede_range_82	double precision		
rede_range_83	double precision		
rede_range_84	double precision		
rede_range_85	double precision		
rede_range_86	double precision		
rede_range_87	double precision		
rede_range_88	double precision		
rede_range_89	double precision		
rede_range_90	double precision		
rede_range_91	double precision		
rede_range_92	double precision		
rede_range_93	double precision		
rede_range_94	double precision		
rede_range_95	double precision		
rede_range_96	double precision		
rede_range_97	double precision		
rede_range_98	double precision		
rede_range_99	double precision		
rede_range_100	double precision		

Tabela rede_8

CAMPO	TIPO	TAMANHO	OBS
rede_neuronio	int		
rede_carac	int		
rede_range_0	double precision		
rede_range_1	double precision		
rede_range_2	double precision		
rede_range_3	double precision		
rede_range_4	double precision		
rede_range_5	double precision		
rede_range_6	double precision		
rede_range_7	double precision		
rede_range_8	double precision		
rede_range_9	double precision		
rede_range_10	double precision		
rede_range_11	double precision		
rede_range_12	double precision		
rede_range_13	double precision		
rede_range_14	double precision		
rede_range_15	double precision		
rede_range_16	double precision		
rede_range_17	double precision		
rede_range_18	double precision		
rede_range_19	double precision		
rede_range_20	double precision		
rede_range_21	double precision		
rede_range_22	double precision		
rede_range_23	double precision		
rede_range_24	double precision		
rede_range_25	double precision		
rede_range_26	double precision		
rede_range_27	double precision		
rede_range_28	double precision		
rede_range_29	double precision		
rede_range_30	double precision		

rede_range_31	double precision		
rede_range_32	double precision		
rede_range_33	double precision		
rede_range_34	double precision		
rede_range_35	double precision		
rede_range_36	double precision		
rede_range_37	double precision		
rede_range_38	double precision		
rede_range_39	double precision		
rede_range_40	double precision		
rede_range_41	double precision		
rede_range_42	double precision		
rede_range_43	double precision		
rede_range_44	double precision		
rede_range_45	double precision		
rede_range_46	double precision		
rede_range_47	double precision		
rede_range_48	double precision		
rede_range_49	double precision		
rede_range_50	double precision		
rede_range_51	double precision		
rede_range_52	double precision		
rede_range_53	double precision		
rede_range_54	double precision		
rede_range_55	double precision		
rede_range_56	double precision		
rede_range_57	double precision		
rede_range_58	double precision		
rede_range_59	double precision		
rede_range_60	double precision		
rede_range_61	double precision		
rede_range_62	double precision		
rede_range_63	double precision		
rede_range_64	double precision		
rede_range_65	double precision		
rede_range_66	double precision		

rede_range_67	double precision		
rede_range_68	double precision		
rede_range_69	double precision		
rede_range_70	double precision		
rede_range_71	double precision		
rede_range_72	double precision		
rede_range_73	double precision		
rede_range_74	double precision		
rede_range_75	double precision		
rede_range_76	double precision		
rede_range_77	double precision		
rede_range_78	double precision		
rede_range_79	double precision		
rede_range_80	double precision		
rede_range_81	double precision		
rede_range_82	double precision		
rede_range_83	double precision		
rede_range_84	double precision		
rede_range_85	double precision		
rede_range_86	double precision		
rede_range_87	double precision		
rede_range_88	double precision		
rede_range_89	double precision		
rede_range_90	double precision		
rede_range_91	double precision		
rede_range_92	double precision		
rede_range_93	double precision		
rede_range_94	double precision		
rede_range_95	double precision		
rede_range_96	double precision		
rede_range_97	double precision		
rede_range_98	double precision		
rede_range_99	double precision		
rede_range_100	double precision		

Tabela rede_9

CAMPO	TIPO	TAMANHO	OBS
rede_neuronio	int		
rede_carac	int		
rede_range_0	double precision		
rede_range_1	double precision		
rede_range_2	double precision		
rede_range_3	double precision		
rede_range_4	double precision		
rede_range_5	double precision		
rede_range_6	double precision		
rede_range_7	double precision		
rede_range_8	double precision		
rede_range_9	double precision		
rede_range_10	double precision		
rede_range_11	double precision		
rede_range_12	double precision		
rede_range_13	double precision		
rede_range_14	double precision		
rede_range_15	double precision		
rede_range_16	double precision		
rede_range_17	double precision		
rede_range_18	double precision		
rede_range_19	double precision		
rede_range_20	double precision		
rede_range_21	double precision		
rede_range_22	double precision		
rede_range_23	double precision		
rede_range_24	double precision		
rede_range_25	double precision		
rede_range_26	double precision		
rede_range_27	double precision		
rede_range_28	double precision		
rede_range_29	double precision		
rede_range_30	double precision		

rede_range_31	double precision		
rede_range_32	double precision		
rede_range_33	double precision		
rede_range_34	double precision		
rede_range_35	double precision		
rede_range_36	double precision		
rede_range_37	double precision		
rede_range_38	double precision		
rede_range_39	double precision		
rede_range_40	double precision		
rede_range_41	double precision		
rede_range_42	double precision		
rede_range_43	double precision		
rede_range_44	double precision		
rede_range_45	double precision		
rede_range_46	double precision		
rede_range_47	double precision		
rede_range_48	double precision		
rede_range_49	double precision		
rede_range_50	double precision		
rede_range_51	double precision		
rede_range_52	double precision		
rede_range_53	double precision		
rede_range_54	double precision		
rede_range_55	double precision		
rede_range_56	double precision		
rede_range_57	double precision		
rede_range_58	double precision		
rede_range_59	double precision		
rede_range_60	double precision		
rede_range_61	double precision		
rede_range_62	double precision		
rede_range_63	double precision		
rede_range_64	double precision		
rede_range_65	double precision		
rede_range_66	double precision		

rede_range_67	double precision		
rede_range_68	double precision		
rede_range_69	double precision		
rede_range_70	double precision		
rede_range_71	double precision		
rede_range_72	double precision		
rede_range_73	double precision		
rede_range_74	double precision		
rede_range_75	double precision		
rede_range_76	double precision		
rede_range_77	double precision		
rede_range_78	double precision		
rede_range_79	double precision		
rede_range_80	double precision		
rede_range_81	double precision		
rede_range_82	double precision		
rede_range_83	double precision		
rede_range_84	double precision		
rede_range_85	double precision		
rede_range_86	double precision		
rede_range_87	double precision		
rede_range_88	double precision		
rede_range_89	double precision		
rede_range_90	double precision		
rede_range_91	double precision		
rede_range_92	double precision		
rede_range_93	double precision		
rede_range_94	double precision		
rede_range_95	double precision		
rede_range_96	double precision		
rede_range_97	double precision		
rede_range_98	double precision		
rede_range_99	double precision		
rede_range_100	double precision		

Tabela rede_10

CAMPO	TIPO	TAMANHO	OBS
rede_neuronio	int		
rede_carac	int		
rede_range_0	double precision		
rede_range_1	double precision		
rede_range_2	double precision		
rede_range_3	double precision		
rede_range_4	double precision		
rede_range_5	double precision		
rede_range_6	double precision		
rede_range_7	double precision		
rede_range_8	double precision		
rede_range_9	double precision		
rede_range_10	double precision		
rede_range_11	double precision		
rede_range_12	double precision		
rede_range_13	double precision		
rede_range_14	double precision		
rede_range_15	double precision		
rede_range_16	double precision		
rede_range_17	double precision		
rede_range_18	double precision		
rede_range_19	double precision		
rede_range_20	double precision		
rede_range_21	double precision		
rede_range_22	double precision		
rede_range_23	double precision		
rede_range_24	double precision		
rede_range_25	double precision		
rede_range_26	double precision		
rede_range_27	double precision		
rede_range_28	double precision		

rede_range_29	double precision		
rede_range_30	double precision		
rede_range_31	double precision		
rede_range_32	double precision		
rede_range_33	double precision		
rede_range_34	double precision		
rede_range_35	double precision		
rede_range_36	double precision		
rede_range_37	double precision		
rede_range_38	double precision		
rede_range_39	double precision		
rede_range_40	double precision		
rede_range_41	double precision		
rede_range_42	double precision		
rede_range_43	double precision		
rede_range_44	double precision		
rede_range_45	double precision		
rede_range_46	double precision		
rede_range_47	double precision		
rede_range_48	double precision		
rede_range_49	double precision		
rede_range_50	double precision		
rede_range_51	double precision		
rede_range_52	double precision		
rede_range_53	double precision		
rede_range_54	double precision		
rede_range_55	double precision		
rede_range_56	double precision		
rede_range_57	double precision		
rede_range_58	double precision		
rede_range_59	double precision		
rede_range_60	double precision		
rede_range_61	double precision		
rede_range_62	double precision		
rede_range_63	double precision		
rede_range_64	double precision		

rede_range_65	double precision		
rede_range_66	double precision		
rede_range_67	double precision		
rede_range_68	double precision		
rede_range_69	double precision		
rede_range_70	double precision		
rede_range_71	double precision		
rede_range_72	double precision		
rede_range_73	double precision		
rede_range_74	double precision		
rede_range_75	double precision		
rede_range_76	double precision		
rede_range_77	double precision		
rede_range_78	double precision		
rede_range_79	double precision		
rede_range_80	double precision		
rede_range_81	double precision		
rede_range_82	double precision		
rede_range_83	double precision		
rede_range_84	double precision		
rede_range_85	double precision		
rede_range_86	double precision		
rede_range_87	double precision		
rede_range_88	double precision		
rede_range_89	double precision		
rede_range_90	double precision		
rede_range_91	double precision		
rede_range_92	double precision		
rede_range_93	double precision		
rede_range_94	double precision		
rede_range_95	double precision		
rede_range_96	double precision		
rede_range_97	double precision		
rede_range_98	double precision		
rede_range_99	double precision		
rede_range_100	double precision		

Tabela rede_11

CAMPO	TIPO	TAMANHO	OBS
rede_neuronio	int		
rede_carac	int		
rede_range_0	double precision		
rede_range_1	double precision		
rede_range_2	double precision		
rede_range_3	double precision		
rede_range_4	double precision		
rede_range_5	double precision		
rede_range_6	double precision		
rede_range_7	double precision		
rede_range_8	double precision		
rede_range_9	double precision		
rede_range_10	double precision		
rede_range_11	double precision		
rede_range_12	double precision		
rede_range_13	double precision		
rede_range_14	double precision		
rede_range_15	double precision		
rede_range_16	double precision		
rede_range_17	double precision		
rede_range_18	double precision		
rede_range_19	double precision		
rede_range_20	double precision		
rede_range_21	double precision		
rede_range_22	double precision		
rede_range_23	double precision		
rede_range_24	double precision		
rede_range_25	double precision		
rede_range_26	double precision		
rede_range_27	double precision		
rede_range_28	double precision		
rede_range_29	double precision		

rede_range_30	double precision		
rede_range_31	double precision		
rede_range_32	double precision		
rede_range_33	double precision		
rede_range_34	double precision		
rede_range_35	double precision		
rede_range_36	double precision		
rede_range_37	double precision		
rede_range_38	double precision		
rede_range_39	double precision		
rede_range_40	double precision		
rede_range_41	double precision		
rede_range_42	double precision		
rede_range_43	double precision		
rede_range_44	double precision		
rede_range_45	double precision		
rede_range_46	double precision		
rede_range_47	double precision		
rede_range_48	double precision		
rede_range_49	double precision		
rede_range_50	double precision		
rede_range_51	double precision		
rede_range_52	double precision		
rede_range_53	double precision		
rede_range_54	double precision		
rede_range_55	double precision		
rede_range_56	double precision		
rede_range_57	double precision		
rede_range_58	double precision		
rede_range_59	double precision		
rede_range_60	double precision		
rede_range_61	double precision		
rede_range_62	double precision		
rede_range_63	double precision		
rede_range_64	double precision		
rede_range_65	double precision		

rede_range_66	double precision		
rede_range_67	double precision		
rede_range_68	double precision		
rede_range_69	double precision		
rede_range_70	double precision		
rede_range_71	double precision		
rede_range_72	double precision		
rede_range_73	double precision		
rede_range_74	double precision		
rede_range_75	double precision		
rede_range_76	double precision		
rede_range_77	double precision		
rede_range_78	double precision		
rede_range_79	double precision		
rede_range_80	double precision		
rede_range_81	double precision		
rede_range_82	double precision		
rede_range_83	double precision		
rede_range_84	double precision		
rede_range_85	double precision		
rede_range_86	double precision		
rede_range_87	double precision		
rede_range_88	double precision		
rede_range_89	double precision		
rede_range_90	double precision		
rede_range_91	double precision		
rede_range_92	double precision		
rede_range_93	double precision		
rede_range_94	double precision		
rede_range_95	double precision		
rede_range_96	double precision		
rede_range_97	double precision		
rede_range_98	double precision		
rede_range_99	double precision		
rede_range_100	double precision		

Tabela rede_12

CAMPO	TIPO	TAMANHO	OBS
rede_neuronio	int		
rede_carac	int		
rede_range_0	double precision		
rede_range_1	double precision		
rede_range_2	double precision		
rede_range_3	double precision		
rede_range_4	double precision		
rede_range_5	double precision		
rede_range_6	double precision		
rede_range_7	double precision		
rede_range_8	double precision		
rede_range_9	double precision		
rede_range_10	double precision		
rede_range_11	double precision		
rede_range_12	double precision		
rede_range_13	double precision		
rede_range_14	double precision		
rede_range_15	double precision		
rede_range_16	double precision		
rede_range_17	double precision		
rede_range_18	double precision		
rede_range_19	double precision		
rede_range_20	double precision		
rede_range_21	double precision		
rede_range_22	double precision		
rede_range_23	double precision		
rede_range_24	double precision		
rede_range_25	double precision		
rede_range_26	double precision		
rede_range_27	double precision		
rede_range_28	double precision		
rede_range_29	double precision		

rede_range_30	double precision		
rede_range_31	double precision		
rede_range_32	double precision		
rede_range_33	double precision		
rede_range_34	double precision		
rede_range_35	double precision		
rede_range_36	double precision		
rede_range_37	double precision		
rede_range_38	double precision		
rede_range_39	double precision		
rede_range_40	double precision		
rede_range_41	double precision		
rede_range_42	double precision		
rede_range_43	double precision		
rede_range_44	double precision		
rede_range_45	double precision		
rede_range_46	double precision		
rede_range_47	double precision		
rede_range_48	double precision		
rede_range_49	double precision		
rede_range_50	double precision		
rede_range_51	double precision		
rede_range_52	double precision		
rede_range_53	double precision		
rede_range_54	double precision		
rede_range_55	double precision		
rede_range_56	double precision		
rede_range_57	double precision		
rede_range_58	double precision		
rede_range_59	double precision		
rede_range_60	double precision		
rede_range_61	double precision		
rede_range_62	double precision		
rede_range_63	double precision		
rede_range_64	double precision		
rede_range_65	double precision		

rede_range_66	double precision		
rede_range_67	double precision		
rede_range_68	double precision		
rede_range_69	double precision		
rede_range_70	double precision		
rede_range_71	double precision		
rede_range_72	double precision		
rede_range_73	double precision		
rede_range_74	double precision		
rede_range_75	double precision		
rede_range_76	double precision		
rede_range_77	double precision		
rede_range_78	double precision		
rede_range_79	double precision		
rede_range_80	double precision		
rede_range_81	double precision		
rede_range_82	double precision		
rede_range_83	double precision		
rede_range_84	double precision		
rede_range_85	double precision		
rede_range_86	double precision		
rede_range_87	double precision		
rede_range_88	double precision		
rede_range_89	double precision		
rede_range_90	double precision		
rede_range_91	double precision		
rede_range_92	double precision		
rede_range_93	double precision		
rede_range_94	double precision		
rede_range_95	double precision		
rede_range_96	double precision		
rede_range_97	double precision		
rede_range_98	double precision		
rede_range_99	double precision		
rede_range_100	double precision		

