

UERJ

Dissertação de Mestrado em Engenharia de Computação
Área de Concentração em Geomática

**SISTEMA DE INFORMAÇÃO GEOGRÁFICA PARA AUXILIAR
A IMPLANTAÇÃO E AVALIAÇÃO DE PROVEDORES DE
INTERNET - SIGIAPI**

Autor: Carlos Augusto Sicsú Ayres do Nascimento

Orientador: Orlando Bernardo Filho

Programa de Pós Graduação em Engenharia de Computação
Área de Concentração em Geomática

Janeiro - 2004



Faculdade de Engenharia

SISTEMA DE INFORMAÇÃO GEOGRÁFICA PARA AUXILIAR A IMPLANTAÇÃO E AVALIAÇÃO DE PROVEDORES DE INTERNET - SIGAIPI

Carlos Augusto Sicsú Ayres do Nascimento

Dissertação submetida ao corpo docente da Faculdade de Engenharia da Universidade do Estado do Rio de Janeiro – UERJ, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia de Computação – Área de Concentração Geomática.

Orientador: Prof. Orlando Bernardo Filho, D.Sc.

Programa de Pós Graduação em Engenharia de Computação
Área de Concentração em Geomática

RIO DE JANEIRO, RJ - BRASIL

JANEIRO DE 2004

NASCIMENTO,CARLOS AUGUSTO
SICSÚ AYRES DO

Sistema de Informação Geográfica para
Auxiliar a Implantação e Avaliação de
Provedores de Internet –SIGAI-API [Rio
de Janeiro] 2004

XI, 122 p. 29,7 cm (FEN/UERJ, M.Sc.,
Engenharia de Computação – Área de
Concentração Geomática, 2004)

Tese - Universidade do Estado do Rio de
Janeiro - UERJ

1. Lógica Nebulosa
2. Sistema de Informação Geográfica
3. Cartografia
4. Banco de Dados
5. Internet
6. Aplicações para a WEB

I. FEN/UERJ II. Título (série)

FOLHA DE JULGAMENTO

Título: Sistema de Informação Geográfica para Auxiliar a Implantação e Avaliação de Provedores de Internet – SIGAI-API

Candidato: Carlos Augusto Sicsú Ayres do Nascimento

Programa de Pós Graduação em Engenharia de Computação

Área de Concentração em Geomática

Data da defesa: 27 de janeiro de 2004.

Aprovada por:

Prof. Orlando Bernardo Filho, D.Sc., UERJ

Prof. João Araújo Ribeiro, Dr., UERJ

Prof. Oscar Luis Monteiro de Farias, D.Sc., UERJ

Prof. Aloysio de Castro Pinto Pedroza, Dr., UFRJ.

Prof. Roberto Pires Vasques, D.Sc., UNESA

Dedicatória

À minha mulher Adriana e ao
nosso filho Matheus que está para
nascer, ao meu pai Fernando e à
minha mãe Hália pelo total apoio
nas horas difíceis durante o curso
e no decorrer deste trabalho.

AGRADECIMENTOS

À UERJ.

Ao departamento de Engenharia da Computação e ao departamento de Engenharia Cartográfica, pelo padrão do curso.

Ao professor orientador Orlando Bernardo Filho, por ter sido paciente com minhas ausências e um orientador exemplar.

Aos amigos Luis Serpa e Paula Moeller, pelo apoio e ajuda na configuração e no início da programação.

Ao professor Oscar Luis Monteiro de Farias, que sempre me incentivou e orientou para que me formasse de forma honrosa e ética.

À professora Margareth Simões Penello Meirelles, pelo apoio aos “antigos discentes” nas horas difíceis.

Ao Professor Íris Escobar, pelo apoio no curso de cartografia e no auxílio para o cálculo da distância entre os municípios.

À “Mãe Beth”, Maria Elisabeth Nascimento da Silva, que sempre tratou os alunos como uma mãe, onde fez sempre o que podia e até talvez o que não podia, mas, também nunca nos passou a mão pela cabeça, nos cobrando sempre.

Aos demais professores e amigos do curso de Geomática, cada um ao seu jeito contribuiu para com minha formação.

Aos meus chefes, José Luis e André de Castro que sempre acreditaram no meu sucesso.

Aos amigos que aqui deixo, Oscar, Orlando, Margareth, Beth que nunca esquecerei nas próximas jornadas e etapas da vida.

Aos amigos que sempre me incentivaram e apoiaram José Luis, Letícia, André, Carlos Alexandre, Martha Barbedo, Reginaldo, Maria Cristina.

Resumo da Dissertação apresentada à FEN/UERJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.).

SISTEMA DE INFORMAÇÃO GEOGRÁFICA PARA AUXILIAR A IMPLANTAÇÃO E AVALIAÇÃO DE PROVEDORES DE INTERNET - SIGAIPI

Carlos Augusto Sicsú Ayres do Nascimento

Janeiro/2004

Orientador: Orlando Bernardo Filho, Dsc., UERJ.

Programa de Pós-Graduação em Engenharia de Computação - Área de Concentração
Geomática – Mestrado

Esta dissertação tem como objetivo desenvolver um Sistema de Informações Geográficas que contenha informações sobre os municípios e os provedores de acesso à Internet do Estado do Rio de Janeiro - Brasil. Estas informações estão relacionadas com a população, com a qualidade das linhas de comunicação de dados, da renda, urbanização e concorrência em uma dada região composta por um determinado grupo de municípios. O SIG permitirá consultas para se obter a melhor opção para implantar um novo provedor de acesso.

Abstract of Dissertation presented to FEN/UERJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.).

GEOGRAPHIC INFORMATION SYSTEM TO TO SET AND EVALUATE NEW INTERNET HOSTS – GIS2SEIH

Carlos Augusto Sicsú Ayres do Nascimento

January/2004

Advisors: Orlando Bernardo Filho

Program Computing Engineering - Geomatic

This dissertation aims the development of a Geographical Information System (SIG) capable of keeping data concerning cities of Rio de Janeiro State (Brazil) and Internet Service Providers (ISP) there based. Information to be gathered includes population, quality of available communication media, personal income of users, urbanization and competition in regions composed by certain groups of cities. System will allow queries in order to achieve best options in the planning of new ISP.

ÍNDICE

CAPÍTULO I	- 1 -
INTRODUÇÃO	- 1 -
I.1 APRESENTAÇÃO	- 1 -
I.2 OBJETIVO	- 2 -
I.3 CONTRIBUIÇÕES	- 3 -
I.4 SUMÁRIO	- 3 -
CAPÍTULO II	- 4 -
FUNDAMENTOS TEÓRICOS	- 4 -
II.1 INTRODUÇÃO	- 4 -
II.2 LÓGICA NEBULOSA	- 4 -
II.2.1 Teoria dos Conjuntos Nebulosos	- 4 -
II.2.2 Sistema de Inferência	- 9 -
II.3 CARTOGRAFIA	- 12 -
II.4 FORMAÇÃO DAS REGIÕES PARA O ATENDIMENTO DO PROVEDOR	- 12 -
II.4.1 O Sistema de Formação das Regiões de Atendimento Proposto	- 13 -
II.4.2 O Cálculo da Distância Entre os Municípios	- 14 -
II.5 GERENCIADOR DE BANCO DE DADOS	- 20 -
II.6 SISTEMAS DE INFORMAÇÕES GEOGRÁFICAS (SIG)	- 23 -
II.7 DESENVOLVIMENTO PARA A WEB	- 28 -
II.8 COMENTÁRIOS	- 32 -
CAPÍTULO III	- 33 -
ORGANIZAÇÃO DO SISTEMA	- 33 -
III.1 INTRODUÇÃO	- 33 -