Tabela rede_13

CAMPO	TIPO	TAMANHO	OBS
rede_neuronio	int		
rede_carac	int		
rede_range_0	double precision		
rede_range_1	double precision		
rede_range_2	double precision		
rede_range_3	double precision		
rede_range_4	double precision		
rede_range_5	double precision		
rede_range_6	double precision		
rede_range_7	double precision		
rede_range_8	double precision		
rede_range_9	double precision		
rede_range_10	double precision		
rede_range_11	double precision		
rede_range_12	double precision		
rede_range_13	double precision		
rede_range_14	double precision		
rede_range_15	double precision		
rede_range_16	double precision		
rede_range_17	double precision		
rede_range_18	double precision		
rede_range_19	double precision		
rede_range_20	double precision		
rede_range_21	double precision		
rede_range_22	double precision		
rede_range_23	double precision		
rede_range_24	double precision		
rede_range_25	double precision		
rede_range_26	double precision		
rede_range_27	double precision		
rede_range_28	double precision		
rede_range_29	double precision		

rede_range_30	double precision		
rede_range_31	double precision		
rede_range_32	double precision		
rede_range_33	double precision		
rede_range_34	double precision		
rede_range_35	double precision		
rede_range_36	double precision		
rede_range_37	double precision		
rede_range_38	double precision		
rede_range_39	double precision		
rede_range_40	double precision		
rede_range_41	double precision		
rede_range_42	double precision		
rede_range_43	double precision		
rede_range_44	double precision		
rede_range_45	double precision		
rede_range_46	double precision		
rede_range_47	double precision		
rede_range_48	double precision		
rede_range_49	double precision		
rede_range_50	double precision		
rede_range_51	double precision		
rede_range_52	double precision		
rede_range_53	double precision		
rede_range_54	double precision		
rede_range_55	double precision		
rede_range_56	double precision		
rede_range_57	double precision		
rede_range_58	double precision		
rede_range_59	double precision		
rede_range_60	double precision		
rede_range_61	double precision		
rede_range_62	double precision		
rede_range_63	double precision		
rede_range_64	double precision		
rede_range_65	double precision		

rede_range_66	double precision		
rede_range_67	double precision		
rede_range_68	double precision		
rede_range_69	double precision		
rede_range_70	double precision		
rede_range_71	double precision		
rede_range_72	double precision		
rede_range_73	double precision		
rede_range_74	double precision		
rede_range_75	double precision		
rede_range_76	double precision		
rede_range_77	double precision		
rede_range_78	double precision		
rede_range_79	double precision		
rede_range_80	double precision		
rede_range_81	double precision		
rede_range_82	double precision		
rede_range_83	double precision		
rede_range_84	double precision		
rede_range_85	double precision		
rede_range_86	double precision		
rede_range_87	double precision		
rede_range_88	double precision		
rede_range_89	double precision		
rede_range_90	double precision		
rede_range_91	double precision		
rede_range_92	double precision		
rede_range_93	double precision		
rede_range_94	double precision		
rede_range_95	double precision		
rede_range_96	double precision		
rede_range_97	double precision		
rede_range_98	double precision		
rede_range_99	double precision		
rede_range_100	double precision		

Tabela rede_14

CAMPO	TIPO	TAMANHO	OBS
rede_neuronio	int		
rede_carac	int		
rede_range_0	double precision		
rede_range_1	double precision		
rede_range_2	double precision		
rede_range_3	double precision		
rede_range_4	double precision		
rede_range_5	double precision		
rede_range_6	double precision		
rede_range_7	double precision		
rede_range_8	double precision		
rede_range_9	double precision		
rede_range_10	double precision		
rede_range_11	double precision		
rede_range_12	double precision		
rede_range_13	double precision		
rede_range_14	double precision		
rede_range_15	double precision		
rede_range_16	double precision		
rede_range_17	double precision		
rede_range_18	double precision		
rede_range_19	double precision		
rede_range_20	double precision		
rede_range_21	double precision		
rede_range_22	double precision		
rede_range_23	double precision		
rede_range_24	double precision		
rede_range_25	double precision		
rede_range_26	double precision		
rede_range_27	double precision		
rede_range_28	double precision		
rede_range_29	double precision		

rede_range_30	double precision		
rede_range_31	double precision		
rede_range_32	double precision		
rede_range_33	double precision		
rede_range_34	double precision		
rede_range_35	double precision		
rede_range_36	double precision		
rede_range_37	double precision		
rede_range_38	double precision		
rede_range_39	double precision		
rede_range_40	double precision		
rede_range_41	double precision		
rede_range_42	double precision		
rede_range_43	double precision		
rede_range_44	double precision		
rede_range_45	double precision		
rede_range_46	double precision		
rede_range_47	double precision		
rede_range_48	double precision		
rede_range_49	double precision		
rede_range_50	double precision		
rede_range_51	double precision		
rede_range_52	double precision		
rede_range_53	double precision		
rede_range_54	double precision		
rede_range_55	double precision		
rede_range_56	double precision		
rede_range_57	double precision		
rede_range_58	double precision		
rede_range_59	double precision		
rede_range_60	double precision		
rede_range_61	double precision		
rede_range_62	double precision		
rede_range_63	double precision		
rede_range_64	double precision		
rede_range_65	double precision		

rede_range_66	double precision		
rede_range_67	double precision		
rede_range_68	double precision		
rede_range_69	double precision		
rede_range_70	double precision		
rede_range_71	double precision		
rede_range_72	double precision		
rede_range_73	double precision		
rede_range_74	double precision		
rede_range_75	double precision		
rede_range_76	double precision		
rede_range_77	double precision		
rede_range_78	double precision		
rede_range_79	double precision		
rede_range_80	double precision		
rede_range_81	double precision		
rede_range_82	double precision		
rede_range_83	double precision		
rede_range_84	double precision		
rede_range_85	double precision		
rede_range_86	double precision		
rede_range_87	double precision		
rede_range_88	double precision		
rede_range_89	double precision		
rede_range_90	double precision		
rede_range_91	double precision		
rede_range_92	double precision		
rede_range_93	double precision		
rede_range_94	double precision		
rede_range_95	double precision		
rede_range_96	double precision		
rede_range_97	double precision		
rede_range_98	double precision		
rede_range_99	double precision		
rede_range_100	double precision		

Tabela rede_15

CAMPO	TIPO	TAMANHO	OBS
rede_neuronio	int		
rede_carac	int		
rede_range_0	double precision		
rede_range_1	double precision		
rede_range_2	double precision		
rede_range_3	double precision		
rede_range_4	double precision		
rede_range_5	double precision		
rede_range_6	double precision		
rede_range_7	double precision		
rede_range_8	double precision		
rede_range_9	double precision		
rede_range_10	double precision		
rede_range_11	double precision		
rede_range_12	double precision		
rede_range_13	double precision		
rede_range_14	double precision		
rede_range_15	double precision		
rede_range_16	double precision		
rede_range_17	double precision		
rede_range_18	double precision		
rede_range_19	double precision		
rede_range_20	double precision		
rede_range_21	double precision		
rede_range_22	double precision		
rede_range_23	double precision		
rede_range_24	double precision		
rede_range_25	double precision		
rede_range_26	double precision		
rede_range_27	double precision		
rede_range_28	double precision		
rede_range_29	double precision		

rede_range_30	double precision		
rede_range_31	double precision		
rede_range_32	double precision		
rede_range_33	double precision		
rede_range_34	double precision		
rede_range_35	double precision		
rede_range_36	double precision		
rede_range_37	double precision		
rede_range_38	double precision		
rede_range_39	double precision		
rede_range_40	double precision		
rede_range_41	double precision		
rede_range_42	double precision		
rede_range_43	double precision		
rede_range_44	double precision		
rede_range_45	double precision		
rede_range_46	double precision		
rede_range_47	double precision		
rede_range_48	double precision		
rede_range_49	double precision		
rede_range_50	double precision		
rede_range_51	double precision		
rede_range_52	double precision		
rede_range_53	double precision		
rede_range_54	double precision		
rede_range_55	double precision		
rede_range_56	double precision		
rede_range_57	double precision		
rede_range_58	double precision		
rede_range_59	double precision		
rede_range_60	double precision		
rede_range_61	double precision		
rede_range_62	double precision		
rede_range_63	double precision		
rede_range_64	double precision		
rede_range_65	double precision		

rede_range_66	double precision		
rede_range_67	double precision		
rede_range_68	double precision		
rede_range_69	double precision		
rede_range_70	double precision		
rede_range_71	double precision		
rede_range_72	double precision		
rede_range_73	double precision		
rede_range_74	double precision		
rede_range_75	double precision		
rede_range_76	double precision		
rede_range_77	double precision		
rede_range_78	double precision		
rede_range_79	double precision		
rede_range_80	double precision		
rede_range_81	double precision		
rede_range_82	double precision		
rede_range_83	double precision		
rede_range_84	double precision		
rede_range_85	double precision		
rede_range_86	double precision		
rede_range_87	double precision		
rede_range_88	double precision		
rede_range_89	double precision		
rede_range_90	double precision		
rede_range_91	double precision		
rede_range_92	double precision		
rede_range_93	double precision		
rede_range_94	double precision		
rede_range_95	double precision		
rede_range_96	double precision		
rede_range_97	double precision		
rede_range_98	double precision		
rede_range_99	double precision		
rede_range_100	double precision		

Tabela nx

CAMPO	TIPO	TAMANHO	OBS
rede_codigo	int		
carac1	double precision		
carac2	double precision		
carac3	double precision		
carac4	double precision		
carac5	double precision		
carac6	double precision		
carac7	double precision		

15.4 ANEXO IV
CÓDIGO FONTE DO PROTÓTIPO

CÓDIGO FONTE PROTÓTIPO, FORM FASE_21 (TESTE 2.1 PRODUÇÃO ORAL DE PALAVRAS)

```
unit fase_21;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,  
jpeg, ExtCtrls, StdCtrls, TFlatEditUnit, TFlatGroupBoxUnit, Buttons,  
TFlatSpeedButtonUnit, Grids, TFlatComboBoxUnit, TFlatPanelUnit, MMSystem,  
DB, ZAbstractRODataset, ZAbstractDataset, ZDataset,  
TFlatListBoxUnit, TFlatTabControlUnit, Audio, TeEngine, Series, TeeProcs,  
Chart;
```

```
type
```

```
Tf_fase_21 = class(TForm)  
  pnsuspende: TPanel;  
  FlatSpeedButton3: TFlatSpeedButton;  
  FlatSpeedButton4: TFlatSpeedButton;  
  FlatSpeedButton5: TFlatSpeedButton;  
  FlatSpeedButton1: TFlatSpeedButton;  
  cartelas: TFlatTabControl;  
  Shape7: TShape;  
  Shape4: TShape;  
  Shape6: TShape;  
  Shape5: TShape;  
  shape3: TShape;  
  Shape2: TShape;  
  Shape1: TShape;  
  Label1: TLabel;  
  data: TLabel;  
  aplicador: TLabel;  
  Image6: TImage;  
  Image5: TImage;  
  Image4: TImage;  
  Image3: TImage;  
  Image2: TImage;  
  Image1: TImage;  
  Shape8: TShape;  
  Shape9: TShape;  
  Shape10: TShape;  
  Shape11: TShape;  
  Shape12: TShape;  
  Shape13: TShape;  
  panel_action: TPanel;  
  btaction: TFlatSpeedButton;  
  Timer1: TTimer;  
  Audio1: TAudio;  
  lab1: TLabel;  
  lab6: TLabel;  
  lab5: TLabel;  
  lab4: TLabel;  
  lab3: TLabel;  
  lab2: TLabel;  
  memopd: TMemo;
```

```

procedure FormCreate(Sender: TObject);
procedure Formatar;
procedure Alocar;
procedure Suprimir;
procedure Comprimir;
procedure ExtrairCaract;
procedure TestarRede;
procedure Normaliza;
procedure le_teste(teste: integer);
procedure carrega_rede_banco(teste: integer);
procedure Image1Click(Sender: TObject);
procedure Image2Click(Sender: TObject);
procedure Image3Click(Sender: TObject);
procedure Image4Click(Sender: TObject);
procedure Image5Click(Sender: TObject);
procedure Image6Click(Sender: TObject);
procedure Timer1Timer(Sender: TObject);
procedure troca_imagem_mouse;
procedure limpa_imagens;
procedure inicia_teste(t_fase,t_cartela : integer);
procedure Image1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure Image2MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure Image3MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure Image4MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure Image5MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure Image6MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure suspende_teste;
procedure FlatSpeedButton3Click(Sender: TObject);
procedure FlatSpeedButton4Click(Sender: TObject);
procedure FlatSpeedButton5Click(Sender: TObject);
procedure FlatSpeedButton1Click(Sender: TObject);
procedure FormKeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
procedure FormActivate(Sender: TObject);
procedure gravar_imagem;
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure cartelasTabChanged(Sender: TObject);
procedure cartelasEnter(Sender: TObject);
procedure cartelasKeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
procedure btactionClick(Sender: TObject);
procedure Audio1Record(Sender: TObject; LP, RP: Pointer;
  BufferSize: Word);
procedure inicializa_vetores;
function quadro(num : real):real;
function tests(neu : integer;arg : array of real):real;
function mu_x(x, centro, z : real) : real;

private
  query : TZQuery;

```



```

    { Private declarations }
public
    t_fase,t_cartela : integer;
    { Public declarations }
end;

const
    MAX_CLA = 6;
    MAX_CAR = 7;
    MAX_RAN = 100;
    n_neur = 6;
    n_entr = 7;
    max_range = 100;

var

    n_som: Int64;
    a,b,ini,fin,hh,epc,d,num_carac,n_pdr,classe,cart,indice: integer;
    s : string;
    mat: array [0..2000,0..7] of integer;
    vet: array [0..1000000,1..2] of integer;
    som: array [0..1000000] of integer;
    h : array [0..10000] of integer;
    dif: array [0..10000] of real;
    clean: array [0..7] of real;
    classe_real, classe_fan, classe_certo, classe_erro : array [0..MAX_CLA] of integer;
    bxin, nx: array [0..MAX_CAR] of real;
    rede : array [0..MAX_CLA,0..MAX_CAR,0..MAX_RAN] of real;
    ess,peso_neu : array [0..MAX_CLA] of double;
    mat_carac : array of array [0..MAX_CAR-1] of double;
    eh: boolean;
    media, tempera: real;
    cartela : array [1..15,1..6] of integer;
    vet_carac : array of integer;

    // Variaveis já existentes

    stop : boolean;
    nomes : array [1..6,0..14] of string;
    fase : smallint; // 1 - Apresentação Cartelas 2 - Teste
    imagem : smallint;
    imagem_mouse : smallint;
    n_cartela : smallint;
    str_cartela : string;
    LeftStream, RightStream, str1, str2: TMemoryStream;
    mu_fz : array of array of double;
    s_eng, tr : array of double;
    decr, max_sem, taichi, max_tot, eps : real;
    n : integer;
    m_graf, max_cont : integer;
    n_floats : Longword;
    s_harm : double;
    mostra, stp, omt : char;
    f_fase_21: Tf_fase_21;

```

implementation

uses u_dmconexao, Math, u_cls_historico, u_cls_resultado, u_identificacao;

{SR *.DFM}

```
procedure Tf_fase_21.FormCreate(Sender: TObject);
begin
  query := TZQuery.Create(nil);
  query.Connection := dmconexao.ad;
  Randomize;
  imagem := 0;
  cart := 1;
  indice := 11;
  n_cartela := 1;
  str_cartela := '01';
  Image1.Picture.LoadFromFile('Imagens_tratadas\011p.jpg');
  Image2.Picture.LoadFromFile('Imagens_tratadas\012p.jpg');
  Image3.Picture.LoadFromFile('Imagens_tratadas\013p.jpg');
  Image4.Picture.LoadFromFile('Imagens_tratadas\014p.jpg');
  Image5.Picture.LoadFromFile('Imagens_tratadas\015p.jpg');
  Image6.Picture.LoadFromFile('Imagens_tratadas\016p.jpg');
  f_fase_21.WindowState := wsMaximized;
  dmconexao.abre_cls_historico;
  dmconexao.abre_cls_resultado;
  dmconexao.objhistorico.tabela := '21';
  dmconexao.objresultado.tabela := '21';
  SetLength(mu_fz,8,50000);
  SetLength(s_eng,50000);
  SetLength(tr,10000);
  nomes[1,0] := 'Bode';
  nomes[2,0] := 'Vaca';
  nomes[3,0] := 'Gato';
  nomes[4,0] := 'Gado';
  nomes[5,0] := 'Bote';
  nomes[6,0] := 'Faca';
  nomes[1,1] := 'Calo';
  nomes[2,1] := 'Queixo';
  nomes[3,1] := 'Doce';
  nomes[4,1] := 'Doze';
  nomes[5,1] := 'Galo';
  nomes[6,1] := 'Queijo';
  nomes[1,2] := 'Cama';
  nomes[2,2] := 'Torta';
  nomes[3,2] := 'Saca';
  nomes[4,2] := 'Porta';
  nomes[5,2] := 'Faca';
  nomes[6,2] := 'Cana';
  nomes[1,3] := 'Bola';
  nomes[2,3] := 'Folha';
  nomes[3,3] := 'Escada';
  nomes[4,3] := 'Rolha';
  nomes[5,3] := 'Espada';
  nomes[6,3] := 'Gola';
  nomes[1,4] := 'Porta';
  nomes[2,4] := 'Roupa';
```

nomes[3,4] := 'Velha';
nomes[4,4] := 'Sopa';
nomes[5,4] := 'Porca';
nomes[6,4] := 'Vela';
nomes[1,5] := 'Taco';
nomes[2,5] := 'Caco';
nomes[3,5] := 'Pilha';
nomes[4,5] := 'Filha';
nomes[5,5] := 'Saco';
nomes[6,5] := 'Carro';
nomes[1,6] := 'Dente';
nomes[2,6] := 'Batata';
nomes[3,6] := 'Roda';
nomes[4,6] := 'Rosa';
nomes[5,6] := 'Lente';
nomes[6,6] := 'Barata';
nomes[1,7] := 'Mola';
nomes[2,7] := 'Pena';
nomes[3,7] := 'Casa';
nomes[4,7] := 'Cana';
nomes[5,7] := 'Bola';
nomes[6,7] := 'Pera';
nomes[1,8] := 'Pinha';
nomes[2,8] := 'Dado';
nomes[3,8] := 'Cano';
nomes[4,8] := 'Calo';
nomes[5,8] := 'Pilha';
nomes[6,8] := 'Nado';
nomes[1,9] := 'Mureta';
nomes[2,9] := 'Pelada';
nomes[3,9] := 'Vara';
nomes[4,9] := 'Perada';
nomes[5,9] := 'Muleta';
nomes[6,9] := 'Vala';
nomes[1,10] := 'Resa';
nomes[2,10] := 'Bicho';
nomes[3,10] := 'Pente';
nomes[4,10] := 'Bucho';
nomes[5,10] := 'Ponte';
nomes[6,10] := 'Rosa';
nomes[1,11] := 'Piso';
nomes[2,11] := 'Linha';
nomes[3,11] := 'Murro';
nomes[4,11] := 'Morro';
nomes[5,11] := 'Peso';
nomes[6,11] := 'Lenha';
nomes[1,12] := 'Feira';
nomes[2,12] := 'Vovó';
nomes[3,12] := 'Toca';
nomes[4,12] := 'Vovó';
nomes[5,12] := 'Touca';
nomes[6,12] := 'Ferra';
nomes[1,13] := 'Sola';
nomes[2,13] := 'Mola';
nomes[3,13] := 'Bala';
nomes[4,13] := 'Bola';

```

nomes[5,13] := 'Sala';
nomes[6,13] := 'Mala';
nomes[1,14] := 'Trêns';
nomes[2,14] := 'Concha';
nomes[3,14] := 'Traça';
nomes[4,14] := 'Coxa';
nomes[5,14] := 'Trança';
nomes[6,14] := 'Três';
carrega_rede_banco(cart);
le_teste(cart);
Normaliza();
end;