III.2 O USO DO HIPERTEXTO (HTML)	- 33 -
III.3 O USO DO ASP	- 34 -
III.4 DESCRIÇÃO LÓGICA DO SISTEMA	- 35 -
III.4.1 Funcionamento Básico do Sistema	- 35 -
III.4.2 Definição das Interfaces com o Usuário	- 37 -
III.4.2.1 A Página Index.Html	- 37 -
III.4.2.2 A Página Index.Asp	- 37 -
III.4.2.3 A Página Fuzzy.Asp	- 39 -
III.5 A CRIAÇÃO DA DLL (DYNAMIC LINK LIBRARY) FUZZY	- 40 -
III.5.1 A Classe clsAcesso	- 41 -
III.5.2 A Classe clsFuzzy	- 42 -
III.6 A BASE DE DADOS	- 47 -
III.6.1 A Tabela Usuários	- 47 -
III.6.2 A Tabela Funções	- 48 -
III.6.3 A Tabela Regras	- 48 -
III.6.4 A Tabela Municípios	- 49 -
III.6.5 A Tabela Síntese	- 50 -
III.7 COMENTÁRIOS	- 51 -
 CAPÍTULO IV	 - 52 -
IMPLANTAÇÃO DO SIGAPI	- 52 -
IV.1 INTRODUÇÃO	- 52 -
IV.2 DEFINIÇÃO DO PROJETO	- 52 -
IV.3 APLICAÇÃO DA TEORIA DOS CONJUNTOS NEBULOSOS AO PROBLEMA	- 53 -
IV.4 DEFINIÇÃO DAS REGRAS DE INFERÊNCIA	- 68 -
IV.5 DEFINIÇÃO DAS CLASSES E MÉTODOS	- 72 -
IV.5.1 A Classe clsAcesso	- 72 -

IV.5.2 A Classe clsFuzzy _____	- 73 -
IV.5.2.1 - Métodos Responsáveis pelo Acesso às Tabelas e pela Montagem do Grupo de Regras e das Funções de Pertinência	- 74 -
IV.5.2.2 - Métodos Responsáveis pelos Cálculos Cartográficos e Geração da Região a Partir do Município Base _____	- 75 -
IV.5.2.3 - Métodos Responsáveis por Formatar a Resposta e Enviar os Resultados por E-Mail _____	- 77 -
IV.5.2.4 - Métodos Responsáveis por Efetuar a Inferência (Motor de Inferência do Sistema) _____	- 78 -
IV.6 TESTES REALIZADOS _____	- 85 -
IV.6.1 Teste Com O Município De Saquarema _____	- 85 -
IV.6.2 Teste Com O Município De Três Rios _____	- 87 -
IV.7 COMENTÁRIOS _____	- 89 -
 CAPÍTULO V _____	 - 90 -
CONCLUSÕES _____	- 90 -
 REFERÊNCIAS BIBLIOGRÁFICAS _____	 - 93 -
 APÊNDICES _____	 - 98 -
APÊNDICE A – REGRAS _____	- 98 -
APÊNDICE B1 – CODIFICAÇÃO: CLASSE CLSACESSO _____	- 106 -
APÊNDICE B2 – CODIFICAÇÃO: CLASSE CLSFUZZY _____	- 106 -

CAPÍTULO I

Introdução

I.1 Apresentação

A Internet é um conglomerado de dezenas de milhares de redes eletrônicas interconectadas, criando um meio global de comunicação. Atualmente, a Internet é uma das áreas que mais cresce na informática, com a atual expansão já temos todo o globo terrestre interligado, trocando informações, fazendo negócios e interagindo entre si.

Existe uma grande quantidade de serviços disponíveis na Internet, entre eles o mais popular é o WWW (*World Wide Web*), devido a uma interface gráfica interativa, simples e de fácil utilização, em que qualquer usuário desfruta dos seus recursos. A WWW baseia-se no conceito de *HiperTexto*, onde palavras e imagens possuem referências próprias ou *links* com outros pontos no mundo, facilitando assim a navegação pela extensa rede.

O comércio eletrônico com uso da Internet já cresceu bastante, mas ainda tem um imenso potencial para continuar crescendo, por conta disso todos os grandes centros de pesquisa que desenvolvem trabalhos voltados para o aperfeiçoamento da Internet, têm objetivado oferecer meios para que os empresários dispostos a explorar a Internet como ferramenta, tenham maiores garantias no que diz respeito à segurança e velocidade de tal forma a disponibilizar os seus produtos e serviços com excelência na qualidade.

A expansão da Internet é uma certeza e, portanto, novos provedores de acesso e páginas surgirão em diversas cidades do mundo, hospedando as mais variadas opções de negócios, sendo oferecidos a uma imensidão de clientes. É possível perceber que um dado empresário interessado em implantar um serviço de acesso à Internet se defrontará com uma série de questões que irão afetar a sua decisão final do local exato para instalar o seu provedor.

Podem ser citadas como questões que influenciarão a decisão dos empresários: o total de pessoas que residem na região a ser oferecido o serviço; a faixa de poder aquisitivo da comunidade alvo; a disponibilidade de linhas para acesso à Internet, bem como a banda passante de tais linhas; o número de provedores já instalados na região; e, a disponibilidade e a qualidade das linhas de telefonia convencionais dos usuários da comunidade.

Levando-se em conta as questões citadas acima, seria de grande utilidade a existência de uma base de dados que reunisse todas essas informações de modo a permitir não apenas o melhor caminho para o desenvolvimento dos negócios virtuais, como também um acompanhamento de tal expansão, pois no caso dos órgãos do governo voltados para o planejamento homogêneo do progresso do seu território, essa mesma base de dados forneceria um panorama das áreas carentes de investimento para um melhor desenvolvimento ordenado a fim de se evitar distorções na população que venham a prejudicar o crescimento da economia do país e a conseqüente qualidade de vida.

Uma base de dados que reúne as informações sobre provedores de Internet não seria útil apenas para auxiliar a implantação dos mesmos, mas também ajudaria a um empresário que estivesse interessado em explorar o comércio eletrônico, pois ele teria que escolher o melhor provedor para hospedar a sua loja virtual, de acordo com o perfil e a localização geográfica do público alvo consumidor dos seus produtos. Tal empresário deveria levar em conta, por exemplo, as seguintes questões: pontos de distribuição dos seus produtos na região do seu público; pontos de exposição dos seus produtos; e, o provedor mais adequado para o seu público de acordo com a sua distância e capacidade de linha de tal forma que as páginas da loja possam ser acessadas com rapidez e confiabilidade.

I.2 Objetivo

Este trabalho tem como objetivo desenvolver um Sistema de Informação Geográfica [1,2,3] que contenha informações sobre os provedores de acesso à Internet no estado do Rio de Janeiro relacionadas com informações da população e com a disponibilidade e qualidade das linhas de telefonia pública e linhas para comunicação de dados.

O Sistema de Informação Geográfica desenvolvido oferecerá consulta sobre a viabilidade de se implantar novos provedores de acesso no estado do Rio de Janeiro.

No que diz respeito à montagem da base de dados propriamente dita, foram coletados dados sobre a população do estado do Rio de Janeiro a partir dos resultados dos censos realizados pelo Instituto Brasileiro de Geografia Estatística (IBGE), foram coletados dados sobre as linhas de telefonia e comunicação de dados, através dos *sites* das empresas prestadoras desses serviços, além de terem sido feitas consultas através de mecanismos de busca, para obter uma lista dos provedores de Internet do estado do Rio de Janeiro.

As análises para as consultas a serem processadas no Sistema de Informação Geográfica serão desenvolvidas com o emprego da definição e implementação dos chamados

Sistemas de Lógica Nebulosa que são bem adequados para lidar com a imprecisão das faixas de valores, caso característico do problema considerado, ao invés de valores perfeitamente precisos.

I.3 Contribuições

A técnica de apoio a decisão desenvolvida para o sistema constitui uma abordagem que envolve planejamento urbano visto que o SIGAI-API de maneira indireta indica carências de infra-estrutura em algum(ns) município(s) do estado do Rio de Janeiro. O trabalho aqui descrito foi publicado em [4] que até então trata de uma ferramenta inédita, acessada via *Web*, para nortear investimentos na área tecnológica por parte de empresários do setor de telecomunicações como também investimentos na área de infra-estrutura pública.

I.4 Sumário

No capítulo II, mostraremos uma visão conceitual da Lógica Nebulosa (*Fuzzy Logic*) e de um sistema de inferência, necessários para o desenvolvimento do projeto. Daremos uma introdução aos elementos da Cartografia, para auxiliar o entendimento do cálculo da distância entre dois pontos na superfície do globo terrestre. Veremos, assim, os passos necessários para calcularmos a distância entre dois pontos do globo terrestre, permitindo a definição da área de atuação do provedor. Geraremos então uma região com os municípios próximos o suficiente para poderem ser atendidos pelo provedor. Esta dissertação propõe uma nova forma de cobrança de ligações, para facilitar as ligações entre o provedor e seus usuários, além de outros benefícios. Neste capítulo, os conceitos fundamentais para a implementação do projeto são descritos de forma a explicar e exemplificar o seu uso.

No capítulo III, abordaremos a solução, indicando o fluxo dos acontecimentos e como os fundamentos teóricos foram aplicados no projeto para a resolução dos problemas apresentados. Neste capítulo teremos as descrições das técnicas empregadas para a resolução do problema, de acordo com os fundamentos teóricos observados no capítulo II.