```

```

procedure Tf_fase_21.Image1Click(Sender: TObject);
begin
if (fase = 2) and (stop) then
begin
Timer1.Interval := 0;
if (n_cartela > 0) and (imagem <> 1) then
dmconexao.objresultado.alterar(n_cartela,imagem,1)
else
if imagem = 1 then
dmconexao.objresultado.alterar(n_cartela,imagem,0);
gravar_imagem;
Timer1.Interval := 6000;
end;
end;
end;

```

```

procedure Tf_fase_21.Image2Click(Sender: TObject);
begin
if (fase = 2) and (stop) then
begin
Timer1.Interval := 0;
if (n_cartela > 0) and (imagem <> 2) then
dmconexao.objresultado.alterar(n_cartela,imagem,2)
else
if imagem = 2 then
dmconexao.objresultado.alterar(n_cartela,imagem,0);
gravar_imagem;
Timer1.Interval := 6000;
end;
end;
end;

```

```

procedure Tf_fase_21.Image3Click(Sender: TObject);
begin
if (fase = 2) and (stop) then
begin
Timer1.Interval := 0;
if (n_cartela > 0) and (imagem <> 3) then
dmconexao.objresultado.alterar(n_cartela,imagem,3)
else
if imagem = 3 then
dmconexao.objresultado.alterar(n_cartela,imagem,0);
gravar_imagem;
Timer1.Interval := 6000;
end;
end;
end;

```

end;

```
procedure Tf_fase_21.Image4Click(Sender: TObject);
begin
if (fase = 2) and (stop) then
begin
Timer1.Interval := 0;
if (n_cartela > 0) and (imagem <> 4) then
dmconexao.objresultado.alterar(n_cartela,imagem,4)
else
if imagem = 4 then
dmconexao.objresultado.alterar(n_cartela,imagem,0);
gravar_imagem;
Timer1.Interval := 6000;
end;
end;
```

```
procedure Tf_fase_21.Image5Click(Sender: TObject);
begin
if (fase = 2) and (stop) then
begin
Timer1.Interval := 0;
if (n_cartela > 0) and (imagem <> 5) then
dmconexao.objresultado.alterar(n_cartela,imagem,5)
else
if imagem = 5 then
dmconexao.objresultado.alterar(n_cartela,imagem,0);
gravar_imagem;
Timer1.Interval := 6000;
end;
end;
```

```
procedure Tf_fase_21.Image6Click(Sender: TObject);
begin
if (fase = 2) and (stop) then
begin
Timer1.Interval := 0;
if (n_cartela > 0) and (imagem <> 6) then
dmconexao.objresultado.alterar(n_cartela,imagem,6)
else
if imagem = 6 then
dmconexao.objresultado.alterar(n_cartela,imagem,0);
gravar_imagem;
Timer1.Interval := 6000;
end;
end;
```

```
procedure Tf_fase_21.Timer1Timer(Sender: TObject);
begin
case imagem of
1 : begin
image1.Picture.LoadFromFile('imagens_tratadas\' + str_cartela + '1p.jpg');
image1.Width := 270;
shape8.Width := 272;
image1.Height := 270;
shape8.Height := 272;
```

```

image1.Left := 62;
shape8.Left := 61;
image1.Top := 56;
shape8.Top := 55;
shape8.Pen.Width := 1;
end;
2 : begin
image2.Picture.LoadFromFile('imagens_tratadas\' + str_cartela + '2p.jpg');
image2.Width := 270;
shape9.Width := 272;
image2.Height := 270;
shape9.Height := 272;
image2.Left := 363;
shape9.Left := 362;
image2.Top := 56;
shape9.Top := 55;
shape9.Pen.Width := 1;
end;
3 : begin
image3.Picture.LoadFromFile('imagens_tratadas\' + str_cartela + '3p.jpg');
image3.Width := 270;
shape10.Width := 272;
image3.Height := 270;
shape10.Height := 272;
image3.Left := 666;
shape10.Left := 665;
image3.Top := 56;
shape10.Top := 55;
shape10.Pen.Width := 1;
end;
4 : begin
image4.Picture.LoadFromFile('imagens_tratadas\' + str_cartela + '4p.jpg');
image4.Width := 270;
shape11.Width := 272;
image4.Height := 270;
shape11.Height := 272;
image4.Left := 62;
shape11.Left := 61;
image4.Top := 374;
shape11.Top := 373;
shape11.Pen.Width := 1;
end;
5 : begin
image5.Picture.LoadFromFile('imagens_tratadas\' + str_cartela + '5p.jpg');
image5.Width := 270;
shape12.Width := 272;
image5.Height := 270;
shape12.Height := 272;
image5.Left := 363;
shape12.Left := 362;
image5.Top := 374;
shape12.Top := 373;
shape12.Pen.Width := 1;
end;
6 : begin
image6.Picture.LoadFromFile('imagens_tratadas\' + str_cartela + '6p.jpg');

```

```

image6.Width := 270;
shape13.Width := 272;
image6.Height := 270;
shape13.Height := 272;
image6.Left := 666;
shape13.Left := 665;
image6.Top := 374;
shape13.Top := 373;
shape13.Pen.Width :=1;
end;
end;
gravar_imagem;
case imagem of
1 : begin
image1.Picture.LoadFromFile('imagens_tratadas'+str_cartela+'1c.jpg');
image1.Width := 308;
shape8.Width := 312;
image1.Height := 308;
shape8.Height := 312;
image1.Left := 43;
shape8.Left := 41;
image1.Top := 37;
shape8.Top := 35;
shape8.Pen.Width :=3;
end;
2 : begin
image2.Picture.LoadFromFile('imagens_tratadas'+str_cartela+'2c.jpg');
image2.Width := 308;
shape9.Width := 312;
image2.Height := 308;
shape9.Height := 312;
image2.Left := 344;
shape9.Left := 342;
image2.Top := 37;
shape9.Top := 35;
shape9.Pen.Width :=3;
end;
3 : begin
image3.Picture.LoadFromFile('imagens_tratadas'+str_cartela+'3c.jpg');
image3.Width := 308;
shape10.Width := 312;
image3.Height := 308;
shape10.Height := 312;
image3.Left := 647;
shape10.Left := 645;
image3.Top := 37;
shape10.Top := 35;
shape10.Pen.Width :=3;
end;
4 : begin
image4.Picture.LoadFromFile('imagens_tratadas'+str_cartela+'4c.jpg');
image4.Width := 308;
shape11.Width := 312;
image4.Height := 308;
shape11.Height := 312;
image4.Left := 43;

```

```

shape11.Left := 41;
image4.Top := 355;
shape11.Top := 353;
shape11.Pen.Width := 3;
end;
5 : begin
image5.Picture.LoadFromFile('imagens_tratadas\' + str_cartela + '5c.jpg');
image5.Width := 308;
shape12.Width := 312;
image5.Height := 308;
shape12.Height := 312;
image5.Left := 344;
shape12.Left := 342;
image5.Top := 355;
shape12.Top := 353;
shape12.Pen.Width := 3;
end;
6 : begin
image6.Picture.LoadFromFile('imagens_tratadas\' + str_cartela + '6c.jpg');
image6.Width := 308;
shape13.Width := 312;
image6.Height := 308;
shape13.Height := 312;
image6.Left := 647;
shape13.Left := 645;
image6.Top := 355;
shape13.Top := 353;
shape13.Pen.Width := 3;
end;
end;
Timer1.Interval := 6000;
end;

procedure Tf_fase_21.troca_imagem_mouse;
begin
if (imagem_mouse <> 0) then
case imagem_mouse of
1 : begin
image1.Picture.LoadFromFile('imagens_tratadas\' + str_cartela + '1p.jpg');
image1.Width := 270;
shape8.Width := 272;
image1.Height := 270;
shape8.Height := 272;
image1.Left := 62;
shape8.Left := 61;
image1.Top := 56;
shape8.Top := 55;
shape8.Pen.Width := 1;
end;
2 : begin
image2.Picture.LoadFromFile('imagens_tratadas\' + str_cartela + '2p.jpg');
image2.Width := 270;
shape9.Width := 272;
image2.Height := 270;
shape9.Height := 272;
image2.Left := 363;

```



```

shape9.Left := 362;
image2.Top := 56;
shape9.Top := 55;
shape9.Pen.Width := 1;
end;
3 : begin
image3.Picture.LoadFromFile('imagens_tratadas\' + str_cartela + '3p.jpg');
image3.Width := 270;
shape10.Width := 272;
image3.Height := 270;
shape10.Height := 272;
image3.Left := 666;
shape10.Left := 665;
image3.Top := 56;
shape10.Top := 55;
shape10.Pen.Width := 1;
end;
4 : begin
image4.Picture.LoadFromFile('imagens_tratadas\' + str_cartela + '4p.jpg');
image4.Width := 270;
shape11.Width := 272;
image4.Height := 270;
shape11.Height := 272;
image4.Left := 62;
shape11.Left := 61;
image4.Top := 374;
shape11.Top := 373;
shape11.Pen.Width := 1;
end;
5 : begin
image5.Picture.LoadFromFile('imagens_tratadas\' + str_cartela + '5p.jpg');
image5.Width := 270;
shape12.Width := 272;
image5.Height := 270;
shape12.Height := 272;
image5.Left := 363;
shape12.Left := 362;
image5.Top := 374;
shape12.Top := 373;
shape12.Pen.Width := 1;
end;
6 : begin
image6.Picture.LoadFromFile('imagens_tratadas\' + str_cartela + '6p.jpg');
image6.Width := 270;
shape13.Width := 272;
image6.Height := 270;
shape13.Height := 272;
image6.Left := 666;
shape13.Left := 665;
image6.Top := 374;
shape13.Top := 373;
shape13.Pen.Width := 1;
end;
end;
end;

```

```

procedure Tf_fase_21.inicia_teste(t_fase, t_cartela: integer);
var i : smallint;
begin
  n_cartela := t_cartela;
  cart := t_cartela;
  cartelas.ActiveTab := t_cartela - 1;
  imagem := 0;
  fase := t_fase;
  if (n_cartela < 10) then
    str_cartela := '0' + inttostr(n_cartela)
  else
    str_cartela := inttostr(n_cartela);
  image1.Picture.LoadFromFile('imagens_tratadas\' + str_cartela + '1p.jpg');
  image2.Picture.LoadFromFile('imagens_tratadas\' + str_cartela + '2p.jpg');
  image3.Picture.LoadFromFile('imagens_tratadas\' + str_cartela + '3p.jpg');
  image4.Picture.LoadFromFile('imagens_tratadas\' + str_cartela + '4p.jpg');
  image5.Picture.LoadFromFile('imagens_tratadas\' + str_cartela + '5p.jpg');
  image6.Picture.LoadFromFile('imagens_tratadas\' + str_cartela + '6p.jpg');
  f_fase_21.Update;
  SndPlaySound(Pchar('sons\21comando1.wav'), SND_ASYNC);
  i := 1;
  while i <= 2 do
    begin
      sleep(3000);
      inc(i);
      f_fase_21.Update;
    end;
    Timer1.Interval := 6000;
  case n_cartela of
    1: imagem := 1;
    2: imagem := 3;
    3: imagem := 4;
    4: imagem := 5;
    5: imagem := 1;
    6: imagem := 3;
    7: imagem := 2;
    8: imagem := 5;
    9: imagem := 4;
    10: imagem := 6;
    11: imagem := 2;
    12: imagem := 1;
    13: imagem := 1;
    14: imagem := 2;
    15: imagem := 5;
  end;
  case imagem of
    1: begin
      image1.Picture.LoadFromFile('imagens_tratadas\' + str_cartela + '1c.jpg');
      image1.Width := 308;
      shape8.Width := 312;
      image1.Height := 308;
      shape8.Height := 312;
      image1.Left := 43;
      shape8.Left := 41;
      image1.Top := 37;
      shape8.Top := 35;
    end;
  end;
end;

```

```

shape8.Pen.Width :=3;
end;
2 : begin
image2.Picture.LoadFromFile('imagens_tratadas\' +str_cartela+'2c.jpg');
image2.Width := 308;
shape9.Width := 312;
image2.Height := 308;
shape9.Height := 312;
image2.Left := 344;
shape9.Left := 342;
image2.Top := 37;
shape9.Top := 35;
shape9.Pen.Width :=3;
end;
3 : begin
image3.Picture.LoadFromFile('imagens_tratadas\' +str_cartela+'3c.jpg');
image3.Width := 308;
shape10.Width := 312;
image3.Height := 308;
shape10.Height := 312;
image3.Left := 647;
shape10.Left := 645;
image3.Top := 37;
shape10.Top := 35;
shape10.Pen.Width :=3;
end;
4 : begin
image4.Picture.LoadFromFile('imagens_tratadas\' +str_cartela+'4c.jpg');
image4.Width := 308;
shape11.Width := 312;
image4.Height := 308;
shape11.Height := 312;
image4.Left := 43;
shape11.Left := 41;
image4.Top := 355;
shape11.Top := 353;
shape11.Pen.Width :=3;
end;
5 : begin
image5.Picture.LoadFromFile('imagens_tratadas\' +str_cartela+'5c.jpg');
image5.Width := 308;
shape12.Width := 312;
image5.Height := 308;
shape12.Height := 312;
image5.Left := 344;
shape12.Left := 342;
image5.Top := 355;
shape12.Top := 353;
shape12.Pen.Width :=3;
end;
6 : begin
image6.Picture.LoadFromFile('imagens_tratadas\' +str_cartela+'6c.jpg');
image6.Width := 308;
shape13.Width := 312;
image6.Height := 308;
shape13.Height := 312;

```

```

image6.Left := 647;
shape13.Left := 645;
image6.Top := 355;
shape13.Top := 353;
shape13.Pen.Width := 3;
end;
end;
SndPlaySound(Pchar('sons\21comando2.wav'),SND_ASYNC);
panel_action.Visible := true;
end;

```

```

procedure Tf_fase_21.Image1MouseMove(Sender: TObject; Shift: TShiftState; X,
Y: Integer);
begin
if (imagem_mouse <> 1) then
if (fase = 2) and (stop) then
begin
Timer1.Interval := 0;
Timer1.Interval := 6000;
troca_imagem_mouse;
imagem_mouse := 1;
image1.Picture.LoadFromFile('imagens_tratadas\' + str_cartela + '1c.jpg');
image1.Width := 308;
shape8.Width := 312;
image1.Height := 308;
shape8.Height := 312;
image1.Left := 43;
shape8.Left := 41;
image1.Top := 37;
shape8.Top := 35;
shape8.Pen.Width := 3;
end;
end;

```

```

procedure Tf_fase_21.Image2MouseMove(Sender: TObject; Shift: TShiftState; X,
Y: Integer);
begin
if (imagem_mouse <> 2) then
if (fase = 2) and (stop) then
begin
Timer1.Interval := 0;
Timer1.Interval := 6000;
troca_imagem_mouse;
imagem_mouse := 2;
image2.Picture.LoadFromFile('imagens_tratadas\' + str_cartela + '2c.jpg');
image2.Width := 308;
shape9.Width := 312;
image2.Height := 308;
shape9.Height := 312;
image2.Left := 344;
shape9.Left := 342;
image2.Top := 37;
shape9.Top := 35;
shape9.Pen.Width := 3;
end;
end;

```

```

procedure Tf_fase_21.Image3MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
if (imagem_mouse <> 3) then
if (fase = 2) and (stop) then
begin
Timer1.Interval := 0;
Timer1.Interval := 6000;
troca_imagem_mouse;
imagem_mouse := 3;
image3.Picture.LoadFromFile('imagens_tratadas\' + str_cartela + '3c.jpg');
image3.Width := 308;
shape10.Width := 312;
image3.Height := 308;
shape10.Height := 312;
image3.Left := 647;
shape10.Left := 645;
image3.Top := 37;
shape10.Top := 35;
shape10.Pen.Width := 3;
end;
end;

```

```

procedure Tf_fase_21.Image4MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
if (imagem_mouse <> 4) then
if (fase = 2) and (stop) then
begin
Timer1.Interval := 0;
Timer1.Interval := 6000;
troca_imagem_mouse;
imagem_mouse := 4;
image4.Picture.LoadFromFile('imagens_tratadas\' + str_cartela + '4c.jpg');
image4.Width := 308;
shape11.Width := 312;
image4.Height := 308;
shape11.Height := 312;
image4.Left := 43;
shape11.Left := 41;
image4.Top := 355;
shape11.Top := 353;
shape11.Pen.Width := 3;
end;
end;

```

```

procedure Tf_fase_21.Image5MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
if (imagem_mouse <> 5) then
if (fase = 2) and (stop) then
begin
Timer1.Interval := 0;
Timer1.Interval := 6000;
troca_imagem_mouse;

```

```

imagem_mouse := 5;
image5.Picture.LoadFromFile('imagens_tratadas'+str_cartela+'5c.jpg');
image5.Width := 308;
shape12.Width := 312;
image5.Height := 308;
shape12.Height := 312;
image5.Left := 344;
shape12.Left := 342;
image5.Top := 355;
shape12.Top := 353;
shape12.Pen.Width :=3;
end;
end;

procedure Tf_fase_21.Image6MouseMove(Sender: TObject; Shift: TShiftState; X,
Y: Integer);
begin
if (imagem_mouse <> 6) then
if (fase = 2) and (stop) then
begin
Timer1.Interval := 0;
Timer1.Interval := 6000;
troca_imagem_mouse;
imagem_mouse := 6;
image6.Picture.LoadFromFile('imagens_tratadas'+str_cartela+'6c.jpg');
image6.Width := 308;
shape13.Width := 312;
image6.Height := 308;
shape13.Height := 312;
image6.Left := 647;
shape13.Left := 645;
image6.Top := 355;
shape13.Top := 353;
shape13.Pen.Width :=3;
end;
end;

procedure Tf_fase_21.suspende_teste;
begin
Tag := Timer1.Interval;
timer1.Interval := 0;
pnsuspende.Visible := true;
stop := false;
end;

procedure Tf_fase_21.FlatSpeedButton3Click(Sender: TObject);
begin
with dmconexao.objhistorico do
if alterar_status(inttostr(fase)+inttostr(n_cartela),1) then
begin
messagedlg('Teste interrompido com sucesso!',mtConfirmation,[mbOk],0);
close;
end;
end;

procedure Tf_fase_21.FlatSpeedButton4Click(Sender: TObject);

```

```

begin
  if MessageDlg('Esta operação não poderá ser desfeita, tem certeza de que deseja
continuar?', mtInformation, [mbOK, mbCancel], 0) = mrOk then
    with dmconexao.objhistorico do
      if alterar_status('000', 0) and dmconexao.objresultado.alterar(0) then
        begin
          messagedlg('Teste interrompido com sucesso!', mtConfirmation, [mbOk], 0);
          close;
        end;
      end;
    end;
end;

```

```

procedure Tf_fase_21.FlatSpeedButton5Click(Sender: TObject);
begin
  pnsuspende.Visible := false;
  timer1.Interval := Tag;
  stop := true;
end;

```

```

procedure Tf_fase_21.FlatSpeedButton1Click(Sender: TObject);
begin
  if MessageDlg('Esta operação não poderá ser desfeita, tem certeza de que deseja
continuar?', mtInformation, [mbOK, mbCancel], 0) = mrOk then
    with dmconexao.objhistorico do
      if alterar_status('000', 3) and dmconexao.objresultado.alterar(3) then
        begin
          messagedlg('Teste interrompido com sucesso!', mtConfirmation, [mbOk], 0);
          close;
        end;
      end;
    end;
end;

```

```

procedure Tf_fase_21.FormKeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
begin
  if (key = 123) then
    suspende_teste;
end;

```

```

procedure Tf_fase_21.FormActivate(Sender: TObject);
begin
  stop := true;
  pnsuspende.Visible := false;
  aplicador.Caption := 'Aplicador: '+dmconexao.login_nome;
  data.Caption := 'Data: '+FormatDateTime('dd/mm/yyyy', now);
  f_fase_21.Update;
  inicia_teste(t_fase, t_cartela);
end;

```

```

procedure Tf_fase_21.gravar_imagem;
var alt_cartela : boolean;
begin
  alt_cartela := false;
  case n_cartela of
    1: case imagem of
        1 : imagem := 5;
        2 : imagem := 3;
        3 : imagem := 4;

```

```

4 : begin
    alt_cartela := true;
    imagem := 3;
    inc(n_cartela);
    end;
5 : imagem := 6;
6 : imagem := 2;
end;
2: case imagem of
  1 : imagem := 5;
  2 : imagem := 6;
  3 : imagem := 4;
  4 : imagem := 1;
  5 : imagem := 2;
  6 : begin
    alt_cartela := true;
    imagem := 4;
    inc(n_cartela);
    end;
end;
3: case imagem of
  1 : imagem := 6;
  2 : imagem := 1;
  3 : begin
    alt_cartela := true;
    imagem := 5;
    inc(n_cartela);
    end;
  4 : imagem := 2;
  5 : imagem := 3;
  6 : imagem := 5;
end;
4: case imagem of
  1 : imagem := 6;
  2 : imagem := 4;
  3 : imagem := 1;
  4 : begin
    alt_cartela := true;
    imagem := 1;
    inc(n_cartela);
    end;
  5 : imagem := 3;
  6 : imagem := 2;
end;
5: case imagem of
  1 : imagem := 5;
  2 : imagem := 6;
  3 : begin
    alt_cartela := true;
    imagem := 3;
    inc(n_cartela);
    end;
  4 : imagem := 2;
  5 : imagem := 4;
  6 : imagem := 3;
end;

```



```

6: case imagem of
  1 : imagem := 5;
  2 : imagem := 6;
  3 : imagem := 4;
  4 : imagem := 1;
  5 : imagem := 2;
  6 : begin
    alt_cartela := true;
    imagem := 2;
    inc(n_cartela);
    end;
end;
7: case imagem of
  1 : imagem := 5;
  2 : imagem := 6;
  3 : imagem := 4;
  4 : imagem := 1;
  5 : begin
    alt_cartela := true;
    imagem := 5;
    inc(n_cartela);
    end;
  6 : imagem := 3;
end;
8: case imagem of
  1 : imagem := 6;
  2 : imagem := 3;
  3 : imagem := 4;
  4 : begin
    alt_cartela := true;
    imagem := 4;
    inc(n_cartela);
    end;
  5 : imagem := 1;
  6 : imagem := 2;
end;
9: case imagem of
  1 : imagem := 2;
  2 : imagem := 6;
  3 : imagem := 5;
  4 : imagem := 3;
  5 : imagem := 1;
  6 : begin
    alt_cartela := true;
    imagem := 6;
    inc(n_cartela);
    end;
end;
10: case imagem of
  1 : imagem := 5;
  2 : imagem := 4;
  3 : imagem := 1;
  4 : begin
    alt_cartela := true;
    imagem := 2;
    inc(n_cartela);

```

```

    end;
    5 : imagem := 2;
    6 : imagem := 3;
end;
11: case imagem of
    1 : imagem := 6;
    2 : imagem := 4;
    3 : imagem := 5;
    4 : imagem := 3;
    5 : imagem := 1;
    6 : begin
        alt_cartela := true;
        imagem := 1;
        inc(n_cartela);
        end;
end;
12: case imagem of
    1 : imagem := 5;
    2 : imagem := 6;
    3 : imagem := 4;
    4 : begin
        alt_cartela := true;
        imagem := 1;
        inc(n_cartela);
        end;
    5 : imagem := 2;
    6 : imagem := 3;
end;
13: case imagem of
    1 : imagem := 6;
    2 : imagem := 4;
    3 : begin
        alt_cartela := true;
        imagem := 2;
        inc(n_cartela);
        end;
    4 : imagem := 5;
    5 : imagem := 3;
    6 : imagem := 2;
end;
14: case imagem of
    1 : imagem := 5;
    2 : imagem := 6;
    3 : imagem := 1;
    4 : imagem := 3;
    5 : begin
        alt_cartela := true;
        imagem := 5;
        inc(n_cartela);
        end;
    6 : imagem := 4;
end;
15: case imagem of
    1 : imagem := 6;
    2 : imagem := 4;
    3 : imagem := 1;

```

```

    4 : begin
        alt_cartela := true;
        inc(n_cartela);
        end;
    5 : imagem := 3;
    6 : imagem := 2;
end;
end;
if (alt_cartela) then
if (n_cartela < 16) then
begin
cartelas.ActiveTab := n_cartela - 1;
if (n_cartela < 10) then
str_cartela := '0' + inttostr(n_cartela)
else
str_cartela := inttostr(n_cartela);
image1.Picture.LoadFromFile('imagens_tratadas\' + str_cartela + '1p.jpg');
image2.Picture.LoadFromFile('imagens_tratadas\' + str_cartela + '2p.jpg');
image3.Picture.LoadFromFile('imagens_tratadas\' + str_cartela + '3p.jpg');
image4.Picture.LoadFromFile('imagens_tratadas\' + str_cartela + '4p.jpg');
image5.Picture.LoadFromFile('imagens_tratadas\' + str_cartela + '5p.jpg');
image6.Picture.LoadFromFile('imagens_tratadas\' + str_cartela + '6p.jpg');
case imagem_mouse of
1: image1.Picture.LoadFromFile('imagens_tratadas\' + str_cartela + '1c.jpg');
2: image2.Picture.LoadFromFile('imagens_tratadas\' + str_cartela + '2c.jpg');
3: image3.Picture.LoadFromFile('imagens_tratadas\' + str_cartela + '3c.jpg');
4: image4.Picture.LoadFromFile('imagens_tratadas\' + str_cartela + '4c.jpg');
5: image5.Picture.LoadFromFile('imagens_tratadas\' + str_cartela + '5c.jpg');
6: image6.Picture.LoadFromFile('imagens_tratadas\' + str_cartela + '6c.jpg');
end;
end
else
if dmconexao.objhistorico.alterar_status('000',2) and dmconexao.objresultado.alterar(2) then
begin
messagedlg('Fim do teste',mtConfirmation,[mbOk],0);
close;
end;
end;
end;
end;

SndPlaySound(Pchar('sons\21comando2.wav'),SND_ASYNC);
end;

procedure Tf_fase_21.FormClose(Sender: TObject; var Action: TCloseAction);
begin
Timer1.Interval := 0;
end;

procedure Tf_fase_21.cartelasTabChanged(Sender: TObject);
begin
Timer1.Tag := Timer1.Interval;
Timer1.Interval := 0;
n_cartela := cartelas.ActiveTab + 1;
cart := cartelas.ActiveTab + 1;
le_teste(cart);
carrega_rede_banco(cart);
if (n_cartela < 10) then
str_cartela := '0' + inttostr(n_cartela)
else

```

```

str_cartela := inttostr(n_cartela);
if fase = 1 then
begin
limpa_imagens;
image2.Picture.LoadFromFile('imagens_tratadas\' + str_cartela + '2p.jpg');
image3.Picture.LoadFromFile('imagens_tratadas\' + str_cartela + '3p.jpg');
image4.Picture.LoadFromFile('imagens_tratadas\' + str_cartela + '4p.jpg');
image5.Picture.LoadFromFile('imagens_tratadas\' + str_cartela + '5p.jpg');
image6.Picture.LoadFromFile('imagens_tratadas\' + str_cartela + '6p.jpg');
image1.Picture.LoadFromFile('imagens_tratadas\' + str_cartela + '1c.jpg');
image1.Width := 308;
shape8.Width := 312;
image1.Height := 308;
shape8.Height := 312;
image1.Left := 43;
shape8.Left := 41;
image1.Top := 37;
shape8.Top := 35;
shape8.Pen.Width := 3;
imagem := 1;
end
else if fase = 2 then
begin
limpa_imagens;
image1.Picture.LoadFromFile('imagens_tratadas\' + str_cartela + '1p.jpg');
image2.Picture.LoadFromFile('imagens_tratadas\' + str_cartela + '2p.jpg');
image3.Picture.LoadFromFile('imagens_tratadas\' + str_cartela + '3p.jpg');
image4.Picture.LoadFromFile('imagens_tratadas\' + str_cartela + '4p.jpg');
image5.Picture.LoadFromFile('imagens_tratadas\' + str_cartela + '5p.jpg');
image6.Picture.LoadFromFile('imagens_tratadas\' + str_cartela + '6p.jpg');
case n_cartela of
1: imagem := 1;
2: imagem := 3;
3: imagem := 4;
4: imagem := 5;
5: imagem := 1;
6: imagem := 3;
7: imagem := 2;
8: imagem := 5;
9: imagem := 4;
10: imagem := 6;
11: imagem := 2;
12: imagem := 1;
13: imagem := 1;
14: imagem := 2;
15: imagem := 5;
end;
end;
SndPlaySound(Pchar('sons\' + str_cartela + inttostr(imagem) + '.wav'), SND_ASYNC);
Timer1.Interval := Timer1.Tag;
end;

procedure Tf_fase_21.limpa_imagens;
begin
case imagem of
2: begin

```

```

image2.Width := 270;
shape9.Width := 272;
image2.Height := 270;
shape9.Height := 272;
image2.Left := 363;
shape9.Left := 362;
image2.Top := 56;
shape9.Top := 55;
shape9.Pen.Width := 1;
end;
3 : begin
image3.Width := 270;
shape10.Width := 272;
image3.Height := 270;
shape10.Height := 272;
image3.Left := 666;
shape10.Left := 665;
image3.Top := 56;
shape10.Top := 55;
shape10.Pen.Width := 1;
end;
4 : begin
image4.Width := 270;
shape11.Width := 272;
image4.Height := 270;
shape11.Height := 272;
image4.Left := 62;
shape11.Left := 61;
image4.Top := 374;
shape11.Top := 373;
shape11.Pen.Width := 1;
end;
5 : begin
image5.Width := 270;
shape12.Width := 272;
image5.Height := 270;
shape12.Height := 272;
image5.Left := 363;
shape12.Left := 362;
image5.Top := 374;
shape12.Top := 373;
shape12.Pen.Width := 1;
end;
6 : begin
image6.Width := 270;
shape13.Width := 272;
image6.Height := 270;
shape13.Height := 272;
image6.Left := 666;
shape13.Left := 665;
image6.Top := 374;
shape13.Top := 373;
shape13.Pen.Width := 1;
end;
end;
end;

```

```

procedure Tf_fase_21.cartelasEnter(Sender: TObject);
begin
f_fase_21.SetFocus;
end;

procedure Tf_fase_21.cartelasKeyDown(Sender: TObject; var Key: Word;
Shift: TShiftState);
begin
if (key = 123) then
suspende_teste;
end;

procedure Tf_fase_21.btactionClick(Sender: TObject);
begin
if btaction.Tag = 0 then
begin
Timer1.Interval := 0;
btaction.Tag := 1;
btaction.Caption := 'Parar Gravação';
if not(Audio1.Recorder.Start) then
MessageDlg(Audio1.ErrorMessage,mtError,[mbOK],0);
if LeftStream <> nil then
begin
LeftStream.Free;
RightStream.Free;
LeftStream := nil;
RightStream:=nil;
Str2.Free;
Str2:= nil;
end;
LeftStream :=TMemoryStream.Create;
RightStream:=TMemoryStream.Create;
end
else
begin
if not(Audio1.Recorder.Stop) then
MessageDlg(Audio1.ErrorMessage,mtError,[mbOK],0);
n_som:=LeftStream.Size;
Alocar();
Suprimir();
Formatar();
Comprimir();
ExtrairCaract();
TestarRede();
Timer1Timer(Sender);
btaction.Tag := 0;
btaction.Caption := 'Iniciar Gravação';
end;
end;

procedure Tf_fase_21.Audio1Record(Sender: TObject; LP, RP: Pointer;
BufferSize: Word);
begin
LeftStream.Write(LP^,BufferSize);
if RP<>nil then

```

```
RightStream.Write(RP^,BufferSize);  
end;
```

```
procedure Tf_fase_21.Alocar;  
var x: integer;  
    dado: byte;  
begin  
    LeftStream.Position:=0;  
    for x=0 to n_som do  
        begin  
            LeftStream.Read(dado, 1);  
            vet[x,1]:=dado;  
            vet[x,2]:=x;  
        end;  
    end;
```

```
procedure Tf_fase_21.Suprimir;  
var x,soma,y,z,med1,med2: integer;  
begin  
    ini:=0;  
    fim:=0;  
    soma:=0;  
    media:=0;  
    for x=0 to n_som do  
        begin  
            soma:=soma+abs(vet[x,1]);  
            media:=soma div n_som;  
        end;  
    for x=0 to n_som do  
        if abs(vet[x,1]) > (media*1.05) then  
            begin  
                med1:=0;  
                med2:=0;  
                for z:=x-10 to x+10 do  
                    med1:=med1+abs(vet[z,1]);  
                for z:=x+10 to x+30 do  
                    med2:=med2+abs(vet[z,1]);  
                if med2 > med1 then  
                    break;  
            end;  
            ini:=x;  
        for x=0 to n_som do  
            if abs(vet[n_som-x,1]) > (media*1.05) then  
                begin  
                    med1:=0;  
                    med2:=0;  
                    for z:=n_som-(x-10) to n_som-(x+10) do  
                        med1:=med1+abs(vet[z,1]);  
                    for z:=n_som-(x+20) to n_som-(x+10) do  
                        med2:=med2+abs(vet[z,1]);  
                    if med2 > med1 then  
                        break;  
                end;  
            fim:=n_som-x;  
            y:=ini;  
            for x=0 to fim-ini do
```

```

begin
  som[x]:=vet[y,1];
  inc(y);
end;
end;

procedure Tf_fase_21.Formatar;
var x,i,j,a,b,ponto,aux: integer;
begin
  a := ini;
  b := fim;
  for i := a to b do
    for j := i+1 to b do
      if vet[i,1] > vet[j,1] then
        begin
          aux := vet[i,1];
          vet[i,1] := vet[j,1];
          vet[j,1] := aux;
          aux := vet[i,2];
          vet[i,2] := vet[j,2];
          vet[j,2] := aux;
        end;
    ponto := Round(b*0.7);
    for x=a to b do
      begin
        if x>=ponto then
          vet[x,1]=1
        else
          vet[x,1]=0;
      end;
    for i := a to b do
      for j := i+1 to b do
        if vet[i,2] > vet[j,2] then
          begin
            aux := vet[i,1];
            vet[i,1] := vet[j,1];
            vet[j,1] := aux;
            aux := vet[i,2];
            vet[i,2] := vet[j,2];
            vet[j,2] := aux;
          end;
    end;
end;

procedure Tf_fase_21.Comprimir;
var x,y,cont: integer;
begin
  cont:=0;
  y:=0;
  for x=ini to fim do
    begin
      if vet[x,1]=vet[x+1,1] then
        inc(cont)
      else
        begin
          cont=cont+1;
          h[y]:=cont;