No capítulo IV, veremos o projeto do SIGAI-API com detalhes, mostrando como foram implementadas as soluções na prática.

CAPÍTULO II

Fundamentos Teóricos

II.1 Introdução

Para que possamos entender as técnicas empregadas no trabalho, são necessários conhecimentos conceituais prévios da teoria dos conjuntos nebulosos e do funcionamento de um Sistema de Inferência Nebuloso, descritos respectivamente nas seções II.2.1 e II.2.2.

No caso das regiões geográficas que aqui propomos, devemos conhecer os cálculos necessários para obtermos a distância em duas posições distintas do globo terrestre, para a obtenção desta distância, são necessários conhecimentos de cartografia, descritos na seção II.3. Com base nos conhecimentos de cartografia, podemos então utilizar as fórmulas de Molodenski em BONFORD [5] para o cálculo da distância entre dois pontos georeferenciados, descrito na seção II.4.

Ainda são necessários os conhecimentos prévios da utilização de um gerenciador de banco de dados, descritos na seção II.5 e os conceitos que definem os sistemas de informações geográficos (SIG), descritos na seção II.6. Ao final, juntamos a tudo isso os conhecimentos necessários para o desenvolvimento de aplicações para a *Web*, descritos na seção II.7

II.2 Lógica Nebulosa

Uma aplicação que use a Lógica Nebulosa, deve utilizar os conceitos da teoria dos conjuntos nebulosos combinados com o uso de um sistema de inferência nebuloso.

II.2.1 Teoria dos Conjuntos Nebulosos

Como sabemos, a Lógica Nebulosa é uma aplicação da teoria dos conjuntos nebulosos [6,7,8,9,10,11,12,13,14,15,16,17,18,19]. Dentro do estudo da Lógica Nebulosa e do raciocínio aproximado é bastante utilizada uma ferramenta conhecida como variável lingüística, também chamada de variável de ordem mais alta. Essas variáveis não possuem números como valores, mas termos ou sentenças de uma linguagem natural ou artificial.

A idéia de se utilizar variáveis lingüísticas com valores de palavras ou frases de linguagens naturais ou artificiais está relacionada ao fato desses valores não serem tão específicos como números. Nesse sentido, nas ocasiões em que o tratamento de um valor

numérico preciso é irrelevante para o resultado final da operação, como em controle de sistemas não lineares, por exemplo, a manipulação dos valores lingüísticos é mais fácil, pois esses estão em menor quantidade e são mnemônicos.

Uma variável lingüística é definida por uma quintupla $(x, T(x), U, G, \tilde{M})$ onde:

- x é o nome da variável;
- $T(x)$ denota o conjunto de termos de x , isto é, o conjunto de nomes dos **valores lingüísticos** de x com cada valor sendo um conjunto nebuloso;
- U é o universo de discurso dos conjuntos nebulosos que formam os termos de $T(x)$;
- G é a regra sintática, que usualmente tem a forma de uma gramática, para gerar os nomes dos valores lingüísticos;
- $\tilde{M}(X)$ é a regra semântica que atribui um significado ao termo X do conjunto $T(x)$, ou seja, $\tilde{M}(X)$ é um subconjunto nebuloso de U .

Como exemplo, podemos citar a variável lingüística com rótulo x =velocidade, com conjunto de termos $T(\text{velocidade})=\{\text{muito_lenta}, \text{lenta}, \text{rápida}, \text{muito_rápida}\}$, universo de discurso $U=[10\text{Km/h}, 120\text{Km/h}]$ e um dos valores $\tilde{M}(X)$ como mostrado a seguir.

$$\tilde{M}(\text{lenta}) = \{(u, \mu_{\text{lenta}}(u)) \mid u \in [10\text{Km/h}, 120\text{Km/h}]\} \quad (\text{Fórmula II.1})$$

Uma variável lingüística é dita estruturada se a regra semântica $\tilde{M}(X)$ e o conjunto de termos $T(x)$ podem ser gerados algoritmicamente, usando para tanto a regra sintática de construção dos termos G e os chamados modificadores (ou *hedges*) que consiste de uma operação para alterar o significado de um termo mais geral, obtendo assim, um conjunto nebuloso a partir de outro.