```



```

    inc(y);
  end;
end;
hh := y;
end;

procedure Tf_fase_21.ExtrairCaract;
var x,y: integer;
begin
  dif[0] := h[0];
  y := 1;
  for x=0 to hh do
    if x < hh-1 then
      begin
        dif[y] := h[x+1]-h[x];
        inc(y);
      end;
    for x=0 to hh-1 do
      dif[x] := round(Ln(dif[x]));
    for x=0 to 7 do
      clean[x] := 0;
    for x=0 to hh do
      begin
        if dif[x] = 1 then
          clean[1] := clean[1]+1;
        if dif[x] = 2 then
          clean[2] := clean[2]+1;
        if dif[x] = 3 then
          clean[3] := clean[3]+1;
        if dif[x] = 4 then
          clean[4] := clean[4]+1;
        if dif[x] = 5 then
          clean[5] := clean[5]+1;
        if dif[x] = 6 then
          clean[6] := clean[6]+1;
        if dif[x] = 7 then
          clean[7] := clean[7]+1;
      end;
    end;
end;

```

```

procedure Tf_fase_21.TestarRede;
var
  ma, mb, prop, errados, maior, maior2, total, media, aux, aux2 : real;
  a, b, q : array [0..MAX_CLA] of real;
  e, xin, bxin : array [0..MAX_CAR] of real;
  i, j, certos, certost, certosm, errosm, errost, result : integer;
  classes: array [0..5,0..1] of real;
begin
  certost := 0;
  certosm := 0;
  errosm := 0;
  errost := 0;
  certos := 0;
  total := 0;
  inc(epc);
  i := 0;

```

```

while (i < n_neur) do
begin
  classe_real [i] := 0;
  classe_fan [i] := 0;
  classe_certo[i] := 0;
  classe_erro [i] := 0;
  inc(i);
end;
for i := 0 to 6 do
  e[i] := clean[i+1];
  i := 0;
  while (i < n_entr) do
  begin
    xin[i] := (e[i]/nx[i]);
    inc(i);
  end;
  i := 0;
  while (i < n_entr) do
  begin
    if (xin[i] > 1) then
      xin[i] := 1.;
    inc(i);
  end;
  i := 0;
  while (i < n_entr) do
  begin
    bxin[i] := sqrt(quadro(xin[i]));
    inc(i);
  end;
  i := 0;
  while (i < n_neur) do
  begin
    a[i] := tests(i,bxin);
    ma := ma + a[i];
    mb := mb + b[i];
    inc(i);
  end;
  i := 0;
  while (i < n_neur) do
  begin
    q[i] := a[i];
    classes[i,0] := q[i];
    classes[i,1] := i;
    if (q[i] > q[trunc(maior)]) then
    begin
      maior2 := maior;
      maior := i;
    end;
    inc(i);
  end;
  if (q[trunc(maior)] <> 0) then
    prop := (q[trunc(maior)]-q[trunc(maior2)])/q[trunc(maior)]
  else
    prop := 1.;
  total := total + 1.;
  for i := 0 to 5 do

```

```

for j := i+1 to 5 do
  if classes[i,0] > classes[j,0] then
    begin
      aux := classes[i,0];
      classes[i,0] := classes[j,0];
      classes[j,0] := aux;
      aux2 := classes[i,1];
      classes[i,1] := classes[j,1];
      classes[j,1] := aux2;
    end;
  classe := round(maior+1);
  case cart of
  1:
  case classe of
  1: result := 1;
  2: result := 5;
  3: result := 3;
  4: result := 4;
  5: result := 2;
  6: result := 6;
  end;
  2:
  case classe of
  1: result := 7;
  2: result := 11;
  3: result := 9;
  4: result := 10;
  5: result := 8;
  6: result := 12;
  end;
  3:
  case classe of
  1: result := 13;
  2: result := 17;
  3: result := 16;
  4: result := 14;
  5: result := 6;
  6: result := 15;
  end;
  4:
  case classe of
  1: result := 22;
  2: result := 20;
  3: result := 18;
  4: result := 23;
  5: result := 19;
  6: result := 21;
  end;
  5:
  case classe of
  1: result := 16;
  2: result := 27;
  3: result := 26;
  4: result := 24;
  5: result := 28;
  6: result := 25;

```

```
end;
6:
case classe of
  1: result := 31;
  2: result := 32;
  3: result := 29;
  4: result := 33;
  5: result := 30;
  6: result := 34;
end;
7:
case classe of
  1: result := 36;
  2: result := 40;
  3: result := 37;
  4: result := 38;
  5: result := 35;
  6: result := 39;
end;
8:
case classe of
  1: result := 18;
  2: result := 41;
  3: result := 44;
  4: result := 42;
  5: result := 43;
  6: result := 17;
end;
9:
case classe of
  1: result := 7;
  2: result := 47;
  3: result := 31;
  4: result := 45;
  5: result := 46;
  6: result := 48;
end;
10:
case classe of
  1: result := 54;
  2: result := 51;
  3: result := 53;
  4: result := 49;
  5: result := 50;
  6: result := 52;
end;
11:
case classe of
  1: result := 56;
  2: result := 58;
  3: result := 57;
  4: result := 59;
  5: result := 55;
  6: result := 38;
end;
12:
```

```

case classe of
  1: result := 60;
  2: result := 64;
  3: result := 61;
  4: result := 65;
  5: result := 62;
  6: result := 63;
end;
13:
case classe of
  1: result := 66;
  2: result := 71;
  3: result := 67;
  4: result := 69;
  5: result := 70;
  6: result := 68;
end;
14:
case classe of
  1: result := 41;
  2: result := 75;
  3: result := 18;
  4: result := 73;
  5: result := 72;
  6: result := 74;
end;
15:
case classe of
  1: result := 80;
  2: result := 78;
  3: result := 76;
  4: result := 81;
  5: result := 77;
  6: result := 79;
end;
end;
with dmconexao.query_caracteristica do
begin
{ Close;
  SQL.Clear;
  SQL.Add('update teste_21 set t_21_0'+inttostr(indice)+'_item_desv = where t_21_historic = (select
max(t_21_historic) from teste_21)');
  ExecSQL; }
  Close;
  SQL.Clear;
  SQL.Add('update teste_21 set t_21_0'+inttostr(indice)+'_confirm = '+inttostr(result)+' where
t_21_historic = (select max(t_21_historic) from teste_21)');
  ExecSQL;
end;
inc(indice);
end;

procedure Tf_fase_21.inicializa_vetores;
var
  r : real;
  i, j, k : integer;

```

```

begin
i := 0;
while (i < n_neur) do
begin
j := 0;
while (j < n_entr) do
begin
k := 0;
while (k < max_range) do
begin
r := random(1000)/1000.;
rede[i,j,k] := r;
rede[i,j,max_range] := rede[i,j,max_range] + r;
inc(k);
end;
inc(j);
end;
inc(i);
end;
i := 0;
while (i < n_neur) do
begin
peso_neu [i] := 500.;
inc(i);
end;
end;

function Tf_fase_21.quadro(num: real): real;
begin
result := num*num;
end;

function Tf_fase_21.tests(neu : integer;arg : array of real):real;
var
pi : Extended;
i, j, iip : integer;
idx, sidx, u, s : real;
x : array [0..MAX_CAR] of real;
begin
pi := 1.;
i := 0;
while (i < n_entr) do
begin
x[i] := (max_range-1.0) * arg[i];
inc(i);
end;
j := 0;
while (j < n_entr) do
begin
sidx := 0.;
i := -d;
while (i <= d) do
begin
iip := trunc(x[j]) + i;
if (iip < 0) then
iip := 0;

```

```

        if (iip > max_range-1) then
iip := max_range-1;
        idx := mu_x(iip,x[j],d);
        sidx := sidx + idx;
inc(i);
end;
i := -d;
while (i <= d) do
begin
        iip := trunc(x[j]) + i;
        if (iip < 0) then
iip := 0;
        if (iip > max_range-1) then
iip := max_range-1;
        idx := mu_x(iip,x[j],d);
        u := rede[neu][j][iip];
        s := rede[neu][j][max_range];
        if (s <> 0) then
                pi := pi * (1.-sqrt(idx*u/(s*sidx)));
inc(i);
end;
inc(j);
end;
result := (1 - power(pi, (1 / n_entr)));
end;

function Tf_fase_21.mu_x(x, centro, z: real): real;
var
d1 : real;
begin
d1 := z + 1;
if (x < centro) then
        result := (1./d1)*(x+d1-centro)
else
        result := (1./d1)*(-x+d1+centro);
end;

procedure Tf_fase_21.carrega_rede_banco(teste: integer);
var i,j,k : integer;
begin
with dmconexao.query_caracteristica do
begin
Close;
SQL.Clear;
SQL.Add('Select rede_'+inttostr(teste)+' from redes');
Open;
if (Fields[0].AsInteger = 1) then
begin
Close;
SQL.Clear;
SQL.Add('select * from nx where rede_codigo = '+inttostr(teste));
open;
for i := 0 to n_entr-1 do
        nx[i] := Fields[i+1].Value;
Close;
SQL.Clear;

```

```

SQL.Add('select * from rede_'+inttostr(teste));
open;
while not(Eof) do
begin
i := Fields[0].Value;
j := Fields[1].Value;
For k := 0 to max_range do
rede[i][j][k] := Fields[k+2].Value;
next;
end;
end;
end;
end;

```

```

procedure Tf_fase_21.le_teste(teste: integer);
var i,j,num : integer;
aux,aux2 : array [0..6000] of integer;
test : TextFile;
begin
AssignFile(test,'teste_ts.dat');
rewrite(test);
with dmconexao.query_caracteristica do
begin
Close;
SQL.Clear;
SQL.Add('Select carac_1,carac_2,carac_3,carac_4,carac_5,carac_6,carac_7,pl_codigo from
carac_'+inttostr(teste)+' --where pl_codigo = :p1 or pl_codigo = :p2 or pl_codigo = :p3 or pl_codigo =
:p4 or pl_codigo = :p5 or pl_codigo = :p6 order by pl_codigo');
Open;
SetLength(mat_carac,RecordCount);
SetLength(vet_carac,RecordCount);
num_carac := RecordCount;
for i := 0 to RecordCount-1 do
begin
for j := 0 to 6 do
mat_carac[i,j] := Fields[j].AsFloat;
vet_carac[i] := Fields[j].AsInteger;
next;
end;
for i := 0 to RecordCount-1 do
begin
for j := 0 to 6 do
write(test,mat_carac[i,j]);
writeln(test,strtofloat(inttostr(vet_carac[i])));
end;
end;
CloseFile(test);
end;

```

```

procedure Tf_fase_21.Normaliza;
var
e, se : array [0..MAX_CAR] of real;
i, j, k : integer;
begin
i := 0;
while (i < n_neur) do

```



```

begin
  ess[i] := 0;
  inc(i);
end;
i := 0;
while (i < n_entr) do
begin
  se[i] := 0;
  inc(i);
end;
k := 0;
num_carac := 7;
while (k < num_carac) do
begin
  j := 0;
  while (j < n_entr) do
begin
  e[j] := mat_carac[k,j];
  inc(j);
end;
inc(k);
j := 0;
while (j < n_entr) do
begin
  if (se[j] < e[j]) then
    se[j] := e[j];
  inc(j);
end;
end;
nx[n_entr] := 0;
i := 0;
while (i < n_entr) do
begin
  nx[n_entr] := nx[n_entr] + nx[i];
  inc(i);
end;
end;

end.

```

15.5 ANEXO V

CÓDIGO FONTE DO LABORATÓRIO DE RECONHECIMENTO

LABORATÓRIO DE RECONHECIMENTO DE PALAVRAS

PROGRAMA DESENVOLVIDO PELA EQUIPE:

DARUS CORDEIRO PASTORELLO
DANIEL VICTOR MONTORIL NEGRÃO
RONALDO ROBERTO DA SILVA
VAGNER RODRIGUES

ORIENTADOR:

PROF. MAURO JOSÉ BELLI
CO-ORIENTADOR
PROF.ROBERTO TADEU RAITTZ

TECNOLOGIA EM INFORMÁTICA - 2004

Tela Principal

```
unit uPrincipal;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, StdCtrls, ExtCtrls;
```

```
type
```

```
Tfrm_Principal = class(TForm)
```

```
Button1: TButton;
```

```
Button2: TButton;
```

```
Button3: TButton;
```

```
Button4: TButton;
```

```
Label1: TLabel;
```

```
ComboBox1: TComboBox;
```

```
Bevel1: TBevel;
```

```
Bevel2: TBevel;
```

```
Button5: TButton;
```

```
Bevel3: TBevel;
```

```
Button6: TButton;
```

```
Button7: TButton;
```

```
Bevel4: TBevel;
```

```
Bevel5: TBevel;
```

```
procedure Button1Click(Sender: TObject);
```

```
procedure Button2Click(Sender: TObject);
```

```
procedure Button3Click(Sender: TObject);
```

```
procedure Button4Click(Sender: TObject);
```

```
procedure Button5Click(Sender: TObject);
```

```
procedure Button6Click(Sender: TObject);
```

```
procedure Button7Click(Sender: TObject);
```

```
private
```

```
{ Private declarations }
```

```
public
```

```
{ Public declarations }
```

```
end;
```

```
var
```

```
frm_Principal: Tfrm_Principal;
```

implementation

uses uRede, uRefinador, uGravarPadroes, uGravarRede, uContador;

{\$R *.dfm}

```
procedure Tfrm_Principal.Button1Click(Sender: TObject);
begin
  if ComboBox1.ItemIndex <> -1 then
    frm_Redde.ShowModal;
end;
```

```
procedure Tfrm_Principal.Button2Click(Sender: TObject);
begin
  frm_Refinador.ShowModal;
end;
```

```
procedure Tfrm_Principal.Button3Click(Sender: TObject);
begin
  frm_GravarPadroes.ShowModal;
end;
```

```
procedure Tfrm_Principal.Button4Click(Sender: TObject);
begin
  frm_GravarRede.ShowModal;
end;
```

```
procedure Tfrm_Principal.Button5Click(Sender: TObject);
begin
  frm_Contador.ShowModal;
end;
```

```
procedure Tfrm_Principal.Button6Click(Sender: TObject);
begin
  Application.Terminate;
end;
```

```
procedure Tfrm_Principal.Button7Click(Sender: TObject);
begin
  WinExec('LabFan.exe', sw_show);
end;
```

end.