Como exemplo de aplicação dos modificadores, podemos citar o caso do conjunto nebuloso *lenta*, valor lingüístico da variável velocidade, que poderia ser alterado para *muito_lenta* com o uso da operação de concentração, que na verdade trata-se de um modificador, para se obter o termo com o intensificador *muito*. Neste trabalho, optou-se por utilizar apenas funções lineares, mas, em outras ocasiões podemos obter o termo *muito_lenta*, elevando-se a função de pertinência do termo básico ao quadrado. Assim sendo, a função de pertinência de *muito_lenta* seria:

$$\mu_{\text{muito_lenta}}(u) = (\mu_{\text{lenta}}(u))^2 \quad (\text{Fórmula II.2})$$

A Lógica Nebulosa normalmente é empregada na construção dos chamados Sistemas de Lógica Nebulosa (SLN) [9], representados pela figura 1. Nesses sistemas, são fornecidas entradas precisas para um módulo *fuzificador* que, por sua vez, fornece parâmetros nebulosos para um motor de inferência, o qual processa a aplicação de uma regra do tipo SE-ENTÃO, constituída de proposições, envolvendo termos de variáveis lingüísticas. Após o processamento de uma regra, o valor nebuloso obtido como resposta da inferência é *desfuzificado*, obtendo-se, dessa forma, a saída precisa do sistema.

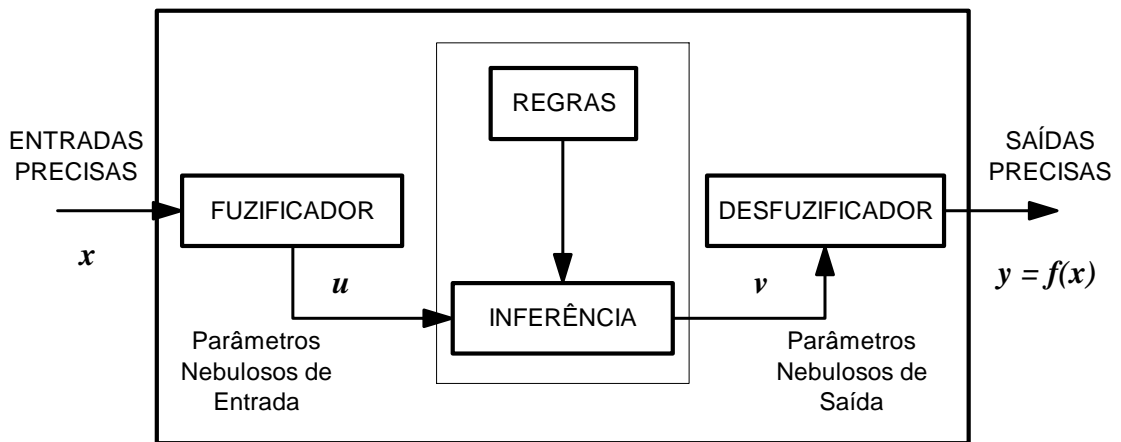


Figura II.1: Sistema de Lógica Nebulosa.

O *fuzificador* mapeia um valor preciso (*crisp*) $x \in X$ para um conjunto nebuloso \tilde{A} em X . Um parâmetro preciso pode ser *fuzificado* seguindo um método simples, trapezoidal, triangular ou gaussiano. O método mais usado é o simples que nada mais é do que criar o conjunto nebuloso \tilde{A} com sua função de pertinência assumindo valor 1 (um) no ponto de X em que o dado parâmetro tem a sua definição e, nos demais pontos, o valor da pertinência é 0 (zero).

O método simples nem sempre é adequado, principalmente quando os dados podem ser corrompidos por ruídos em processos de medidas, por essa razão foram propostos os outros três métodos mostrados nas figuras II.2, II.3 e II.4. Nos casos do triângulo e da curva gaussiana, o valor do parâmetro a ser *fuzificado* é colocado no ponto de máximo da função de pertinência (1.0), enquanto os outros pontos que se afastam desse máximo possuem valores decrescentes de pertinência até chegar a zero. No caso do trapézio, o valor a ser *fuzificado* fica no ponto médio dos máximos da função, como mostra a figura II.4. A largura

do triângulo, do trapézio e da área da gaussiana é escolhida com base no problema considerado.

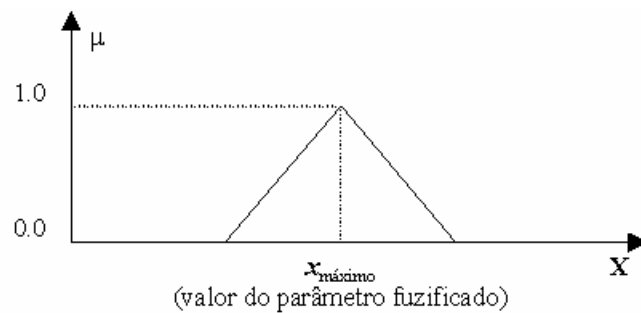


Figura II.2: Formato da função de pertinência de um conjunto nebuloso obtido pela *fuzzificação* de um parâmetro $x_{\text{máximo}}$ pelo método do triângulo.

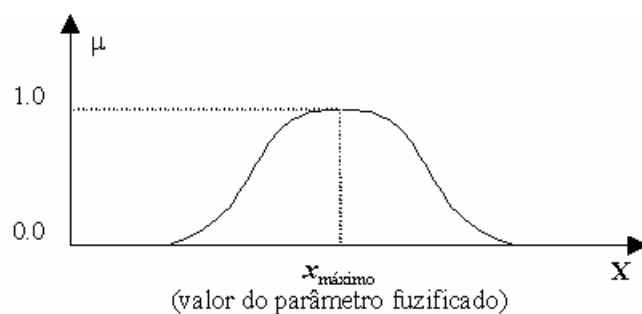


Figura II.3: Formato da função de pertinência de um conjunto nebuloso obtido pela *fuzzificação* de um parâmetro $x_{\text{máximo}}$ pelo método da gaussiana.

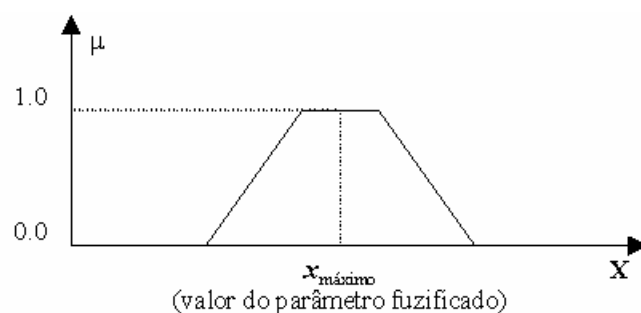


Figura II.4: Formato da função de pertinência de um conjunto nebuloso obtido pela *fuzzificação* de um parâmetro $x_{\text{máximo}}$ pelo método do trapézio.

A *desfuzzificação* pode ser feita através do valor máximo, da média de valores máximos ou por meio do cálculo do centróide. O método do máximo consiste em escolher como valor *desfuzzificado*, o ponto do universo de discurso em que a função de pertinência é máxima. Esse método provoca alguma confusão nos casos em que a função de pertinência

possui vários valores máximos, logo, foi proposto em seguida, a utilização da média dos máximos para achar o valor a ser *desfuzificado*. Todavia, numa situação como a mostrada pela figura II.5, a média dos máximos conduziria a um valor onde a função de pertinência é zero, o que não faria muito sentido.

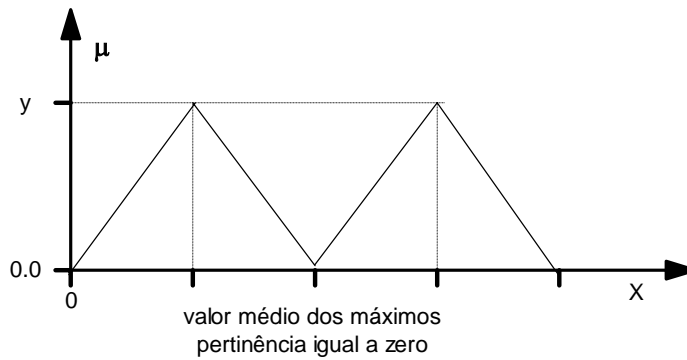


Figura II.5: Exemplo de conjunto nebuloso em que a *desfuzificação* pela média dos máximos conduziria a um valor de pertinência igual a zero.