Tela de Captura da Voz e Teste Online da Rede FAN

unit uRede;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, Audio, ExtCtrls, TeeProcs, TeEngine, Chart, Series,
Buttons;

type

```
Tfrm_Redde = class(TForm)
  Audio1: TAudio;
  botao_gravar: TBitBtn;
  botao_parar: TBitBtn;
  Chart1: TChart;
  Memo: TMemo;
  Series1: TFastLineSeries;
  btn_sair: TBitBtn;
  Chart2: TChart;
  lb2: TListBox;
  Series2: TFastLineSeries;
  Memo1: TMemo;
  l5: TLabel;
  memopd: TMemo;
  l4: TLabel;
  l3: TLabel;
  l2: TLabel;
  l1: TLabel;
  btn_certo: TButton;
  btn_errado: TButton;
  Chart3: TChart;
  Series3: TFastLineSeries;
  btn_desfazer: TButton;
  Bevel1: TBevel;
  procedure Audio1Record(Sender: TObject; LP, RP: Pointer;
    BufferSize: Word);
  procedure botao_gravarClick(Sender: TObject);
  procedure botao_pararClick(Sender: TObject);
  procedure btn_sairClick(Sender: TObject);
  procedure Suprimir();
  procedure Formatar();
  procedure Comprimir();
  procedure Alocar();
  procedure ExtrairCaract();
  procedure LerPadroes();
  procedure CarregaRede();
  procedure SalvarRede();
  procedure TestarRede();
  procedure Normaliza();
  procedure store(neu : integer);
  procedure unstore(neu : integer);
  function quadro(num : real):real;
  function tests(neu : integer;arg : array of real):real;
  function mu_x(x, centro, z : real):real;
  procedure btn_certoClick(Sender: TObject);
```

```

procedure btn_erradoClick(Sender: TObject);
procedure FormShow(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure btn_desfazerClick(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

const
  MAX_CLA = 6;
  MAX_CAR = 7;
  MAX_RAN = 100;
  n_neur = 6;
  n_entr = 7;
  max_range = 100;

var
  frm_Rede: Tfrm_Rede;
  LeftStream, RightStream, ch: TMemoryStream;
  n: Int64;
  a,b,ini,fim,hh,epc,d,num_carac,n_pdr,classe: integer;
  cl1,cl2,cl3,cl4,cl5,cl6: string;
  s,padroes,rededan : string;
  mat: array [0..2000,0..7] of integer;
  vet: array [0..1000000,1..2] of integer;
  som: array [0..1000000] of integer;
  h : array [0..10000] of integer;
  dif: array [0..10000] of real;
  clean: array [0..7] of real;
  classe_real, classe_fan, classe_certo, classe_erro : array [0..MAX_CLA] of integer;
  bxin, nx : array [0..MAX_CAR] of real;
  rede : array [0..MAX_CLA,0..MAX_CAR,0..MAX_RAN] of real;
  ess : array [0..MAX_CLA] of double;
  mat_carac : array of array [0..MAX_CAR-1] of double;
  eh: boolean;
  media, tempera: real;

implementation

uses Math, uPalavra, uPrincipal;

{$R *.dfm}

// Comando para efetuar a gravação da voz utilizando o TAudio
procedure Tfrm_Rede.Audio1Record(Sender: TObject; LP, RP: Pointer;
  BufferSize: Word);
begin
  LeftStream.Write(LP^,BufferSize);
  if RP<>nil then
    RightStream.Write(RP^,BufferSize);
end;

// Inicia a gravação da voz e limpa os objetos do Form (gráficos, labels)
// e zera o vetor de memória do som anteriormente gravado

```

```

procedure Tfrm_Redde.botao_gravarClick(Sender: TObject);
begin
  if not(Audio1.Recorder.Start) then
    Memo.Lines.Add(Audio1.ErrorMessage);
  botao_gravar.Enabled := false;
  botao_gravar.Default := false;
  botao_gravar.Visible := false;
  btn_certo.Enabled := false;
  btn_errado.Enabled := false;
  memo1.Clear;
  Series1.Clear;
  Series2.Clear;
  Series3.Clear;
  if Chart1.Title.Text.Count > 1 then
    Chart1.Title.Text.Delete(1);
  if Chart2.Title.Text.Count > 1 then
    Chart2.Title.Text.Delete(1);
  if LeftStream <> nil then
    begin
      LeftStream.Free;
      RightStream.Free;
      LeftStream := nil;
      RightStream:=nil;
    end;
  LeftStream := TMemoryStream.Create;
  RightStream:=TMemoryStream.Create;
  botao_parar.Visible := true;
  botao_parar.Enabled := true;
  botao_parar.Default := true;
  I1.Caption := "";
  I2.Caption := "";
  I3.Caption := "";
  I4.Caption := "";
  I5.Caption := "";
end;

```

// Para a gravação e chama as funções para efetuar as etapas
// do reconhecimento

```

procedure Tfrm_Redde.botao_pararClick(Sender: TObject);
begin
  if not(Audio1.Recorder.Stop) then
    Memo.Lines.Add(Audio1.ErrorMessage);
  botao_parar.Enabled := false;
  botao_parar.Default := false;
  botao_parar.Visible := false;
  botao_gravar.Default := true;
  botao_gravar.Visible := true;
  n := LeftStream.Size;
  Alocar();
  Suprimir();
  Formatar();
  Comprimir();
  ExtrairCaract();
  TestarRede();
  botao_gravar.Enabled := true;
  btn_certo.Enabled := true;

```

```
btn_errado.Enabled := true;
end;
```

```
// Aloca os valores gravados (amplitudes) para uma matriz
// de números inteiros (numa linha da matriz são as amplitudes
// na outra os índices originais) e plota o sinal original no gráfico
procedure Tfrm_Redde.Alocar;
```

```
var x: integer;
    dado: byte;
begin
    LeftStream.Position:=0;
    for x=0 to n do
        begin
            LeftStream.Read(dado, 1);
            Series1.AddXY(x, dado,"");
            vet[x,1]:=dado;
            vet[x,2]:=x;
        end;
    Chart1.Title.Text.Add(inttostr(n));
end;
```

```
// Suprime o sinal, marcando os pontos de início e final da palavra
// e plota o sinal suprimido no gráfico
```

```
procedure Tfrm_Redde.Suprimir;
var x,soma,y,z,med1,med2: integer;
begin
    ini:=0;
    fim:=0;
    soma:=0;
    media:=0;
    for x=0 to n do
        begin
            soma:=soma+abs(vet[x,1]);
            media:=soma div n;
        end;
    for x=0 to n do
        if abs(vet[x,1]) > (media*1.05) then
            begin
                med1:=0;
                med2:=0;
                for z=x-10 to x+10 do
                    med1:=med1+abs(vet[z,1]);
                for z=x+10 to x+30 do
                    med2:=med2+abs(vet[z,1]);
                if med2 > med1 then
                    break;
            end;
    ini:=x;
    for x=0 to n do
        if abs(vet[n-x,1]) > (media*1.05) then
            begin
                med1:=0;
                med2:=0;
                for z:=n-(x-10) to n-(x+10) do
                    med1:=med1+abs(vet[z,1]);
                for z:=n-(x+20) to n-(x+10) do
```



```

    med2:=med2+abs(vet[z,1]);
    if med2 > med1 then
        break;
    end;
    fim:=n-x;
    y:=ini;
    for x=0 to fim-ini do
        begin
            som[x]:=vet[y,1];
            inc(y);
        end;
        chart2.Title.Text.Add(inttostr(fim-ini));
        for x=ini to fim do
            Series2.AddXY(x-ini,vet[x,1]);
        end;
end;

```

// Ordena o sinal e transforma os 70% maiores valores em 1
// e os 30% menores em 0, em seguida passa os valores para
// sua posição original

```

procedure Tfrm_Redde.Formatar;

```

```

var x,i,j,a,b,ponto,aux: integer;

```

```

begin

```

```

    a := ini;

```

```

    b := fim;

```

```

    for i := a to b do

```

```

        for j := i+1 to b do

```

```

            if vet[i,1] > vet[j,1] then

```

```

                begin

```

```

                    aux := vet[i,1];

```

```

                    vet[i,1] := vet[j,1];

```

```

                    vet[j,1] := aux;

```

```

                    aux := vet[i,2];

```

```

                    vet[i,2] := vet[j,2];

```

```

                    vet[j,2] := aux;

```

```

                end;

```

```

            ponto := Round(b*0.7);

```

```

            for x=a to b do

```

```

                begin

```

```

                    if x>=ponto then

```

```

                        vet[x,1]:=1

```

```

                    else

```

```

                        vet[x,1]:=0;

```

```

                end;

```

```

            for i := a to b do

```

```

                for j := i+1 to b do

```

```

                    if vet[i,2] > vet[j,2] then

```

```

                        begin

```

```

                            aux := vet[i,1];

```

```

                            vet[i,1] := vet[j,1];

```

```

                            vet[j,1] := aux;

```

```

                            aux := vet[i,2];

```

```

                            vet[i,2] := vet[j,2];

```

```

                            vet[j,2] := aux;

```

```

                        end;

```

```

            end;

```

```

end;

```

```

// Marca os pontos de mudança
procedure Tfrm_Rede.Comprimir;
var x,y,cont: integer;
begin
  cont:=0;
  y:=0;
  lb2.Clear;
  for x=ini to fim do
  begin
    if vet[x,1]=vet[x+1,1] then
      inc(cont)
    else
      begin
        cont:=cont+1;
        h[y]:=cont;
        lb2.Items.add(inttostr(h[y]));
        inc(y);
      end;
  end;
  hh := y;
end;

```

```

// Cria o vetor diferença e extrai as características (contagem
// dos resultados do logaritmo neperiano de cada posição do
// vetor diferença) e plota as características no gráfico
procedure Tfrm_Rede.ExtrairCaract;
var x,y: integer;
begin
  dif[0] := h[0];
  y := 1;
  for x=0 to hh do
    if x < hh-1 then
      begin
        dif[y] := h[x+1]-h[x];
        inc(y);
      end;
  for x=0 to hh-1 do
    dif[x] := round(Ln(dif[x]));
  for x=0 to 7 do
    clean[x] := 0;
  for x=0 to hh do
  begin
    if dif[x] = 1 then
      clean[1] := clean[1]+1;
    if dif[x] = 2 then
      clean[2] := clean[2]+1;
    if dif[x] = 3 then
      clean[3] := clean[3]+1;
    if dif[x] = 4 then
      clean[4] := clean[4]+1;
    if dif[x] = 5 then
      clean[5] := clean[5]+1;
    if dif[x] = 6 then
      clean[6] := clean[6]+1;
    if dif[x] = 7 then
      clean[7] := clean[7]+1;
  end;

```

```

end;
for x=1 to 7 do
begin
memo1.Lines[0] := memo1.Lines[0]+floattostr(clean[x])+' ';
Series3.AddXY(x,clean[x],"");
end;
end;

```

```

// Testa a rede FAN com as nova características extraídas.
// Essa e outras funções referente à rede FAN não foram
// programadas pela equipe, foram apenas modificadas
// para se adequar ao nosso programa

```

```

procedure Tfrm_Rede.TestarRede;

```

```

var

```

```

ma, mb, prop, errados, maior, maior2, total, media, aux, aux2 : real;

```

```

a, b, q : array [0..MAX_CLA] of real;

```

```

e, xin, bxin : array [0..MAX_CAR] of real;

```

```

i, j, certos, certost, certosm, errosm, errost : integer;

```

```

classes: array [0..5,0..1] of real;

```

```

begin

```

```

certost := 0;

```

```

certosm := 0;

```

```

errosm := 0;

```

```

errost := 0;

```

```

certos := 0;

```

```

total := 0;

```

```

inc(epc);

```

```

i := 0;

```

```

while (i < n_neur) do

```

```

begin

```

```

classe_real [i] := 0;

```

```

classe_fan [i] := 0;

```

```

classe_certo[i] := 0;

```

```

classe_erro [i] := 0;

```

```

inc(i);

```

```

end;

```

```

for i := 0 to 6 do

```

```

e[i] := clean[i+1];

```

```

i := 0;

```

```

while (i < n_entr) do

```

```

begin

```

```

xin[i] := (e[i]/nx[i]);

```

```

inc(i);

```

```

end;

```

```

i := 0;

```

```

while (i < n_entr) do

```

```

begin

```

```

if (xin[i] > 1) then

```

```

xin[i] := 1.;

```

```

inc(i);

```

```

end;

```

```

i := 0;

```

```

while (i < n_entr) do

```

```

begin

```

```

bxin[i] := sqrt(quadro(xin[i]));

```

```

inc(i);

```

```

end;
i := 0;
while (i < n_neur) do
begin
  a[i] := tests(i,bxin);
  ma := ma + a[i];
  mb := mb + b[i];
  inc(i);
end;
i := 0;
while (i < n_neur) do
begin
  q[i] := a[i];
  classes[i,0] := q[i];
  classes[i,1] := i;
  if (q[i] > q[trunc(maior)]) then
  begin
    maior2 := maior;
    maior := i;
  end;
  inc(i);
end;
if (q[trunc(maior)] <> 0) then
  prop := (q[trunc(maior)]-q[trunc(maior2)])/q[trunc(maior)]
else
  prop := 1.;
total := total + 1.;
for i := 0 to 5 do
  for j := i+1 to 5 do
    if classes[i,0] > classes[j,0] then
    begin
      aux := classes[i,0];
      classes[i,0] := classes[j,0];
      classes[j,0] := aux;
      aux2 := classes[i,1];
      classes[i,1] := classes[j,1];
      classes[j,1] := aux2;
    end;
  classe := round(maior+1);
  case round(classes[1,1]+1) of
    1: I5.Caption := 'BODE';
    2: I5.Caption := 'BOTE';
    3: I5.Caption := 'GATO';
    4: I5.Caption := 'GADO';
    5: I5.Caption := 'VACA';
    6: I5.Caption := 'FACA';
  end;
  case round(classes[2,1]+1) of
    1: I4.Caption := 'BODE';
    2: I4.Caption := 'BOTE';
    3: I4.Caption := 'GATO';
    4: I4.Caption := 'GADO';
    5: I4.Caption := 'VACA';
    6: I4.Caption := 'FACA';
  end;
  case round(classes[3,1]+1) of

```

```

1: l3.Caption := 'BODE';
2: l3.Caption := 'BOTE';
3: l3.Caption := 'GATO';
4: l3.Caption := 'GADO';
5: l3.Caption := 'VACA';
6: l3.Caption := 'FACA';
end;
case round(classes[4,1]+1) of
1: l2.Caption := 'BODE';
2: l2.Caption := 'BOTE';
3: l2.Caption := 'GATO';
4: l2.Caption := 'GADO';
5: l2.Caption := 'VACA';
6: l2.Caption := 'FACA';
end;
case round(classes[5,1]+1) of
1: l1.Caption := 'BODE';
2: l1.Caption := 'BOTE';
3: l1.Caption := 'GATO';
4: l1.Caption := 'GADO';
5: l1.Caption := 'VACA';
6: l1.Caption := 'FACA';
end;
end;

// Função usada pela rede FAN
function Tfrm_Redde.quadro(num : real):real;
begin
    result := num*num;
end;

// Função usada pela rede FAN
function Tfrm_Redde.tests(neu : integer;arg : array of real):real;
var
    pi : Extended;
    i, j, iip : integer;
    idx, sidx, u, s : real;
    x : array [0..MAX_CAR] of real;
begin
    pi := 1.;
    i := 0;
    while (i < n_entr) do
    begin
        x[i] := (max_range-1.0) * arg[i];
        inc(i);
    end;
    j := 0;
    while (j < n_entr) do
    begin
        sidx := 0.;
        i := -d;
        while (i <= d) do
        begin
            iip := trunc(x[j]) + i;
            if (iip < 0) then
                iip := 0;
        end;
        j := j + 1;
    end;
end;

```

```

        if (iip > max_range-1) then
iip := max_range-1;
        idx := mu_x(iip,x[j],d);
        sidx := sidx + idx;
inc(i);
end;
i := -d;
while (i <= d) do
begin
        iip := trunc(x[j]) + i;
        if (iip < 0) then
iip := 0;
        if (iip > max_range-1) then
iip := max_range-1;
        idx := mu_x(iip,x[j],d);
        u := rede[neu][j][iip];
        s := rede[neu][j][max_range];
        if (s <> 0) then
            pi := pi * (1.-sqrt(idx*u/(s*sidx)));
inc(i);
end;
inc(j);
end;
result := (1 - power(pi, (1 / n_entr)));
end;

```

```

// Função usada pela rede FAN
function Tfrm_Redem.mu_x(x,centro, z: real): real;
var
    d1 : real;
begin
    d1 := z + 1;
    if (x < centro) then
        result := (1./d1)*(x+d1-centro)
    else
        result := (1./d1)*(-x+d1+centro);
end;

```

```

// Carrega os padrões do arquivo .txt (usados para o
// treinamento da rede) e aloca numa matriz
procedure Tfrm_Redem.LerPadroes();
var x,i,j: integer;
    linha,s: string;
begin
    memopd.Lines.LoadFromFile(padroes);
    i:=0;
    j:=0;
    x:=1;
    n_pdr:=memopd.Lines.Count;
    while j <= n_pdr-1 do
        begin
            linha:=memopd.Lines.Strings[j];
            s:="";
            while i < 7 do
                begin
                    if copy(linha,x,1) = '' then

```

```

    inc(x);
    while copy(linha,x,1) <> '' do
    begin
        s:=s+copy(linha,x,1);
        inc(x);
    end;
    mat[j,i] := strtoint(s);
    s := "";
    inc(i);
end;
mat[j,i] := strtoint(copy(linha,x+1,1));
inc(j);
x:=1;
i:=0;
end;
end;

```

// Normaliza uma série de dados para serem usados pela rede FAN

// por exemplo os padrões carregados e os valores nx

procedure Tfrm_Redde.Normaliza;

var

e, se : array [0..MAX_CAR] of real;

i, j, k : integer;

begin

i := 0;

while (i < n_neur) do

begin

ess[i] := 0;

inc(i);

end;

i := 0;

while (i < n_entr) do

begin

se[i] := 0;

inc(i);

end;

k := 0;

num_carac := 7;

SetLength(mat_carac,n_pdr-1);

for i := 0 to 6 do

for j := 0 to n_pdr-1 do

mat_carac[j,i] := mat[j,i];

while (k < num_carac) do

begin

j := 0;

while (j < n_entr) do

begin

e[j] := mat_carac[k,j];

inc(j);

end;

inc(k);

j := 0;

while (j < n_entr) do

begin

if (se[j] < e[j]) then

se[j] := e[j];

```

    inc(j);
  end;
end;
i := 0;
while (i < n_entr) do
begin
  nx[i] := se[i];
  inc(i);
end;
nx[n_entr] := 0;
i := 0;
while (i < n_entr) do
begin
  nx[n_entr] := nx[n_entr] + nx[i];
  inc(i);
end;
end;

```

// Carrega a rede FAN (gravada no arquivo .fan) e aloca na
// variável 'rede' para ser usada em tempo real

```

procedure Tfrm_Redde.CarregaRede;

```

```

var i,j,k : integer;

```

```

    cell : real;

```

```

    Arq : TextFile;

```

```

begin

```

```

  AssignFile(Arq, redefan);

```

```

  Reset(Arq);

```

```

  i := 0;

```

```

  while (i < n_neur) do

```

```

  begin

```

```

    j := 0;

```

```

    while (j <= max_range) do

```

```

    begin

```

```

      k := 0;

```

```

      while (k < n_entr) do

```

```

      begin

```

```

        read(Arq, cell);

```

```

        rede[i][k][j] := cell;

```

```

        inc(k);

```

```

      end;

```

```

      readln(Arq);

```

```

      inc(j);

```

```

    end;

```

```

    inc(i);

```

```

  end;

```

```

  CloseFile(Arq);

```

```

end;

```

// Função para fortalecimento de um neurônio (acerto)

```

procedure Tfrm_Redde.store(neu: integer);

```

```

var

```

```

  i, j, iip : integer;

```

```

  idx, u, s, mais : real;

```

```

  x : array [0..MAX_CAR] of real;

```

```

begin

```



```

i := 0;
while (i < n_entr) do
begin
  x[i] := (max_range-1.0) * bxin[i];
  inc(i);
end;
tempera := (10000.+random(5000))/10000.;
tempera := Power(tempera,0.1);
j := 0;
while (j < n_entr) do
begin
  i := -d;
  while (i <= d) do
  begin
    iip := trunc(x[j]) + i;
    if (iip < 0) then
      iip := 0;
    if (iip > max_range-1) then
      iip := max_range-1;
    idx := mu_x(iip,x[j],d);
    u := rede[neu][j][iip];
    s := rede[neu][j][max_range];
    mais := idx;
    u := u + mais;
    s := s + mais;
    rede[neu][j][iip] := u;
    rede[neu][j][max_range] := s;

    inc(i);
  end;
  inc(j);
end;
end;

```

// Função para enfraquecimento de um neurônio (erro)

```

procedure Tfrm_Rede.unstore(neu: integer);
var
  i, j, iip : integer;
  idx, u : real;
  x : array [0..MAX_CAR] of real;
begin
  i := 0;
  tempera := (10000.-random(9900))/10000.;
  tempera := Power(tempera,0.05);
  while (i < n_entr) do
  begin
    x[i] := (max_range-1.0) * bxin[i];
    inc(i);
  end;
  j := 0;
  while (j < n_entr) do
  begin
    i := -d;
    while (i <= d) do
    begin
      iip := trunc(x[j]) + i;
      if (iip < 0) then

```

```

iip := 0;
      if (iip > max_range-1) then
iip := max_range-1;
      idx := mu_x(iip,x[j],d);
      u := rede[neu][j][iip];
if (random(100) > 10) then
  u := u * power(idx,0.00001) * tempera
else
  u := u - idx;
if u < 0 then
  u := 0;
      rede[neu][j][iip] := u;
inc(i);
end;
inc(j);
end;
end;

// Função para salvar as modificações realizadas
// enfraquecimentos/fortalecimentos dos neurônios (treinamento)
procedure Tfrm_Rede.SalvarRede();
var   i, j, k : integer;
      cell : real;
      Arq : TextFile;
begin
  AssignFile(Arq,redefan);
  Rewrite(Arq);
  i := 0;
  while (i < n_neur) do
  begin
    j := 0;
    while (j <= max_range) do
    begin
      k := 0;
      while (k < n_entr) do
      begin
        cell := rede[i][k][j];
        write(Arq,cell);
        inc(k);
      end;
      writeln(Arq);
      inc(j);
    end;
    inc(i);
  end;
  CloseFile(Arq);
end;

// Botão 'Certo', salva as características como sendo
// pertencente à classe apontada pela rede e efetua
// o fortalecimento dessa classe (neurônio)
procedure Tfrm_Rede.btn_certoClick(Sender: TObject);
begin
  store(classe-1);
  memo1.Lines[0] := memo1.Lines[0]+inttostr(classe);
  memopd.Lines.LoadFromFile(padroes);

```

```

memopd.Lines.Add(memo1.Lines[0]);
memopd.Lines.SaveToFile(padroes);
btn_certo.Enabled := false;
btn_errado.Enabled := false;
btn_desfazer.Enabled := true;
end;

// Botão 'Errado', abre o form de escolha da palavra certa
procedure Tfrm_Redde.btn_erradoClick(Sender: TObject);
begin
    frm_Palavra.ShowModal;
end;

// Desfaz última inserção no arquivo de padrões, caso a última
// operação tenha sido efetuada erradamente, por exemplo escolheu
// a opção 'Certo' quando na verdade não era
procedure Tfrm_Redde.btn_desfazerClick(Sender: TObject);
begin
    memopd.Lines.LoadFromFile(padroes);
    memopd.Lines.Delete(memopd.Lines.Count-1);
    memopd.Lines.SaveToFile(padroes);
    btn_desfazer.Enabled := false;
end;

// Salva as modificações da rede e fecha o form de captura
procedure Tfrm_Redde.btn_sairClick(Sender: TObject);
begin
    SalvarRede();
    frm_Redde.Close;
end;

// Ao abrir o form ve a opção selecionada no form principal
// para abrir os padrões e a rede da cartela selecionada
procedure Tfrm_Redde.FormShow(Sender: TObject);
begin
    case frm_Principal.ComboBox1.ItemIndex of
        0: begin
            padroes:='padroes1.txt';
            redefan:='rede1.fan';
            cl1:='BODE';
            cl2:='BOTE';
            cl3:='GATO';
            cl4:='GADO';
            cl5:='VACA';
            cl6:='FACA';
            end;
        1: begin
            padroes:='padroes2.txt';
            redefan:='rede2.fan';
            cl1:='BODE';
            cl2:='BOTE';
            cl3:='GATO';
            cl4:='GADO';
            cl5:='VACA';
            cl6:='FACA';
            end;
    end;
end;

```

```
2: begin
  padroes:='padroes3.txt';
  redefan:='rede3.fan';
  cl1:='BODE';
  cl2:='BOTE';
  cl3:='GATO';
  cl4:='GADO';
  cl5:='VACA';
  cl6:='FACA';
end;
```

```
3: begin
  padroes:='padroes4.txt';
  redefan:='rede4.fan';
  cl1:='BODE';
  cl2:='BOTE';
  cl3:='GATO';
  cl4:='GADO';
  cl5:='VACA';
  cl6:='FACA';
end;
```

```
4: begin
  padroes:='padroes5.txt';
  redefan:='rede5.fan';
  cl1:='BODE';
  cl2:='BOTE';
  cl3:='GATO';
  cl4:='GADO';
  cl5:='VACA';
  cl6:='FACA';
end;
```

```
5: begin
  padroes:='padroes6.txt';
  redefan:='rede6.fan';
  cl1:='BODE';
  cl2:='BOTE';
  cl3:='GATO';
  cl4:='GADO';
  cl5:='VACA';
  cl6:='FACA';
end;
```

```
6: begin
  padroes:='padroes7.txt';
  redefan:='rede7.fan';
  cl1:='BODE';
  cl2:='BOTE';
  cl3:='GATO';
  cl4:='GADO';
  cl5:='VACA';
  cl6:='FACA';
end;
```

```
7: begin
  padroes:='padroes8.txt';
  redefan:='rede8.fan';
  cl1:='BODE';
  cl2:='BOTE';
  cl3:='GATO';
```

```

cl4:='GADO';
cl5:='VACA';
cl6:='FACA';
end;
8: begin
padroes:='padroes9.txt';
redefan:='rede9.fan';
cl1:='BODE';
cl2:='BOTE';
cl3:='GATO';
cl4:='GADO';
cl5:='VACA';
cl6:='FACA';
end;
9: begin
padroes:='padroes10.txt';
redefan:='rede10.fan';
cl1:='BODE';
cl2:='BOTE';
cl3:='GATO';
cl4:='GADO';
cl5:='VACA';
cl6:='FACA';
end;
10: begin
padroes:='padroes11.txt';
redefan:='rede11.fan';
cl1:='BODE';
cl2:='BOTE';
cl3:='GATO';
cl4:='GADO';
cl5:='VACA';
cl6:='FACA';
end;
11: begin
padroes:='padroes12.txt';
redefan:='rede12.fan';
cl1:='BODE';
cl2:='BOTE';
cl3:='GATO';
cl4:='GADO';
cl5:='VACA';
cl6:='FACA';
end;
12: begin
padroes:='padroes13.txt';
redefan:='rede13.fan';
cl1:='BODE';
cl2:='BOTE';
cl3:='GATO';
cl4:='GADO';
cl5:='VACA';
cl6:='FACA';
end;
13: begin
padroes:='padroes14.txt';

```

```

redefan:='rede14.fan';
cl1:='BODE';
cl2:='BOTE';
cl3:='GATO';
cl4:='GADO';
cl5:='VACA';
cl6:='FACA';
end;
14: begin
padroes:='padroes15.txt';
redefan:='rede15.fan';
cl1:='BODE';
cl2:='BOTE';
cl3:='GATO';
cl4:='GADO';
cl5:='VACA';
cl6:='FACA';
end;
end;
LerPadroes();
CarregaRede();
Normaliza();
end;

```

```

// Ao fechar o form limpa os objetos (gráficos, labels, memos)
procedure Tfrm_Redde.FormClose(Sender: TObject, var Action: TCloseAction);
begin
memo1.Clear;
memopd.Clear;
Series1.Clear;
Series2.Clear;
Series3.Clear;
I1.Caption := '';
I2.Caption := '';
I3.Caption := '';
I4.Caption := '';
I5.Caption := '';
end;

end.

```

Tela de Escolha da Palavra Certa para Adicionar no Arquivo de Padrões

```
unit uPalavra;  
  
interface  
  
uses  
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
  Dialogs, StdCtrls;  
  
type  
  Tfrm_Palavra = class(TForm)  
    cb: TComboBox;  
    Button1: TButton;  
    procedure Button1Click(Sender: TObject);  
    procedure FormShow(Sender: TObject);  
  private  
    { Private declarations }  
  public  
    { Public declarations }  
  end;  
  
var  
  frm_Palavra: Tfrm_Palavra;  
  
implementation  
  
uses uRede;  
  
{ $R *.dfm }  
  
procedure Tfrm_Palavra.Button1Click(Sender: TObject);  
var x: integer;  
begin  
  x:=cb.ItemIndex;  
  frm_Rede.unstore(x);  
  frm_Rede.memo1.Lines[0] := frm_Rede.memo1.Lines[0]+inttostr(x+1);  
  frm_Rede.memopd.Lines.LoadFromFile('padroes.txt');  
  frm_Rede.memopd.Lines.Add(frm_Rede.memo1.Lines[0]);  
  frm_Rede.memopd.Lines.SaveToFile('padroes.txt');  
  frm_Rede.btn_certo.Enabled := false;  
  frm_Rede.btn_errado.Enabled := false;  
  frm_Rede.btn_desfazer.Enabled := true;  
  frm_Palavra.Close;  
end;  
  
procedure Tfrm_Palavra.FormShow(Sender: TObject);  
begin  
  cb.SetFocus;  
end;  
  
end.
```

Tela de Refinamento dos Padrões

```
unit uRefinador;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Grids, Menus;

type
  Tfrm_Refinador = class(TForm)
    Memo1: TMemo;
    Memo3: TMemo;
    sv1: TSaveDialog;
    od1: TOpenDialog;
    Label1: TLabel;
    Label2: TLabel;
    Memo2: TMemo;
    Label9: TLabel;
    Label10: TLabel;
    MainMenu1: TMainMenu;
    Abrir1: TMenuItem;
    Salvar1: TMenuItem;
    Cancelar1: TMenuItem;
    procedure FormCreate(Sender: TObject);
    procedure Matriz;
    procedure Apagar;
    procedure Abrir1Click(Sender: TObject);
    procedure Salvar1Click(Sender: TObject);
    procedure Cancelar1Click(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  frm_Refinador: Tfrm_Refinador;
  mat: array[1..10,1..1000] of integer;
  i,j,n,n1,n2: integer;

implementation

{$R *.dfm}

procedure Tfrm_Refinador.FormCreate(Sender: TObject);
begin
  memo1.lines.LoadFromFile('fc.fsy');
  n1 := memo1.Lines.Count;
end;

procedure Tfrm_Refinador.Matriz;
var x,l: integer;
    a,num : string;
```



```

begin
while n < memo1.lines.count do
begin
a:=Memo1.Lines[n];
l:=10;
num:="";
while l<=96 do
begin
for x=l to l+5 do
num:=num+a[x];
mat[i,j]:=strtoint(num);
inc(i);
l:=l+16;
num:="";
end;
mat[7,j]:=strtoint(copy(a,99,1));
l:=mat[1,j];
i:=1;
for x=2 to 6 do
if mat[x,j] > l then
begin
l:=mat[x,j];
i:=x;
end;
mat[8,j]:=i;
if mat[7,j]=mat[8,j] then
mat[9,j]:=1
else
mat[9,j]:=0;
i:=1;
inc(n);
inc(j);
end;
end;

procedure Tfrm_Refinador.apagar;
var x: integer;
begin
for x=1 to j-1 do
if mat[9,x]<>0 then
memo3.Lines.add(memo2.Lines[x-1]);
end;

procedure Tfrm_Refinador.Abrir1Click(Sender: TObject);
var a,s: string;
x: integer;
begin
if od1.Execute then
begin
memo2.Lines.LoadFromFile(od1.filename);
n2 := memo2.Lines.Count;
memo2.Clear;
//if n1 = n2 then
//begin
memo2.Lines.LoadFromFile(od1.filename);
label1.Caption := inttostr(n1)+' linhas';

```

```

s := od1.FileName;
a := "";
x := length(s);
while (copy(s,x,1) <> '\') do
begin
  a := copy(s,x,1)+a;
  dec(x);
end;
frm_Refinador.Caption := a;
n:=0;
i:=1;
j:=1;
Matriz;
Apagar;
label2.Caption := inttostr(memo3.Lines.Count)+' linhas';
MainMenu1.Items[1].Enabled := true;
//end
//else
// ShowMessage('O arquivo fc.fsy não é o correto para esse arquivo de padrões.');
```

```

end
end;
```

```

procedure Tfrm_Refinador.Salvar1Click(Sender: TObject);
begin
  if sv1.Execute then
    memo3.Lines.SaveToFile(sv1.FileName);
end;
```

```

procedure Tfrm_Refinador.Cancelar1Click(Sender: TObject);
begin
  frm_Refinador.Close;
end;
```

```

procedure Tfrm_Refinador.FormClose(Sender: TObject;
var Action: TCloseAction);
begin
  Label1.Caption := "";
  label2.Caption := "";
  memo2.Clear;
  memo3.Clear;
  MainMenu1.Items[1].Enabled := false;
end;
```

```

end.
```

Tela de Contagem dos Padrões

```
unit uContador;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, StdCtrls, Grids, Menus, ExtCtrls;
```

```
type
```

```
Tfrm_Contador = class(TForm)
```

```
od1: TOpenDialog;
```

```
sg: TStringGrid;
```

```
memo: TMemo;
```

```
sg1: TStringGrid;
```

```
sg2: TStringGrid;
```

```
sg3: TStringGrid;
```

```
sg4: TStringGrid;
```

```
sg5: TStringGrid;
```

```
sg6: TStringGrid;
```

```
Memo1: TMemo;
```

```
sv1: TSaveDialog;
```

```
MainMenu1: TMainMenu;
```

```
Abrir1: TMenuItem;
```

```
Salvar1: TMenuItem;
```

```
Cancelar1: TMenuItem;
```

```
GroupBox1: TGroupBox;
```

```
lblpdr: TLabel;
```

```
GroupBox2: TGroupBox;
```

```
Label1: TLabel;
```

```
Label2: TLabel;
```

```
Label3: TLabel;
```

```
Label4: TLabel;
```

```
Label5: TLabel;
```

```
Label6: TLabel;
```

```
lbl1: TLabel;
```

```
lbl2: TLabel;
```

```
lbl3: TLabel;
```

```
lbl4: TLabel;
```

```
lbl5: TLabel;
```

```
lbl6: TLabel;
```

```
GroupBox3: TGroupBox;
```

```
Label7: TLabel;
```

```
Label8: TLabel;
```

```
Label9: TLabel;
```

```
Label10: TLabel;
```

```
Label11: TLabel;
```

```
Label12: TLabel;
```

```
Label13: TLabel;
```

```
lbc1: TLabel;
```

```
lbc2: TLabel;
```

```
lbc3: TLabel;
```

```
lbc4: TLabel;
```

```
lbc5: TLabel;
```

```
lbc6: TLabel;
```

```

lbc7: TLabel;
procedure Alocar;
procedure Adicionar(cl,n: integer);
procedure Salvar1Click(Sender: TObject);
procedure Abrir1Click(Sender: TObject);
procedure Cancelar1Click(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  frm_Contador: Tfrm_Contador;
  mat: array [0..2000,0..7] of integer;

implementation

{$R *.dfm}

procedure Tfrm_Contador.Adicionar(cl,n: integer);
begin
  case cl of
    1: memo1.Lines.Add(sg1.Cells[0,n]+'+sg1.Cells[1,n]+'+sg1.Cells[2,n]+'+sg1.Cells[3,n]+'
'+sg1.Cells[4,n]+'+sg1.Cells[5,n]+'+sg1.Cells[6,n]+'+sg1.Cells[7,n]);
    2: memo1.Lines.Add(sg2.Cells[0,n]+'+sg2.Cells[1,n]+'+sg2.Cells[2,n]+'+sg2.Cells[3,n]+'
'+sg2.Cells[4,n]+'+sg2.Cells[5,n]+'+sg2.Cells[6,n]+'+sg2.Cells[7,n]);
    3: memo1.Lines.Add(sg3.Cells[0,n]+'+sg3.Cells[1,n]+'+sg3.Cells[2,n]+'+sg3.Cells[3,n]+'
'+sg3.Cells[4,n]+'+sg3.Cells[5,n]+'+sg3.Cells[6,n]+'+sg3.Cells[7,n]);
    4: memo1.Lines.Add(sg4.Cells[0,n]+'+sg4.Cells[1,n]+'+sg4.Cells[2,n]+'+sg4.Cells[3,n]+'
'+sg4.Cells[4,n]+'+sg4.Cells[5,n]+'+sg4.Cells[6,n]+'+sg4.Cells[7,n]);
    5: memo1.Lines.Add(sg5.Cells[0,n]+'+sg5.Cells[1,n]+'+sg5.Cells[2,n]+'+sg5.Cells[3,n]+'
'+sg5.Cells[4,n]+'+sg5.Cells[5,n]+'+sg5.Cells[6,n]+'+sg5.Cells[7,n]);
    6: memo1.Lines.Add(sg6.Cells[0,n]+'+sg6.Cells[1,n]+'+sg6.Cells[2,n]+'+sg6.Cells[3,n]+'
'+sg6.Cells[4,n]+'+sg6.Cells[5,n]+'+sg6.Cells[6,n]+'+sg6.Cells[7,n]);
  end;
end;

procedure Tfrm_Contador.Abrir1Click(Sender: TObject);
var s,a: string;
    x: integer;
begin
  if od1.Execute then
    begin
      s := od1.FileName;
      a := "";
      x := length(s);
      while (copy(s,x,1) <> '\') do
        begin
          a := copy(s,x,1)+a;
          dec(x);
        end;
      frm_Contador.Caption := a;
      memo.Clear;
      memo.Lines.LoadFromFile(od1.FileName);
    end;
end;

```

```

if memo.Lines.Count > 0 then
begin
  Randomize;
  Alocar;
  MainMenu1.Items[1].Enabled := true;
end
else
begin
  lblpdr.Caption := '0 linhas';
  lbl1.Caption := "";
  lbl2.Caption := "";
  lbl3.Caption := "";
  lbl4.Caption := "";
  lbl5.Caption := "";
  lbl6.Caption := "";
  lbc1.Caption := "";
  lbc2.Caption := "";
  lbc3.Caption := "";
  lbc4.Caption := "";
  lbc5.Caption := "";
  lbc6.Caption := "";
  lbc7.Caption := "";
end;
end
end;

procedure Tfrm_Contador.Alocar;
var x,y,i,j,z,bd,bt,gt,gd,vc,fc,rd,bdn,btn,gtn,gdn,vcn,fcn,lst,c1,c2,c3,c4,c5,c6,c7: integer;
    linha,s: string;
    ha: boolean;
begin
  z:=0;
  i:=0;
  j:=0;
  x:=1;
  y:=memo.Lines.Count;
  while j <= y-1 do
  begin
    linha:=memo.Lines.Strings[j];
    if linha <> " then
    begin
      s:= "";
      while i < 7 do
      begin
        if copy(linha,x,1) = '' then
          inc(x);
        while copy(linha,x,1) <> '' do
        begin
          s:=s+copy(linha,x,1);
          inc(x);
        end;
        sg.Cells[i,z] := s;
        s := "";
        inc(i);
      end;
      sg.Cells[i,z] := copy(linha,x+1,1);
    end;
  end;
end;

```

```

inc(j);
inc(z);
x:=1;
i:=0;
end
else
inc(j);
end;
y:=y-(j-z);
lblpdr.caption:=inttostr(y)+' linhas';
sg.RowCount := y;
c1 := 0;
c2 := 0;
c3 := 0;
c4 := 0;
c5 := 0;
c6 := 0;
c7 := 0;
j := 0;
while j <= y-1 do
begin
if strtoint(sg.Cells[0,j]) > c1 then
c1 := strtoint(sg.Cells[0,j]);
if strtoint(sg.Cells[1,j]) > c2 then
c2 := strtoint(sg.Cells[1,j]);
if strtoint(sg.Cells[2,j]) > c3 then
c3 := strtoint(sg.Cells[2,j]);
if strtoint(sg.Cells[3,j]) > c4 then
c4 := strtoint(sg.Cells[3,j]);
if strtoint(sg.Cells[4,j]) > c5 then
c5 := strtoint(sg.Cells[4,j]);
if strtoint(sg.Cells[5,j]) > c6 then
c6 := strtoint(sg.Cells[5,j]);
if strtoint(sg.Cells[6,j]) > c7 then
c7 := strtoint(sg.Cells[6,j]);
inc(j);
end;
lbc1.Caption := IntToStr(c1);
lbc2.Caption := IntToStr(c2);
lbc3.Caption := IntToStr(c3);
lbc4.Caption := IntToStr(c4);
lbc5.Caption := IntToStr(c5);
lbc6.Caption := IntToStr(c6);
lbc7.Caption := IntToStr(c7);
bd := 0;
bt := 0;
gt := 0;
gd := 0;
vc := 0;
fc := 0;
for j:=0 to y-1 do
case strtoint(sg.Cells[7,j]) of
1:
begin
for x=0 to 7 do
sg1.Cells[x,bd] := sg.Cells[x,j];

```

```

    inc(bd);
end;
2:
begin
  for x=0 to 7 do
    sg2.Cells[x,bt] := sg.Cells[x,j];
    inc(bt);
  end;
3:
begin
  for x=0 to 7 do
    sg3.Cells[x,gt] := sg.Cells[x,j];
    inc(gt);
  end;
4:
begin
  for x=0 to 7 do
    sg4.Cells[x,gd] := sg.Cells[x,j];
    inc(gd);
  end;
5:
begin
  for x=0 to 7 do
    sg5.Cells[x,vc] := sg.Cells[x,j];
    inc(vc);
  end;
6:
begin
  for x=0 to 7 do
    sg6.Cells[x,fc] := sg.Cells[x,j];
    inc(fc);
  end;
end;
sg1.RowCount := bd;
sg2.RowCount := bt;
sg3.RowCount := gt;
sg4.RowCount := gd;
sg5.RowCount := vc;
sg6.RowCount := fc;
lbl1.Caption := inttostr(bd);
lbl2.Caption := inttostr(bt);
lbl3.Caption := inttostr(gt);
lbl4.Caption := inttostr(gd);
lbl5.Caption := inttostr(vc);
lbl6.Caption := inttostr(fc);
bdn := 0;
btn := 0;
gtn := 0;
gdn := 0;
vcn := 0;
fcn := 0;
j := 0;
Memo1.Clear;
while j < y-1 do
begin
  rd:=random(6);

```

```

case rd of
0:
begin
if (fc<=fc) and ((lst<>1) and (lst<>2)) then
begin
Adicionar(1, bdn);
inc(bdn);
lst := 1;
inc(j);
end;
end;
1:
begin
if (fc<=fc) and ((lst<>1) and (lst<>2)) then
begin
Adicionar(2, btn);
inc(btn);
lst := 2;
inc(j);
end;
end;
2:
begin
if (fc<=fc) and ((lst<>4) and (lst<>3)) then
begin
Adicionar(3, gtn);
inc(gtn);
lst := 3;
inc(j);
end;
end;
3:
begin
if (fc<=fc) and ((lst<>4) and (lst<>3)) then
begin
Adicionar(4, gdn);
inc(gdn);
lst := 4;
inc(j);
end;
end;
4:
begin
if (fc<=fc) and ((lst<>5) and (lst<>6)) then
begin
Adicionar(5, vcn);
inc(vcn);
lst := 5;
inc(j);
end;
end;
5:
begin
if (fc<=fc) and ((lst<>5) and (lst<>6)) then
begin
Adicionar(6, fcn);

```



```

        inc(fcn);
        lst := 6;
        inc(j);
    end;
end;
end;
end;
{ ha := true;
while ha do
begin
    ha := false;
    for j:=0 to memo1.Lines.Count do
    begin
        linha:=memo1.Lines.Strings[j];
        if linha = '    ' then
        begin
            memo1.Lines.Delete(j);
            ha := true;
            break;
        end;
    end;
end; }
end;
end;

```

```

procedure Tfrm_Contador.Salvar1Click(Sender: TObject);
begin
    if sv1.Execute then
        memo1.Lines.SaveToFile(sv1.FileName);
end;

```

```

procedure Tfrm_Contador.Cancelar1Click(Sender: TObject);
begin
    frm_Contador.Close;
end;

```

```

procedure Tfrm_Contador.FormClose(Sender: TObject;
var Action: TCloseAction);
begin
    lblpdr.Caption := "";
    lbl1.Caption := "";
    lbl2.Caption := "";
    lbl3.Caption := "";
    lbl4.Caption := "";
    lbl5.Caption := "";
    lbl6.Caption := "";
    lbc1.Caption := "";
    lbc2.Caption := "";
    lbc3.Caption := "";
    lbc4.Caption := "";
    lbc5.Caption := "";
    lbc6.Caption := "";
    lbc7.Caption := "";
    MainMenu1.Items[1].Enabled := false;
end;

end.

```

Tela de Gravação dos Padrões no Banco

```
unit uGravarPadroes;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, StdCtrls, Grids, Menus, DB, DBTables, ZAbstractRODataset,  
ZAbstractDataset, ZDataset, ZConnection;
```

```
type
```

```
Tfrm_GravarPadroes = class(TForm)  
  od1: TOpenDialog;  
  memo: TMemo;  
  sg: TStringGrid;  
  conexao: TZConnection;  
  q: TZQuery;  
  MainMenu1: TMainMenu;  
  Abrir1: TMenuItem;  
  Executar1: TMenuItem;  
  Fechar1: TMenuItem;  
  GroupBox1: TGroupBox;  
  lblpdr: TLabel;  
  procedure Alocar;  
  procedure Abrir1Click(Sender: TObject);  
  procedure Executar1Click(Sender: TObject);  
  procedure Fechar1Click(Sender: TObject);  
private  
  { Private declarations }  
public  
  { Public declarations }  
end;
```

```
var
```

```
frm_GravarPadroes: Tfrm_GravarPadroes;  
mat: array [0..2000,0..7] of integer;
```

```
implementation
```

```
{$R *.dfm}
```

```
procedure Tfrm_GravarPadroes.Alocar;
```

```
var x,y,i,j: integer;
```

```
    linha,s: string;
```

```
begin
```

```
  i:=0;
```

```
  j:=0;
```

```
  x:=1;
```

```
  y:=memo.Lines.Count;
```

```
  sg.RowCount := y;
```

```
  while j <= y-1 do
```

```
  begin
```

```
    linha:=memo.Lines.Strings[j];
```

```
    s:="";
```

```
    while i < 7 do
```

```

begin
  if copy(linha,x,1) = '' then
    inc(x);
  while copy(linha,x,1) <> '' do
    begin
      s:=s+copy(linha,x,1);
      inc(x);
    end;
    sg.Cells[i,j] := s;
    s := "";
    inc(i);
  end;
  sg.Cells[i,j] := copy(linha,x+1,1);
  inc(j);
  x:=1;
  i:=0;
end;

end;

procedure Tfrm_GravarPadroes.Abrir1Click(Sender: TObject);
var s,a: string;
    x integer;
begin
  if od1.Execute then
    begin
      memo.Lines.LoadFromFile(od1.FileName);
      s := od1.FileName;
      a := "";
      x := length(s);
      while (copy(s,x,1) <> '\') do
        begin
          a := copy(s,x,1)+a;
          dec(x);
        end;
      frm_GravarPadroes.Caption := a;
      lblpdr.caption:=inttostr(memo.Lines.Count)+' linhas';
      Alocar;
    end
  end;
end;

procedure Tfrm_GravarPadroes.Executar1Click(Sender: TObject);
var x integer;
    pl_cod,c1,c2,c3,c4,c5,c6,c7: string;
begin
  for x=0 to memo.Lines.Count-1 do
    begin
      q.SQL.Clear;
      case strtoint(sg.Cells[7,x]) of
        1: pl_cod := '86';
        2: pl_cod := '87';
        3: pl_cod := '89';
        4: pl_cod := '88';
        5: pl_cod := '90';
        6: pl_cod := '91';
      end;
    end;
  end;
end;

```

```
c1 := sg.Cells[0,x];
c2 := sg.Cells[1,x];
c3 := sg.Cells[2,x];
c4 := sg.Cells[3,x];
c5 := sg.Cells[4,x];
c6 := sg.Cells[5,x];
c7 := sg.Cells[6,x];
q.SQL.Clear;
q.SQL.Add('insert into carac_15 values ('+pl_cod+', '+c1+', '+c2+', '+c3+', '+c4+', '+c5+', '+c6+', '+c7+')');
q.ExecSQL;
end;
end;

procedure Tfrm_GravarPadroes.Fechar1Click(Sender: TObject);
begin
  Application.Terminate;
end;

end.
```

Tela de Gravação da Rede FAN no Banco

```
unit uGravarRede;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, StdCtrls, ZConnection, DB, ZAbstractRODataset, ZAbstractDataset,  
ZDataset, ExtCtrls;
```

```
type
```

```
Tfrm_GravarRede = class(TForm)  
  Button1: TButton;  
  query_caracteristica: TZQuery;  
  ZConnection1: TZConnection;  
  cb1: TComboBox;  
  Bevel1: TBevel;  
  procedure salvar_rede_banco;  
  procedure CarregaRede(nome: string);  
  procedure Button1Click(Sender: TObject);
```

```
private
```

```
{ Private declarations }
```

```
public
```

```
{ Public declarations }
```

```
end;
```

```
const
```

```
MAX_CLA = 6;  
MAX_CAR = 7;  
MAX_RAN = 100;  
n_neur = 6;  
n_entr = 7;  
max_range = 100;
```

```
var
```

```
frm_GravarRede: Tfrm_GravarRede;  
rede : array [0..MAX_CLA,0..MAX_CAR,0..MAX_RAN] of real;  
cart : integer;
```

```
implementation
```

```
{$R *.dfm}
```

```
procedure Tfrm_GravarRede.salvar_rede_banco;
```

```
var aux : string;
```

```
  i,j,k : integer;
```

```
begin
```

```
  with query_caracteristica do
```

```
  begin
```

```
    Close;
```

```
    SQL.Clear;
```

```
    SQL.Add('select rede_'+inttostr(cart)+' from redes');
```

```
    Open;
```

```
    if (Fields[0].AsInteger = 1) then
```

```
    begin
```

```
      Close;
```

```

SQL.Clear;
SQL.Add('drop table rede_'+inttostr(cart)+';');
ExecSQL;
Close;
SQL.Clear;
SQL.Add('delete from nx where rede_codigo = '+inttostr(cart)+';');
ExecSQL;
end;
Close;
SQL.Clear;
aux := 'create table rede_'+inttostr(cart)+' (rede_neuronio int,rede_carac int';
for i := 0 to max_range do
  aux := aux + ',rede_range_'+inttostr(i)+' double precision';
aux := aux + ')';
SQL.Add(aux);
ExecSQL;
Close;
SQL.Clear;
SQL.Add('update redes set rede_'+inttostr(cart)+' = 1');
ExecSQL;
Close;
SQL.Clear;
aux := 'insert into rede_'+inttostr(cart)+' values (:neuronio,,:carac';
for i := 0 to max_range do
  aux := aux + ',:rede_range_'+inttostr(i);
  aux := aux + ')';
SQL.Add(aux);
for i := 0 to n_neur-1 do
begin
  Params[0].Value := i;
  For j := 0 to n_entr-1 do
begin
  Close;
  Params[1].Value := j;
  For k := 0 to max_range do
    Params[k+2].Value := rede[i][j][k];
    ExecSQL;
  end;
end;
end;
end;
end;

```

```

procedure Tfrm_GravarRede.CarregaRede(nome: string);
var i,j,k : integer;
    cell : real;
    Arq : TextFile;
begin
  AssignFile(Arq,nome);
  Reset(Arq);
  i := 0;
  while (i < n_neur) do
begin
  j := 0;
  while (j <= max_range) do
begin
  k := 0;

```

```

while (k < n_entr) do
begin
read(Arq,cell);
rede[i][k][j] := cell;
inc(k);
end;
readln(Arq);
inc(j);
end;
inc(i);
end;
CloseFile(Arq);
end;

```

```

procedure Tfrm_GravarRede.Button1Click(Sender: TObject);
var a: string;
begin
if MessageDlg('Esta operação afetará dados gravados no banco de dados. Você tem certeza que deseja continuar?', mtWarning, mbOKCancel, 0)=mrOK then
case cb1.ItemIndex of
0: begin a := 'rede1.fan'; cart := 1; end;
1: begin a := 'rede2.fan'; cart := 2; end;
2: begin a := 'rede3.fan'; cart := 3; end;
3: begin a := 'rede4.fan'; cart := 4; end;
4: begin a := 'rede5.fan'; cart := 5; end;
5: begin a := 'rede6.fan'; cart := 6; end;
6: begin a := 'rede7.fan'; cart := 7; end;
7: begin a := 'rede8.fan'; cart := 8; end;
8: begin a := 'rede9.fan'; cart := 9; end;
9: begin a := 'rede10.fan'; cart := 10; end;
10: begin a := 'rede11.fan'; cart := 11; end;
11: begin a := 'rede12.fan'; cart := 12; end;
12: begin a := 'rede13.fan'; cart := 13; end;
13: begin a := 'rede14.fan'; cart := 14; end;
14: begin a := 'rede15.fan'; cart := 15; end;
end;
CarregaRede(a);
salvar_rede_banco;
ShowMessage('Rede gravada no banco com sucesso.');
```