Diante dos inconvenientes dos métodos do máximo e da média dos máximos, foi proposto em seguida, o método do centróide, que se baseia no cálculo do centro de gravidade conforme a expressão mostrada abaixo. O único inconveniente desse último método seria apenas a complexidade do cálculo que, no caso geral, é obtido de forma numérica.

$$\bar{x} = \frac{\int_S x \mu_{\tilde{A}}(x) dx}{\int_S \mu_{\tilde{A}}(x) dx} \quad (\text{Fórmula II.3})$$

Na expressão (II.3), \bar{x} é o valor *desfuzificado* de um dado parâmetro que corresponde ao centro de gravidade (centróide) do conjunto nebuloso com função de pertinência $\mu_{\tilde{A}(x)}$ e suporte S.

Uma idéia básica na Lógica Nebulosa é a de que uma proposição P em uma linguagem natural ou artificial pode ser vista como uma coleção de restrições elásticas [19], restringindo os valores de uma coleção de variáveis $x = (x_1, x_2, \dots, x_n)$ definidas nos universos U_n . Geralmente as variáveis, bem como, as restrições aparecem em P de maneira implícita. Visto nessa perspectiva, a representação do conhecimento de P consiste, em essência, no processo de tornar explícitas as variáveis e restrições. Isso é obtido, representando P através de sua forma canônica $P \rightarrow x \text{ É } A$, onde A é um predicado nebuloso.

II.2.2 Sistema de Inferência

Um sistema de lógica nebulosa típico será composto de várias regras de inferência, manipulando termos de algumas variáveis lingüísticas como no exemplo exposto a seguir:

- SE Largura (ϵ) É estreita E Trânsito (ϵ) É intenso ENTÃO Velocidade (ϵ) é muito_lenta
- SE Largura (ϵ) É estreita E Trânsito (ϵ) É livre ENTÃO Velocidade (ϵ) é rápida
- SE Largura (ϵ) É estreita E Trânsito (ϵ) É moderado ENTÃO Velocidade (ϵ) é lenta
- SE Largura (ϵ) É média E Trânsito (ϵ) É intenso ENTÃO Velocidade (ϵ) é lenta
- SE Largura (ϵ) É ampla E Trânsito (ϵ) É intenso ENTÃO Velocidade (ϵ) é rápida
- SE Largura (ϵ) É ampla E Trânsito (ϵ) É livre ENTÃO Velocidade (ϵ) é muito_rápida

As regras apresentadas anteriormente modelam um raciocínio nebuloso com o intuito de estabelecer o limite de velocidade em uma dada estrada ϵ . As variáveis lingüísticas empregadas nesse exemplo seriam:

- $T(\text{velocidade}) = \{\text{muito_lenta}, \text{lenta}, \text{rápida}, \text{muito_rápida}\}$
- $T(\text{largura}) = \{\text{ampla}, \text{média}, \text{estreita}\}$
- $T(\text{trânsito}) = \{\text{livre}, \text{moderado}, \text{intenso}\}$

Cada um dos termos dessas variáveis lingüísticas devem ter suas funções de pertinência definidas de acordo com os seus respectivos universos de discurso. A variável velocidade teria como universo de discurso, um intervalo de valores reais de 10Km/h até 120Km/h. Já a variável largura poderia ter como universo de discurso valores reais de 3m até 30m, enquanto a variável trânsito poderia estar definida em um universo de discurso fazendo referência à quantidade de cruzamentos e/ou semáforos existentes, logo, esse universo seria um intervalo inteiro de 0 até 20, por exemplo.

Os sistemas axiomáticos, construídos com lógica clássica, possuem um conjunto de implicações ou premissas (axiomas) descrevendo o conhecimento sobre um determinado assunto. Ao ser feita uma pergunta a esse conjunto de axiomas básicos, podemos deduzir uma fórmula como resposta do sistema através da aplicação de regras de inferência que atuam

sobre as fórmulas constituintes dos referidos axiomas. Uma das regras de inferência utilizadas na lógica clássica é a regra *modus ponens*. Tal regra é mostrada a seguir:

$$\frac{A \supset B}{A} \quad \text{(Fórmula II.4)}$$

Na regra *modus ponens* (II.4), tem-se a implicação SE A ENTÃO B (axioma), logo, caso se verifique A (pergunta), conclui-se B. Isso é empregado na lógica clássica, mas na lógica nebulosa é necessário estender a definição dessa regra de inferência que passa, nesse caso, a ser chamada de *modus ponens* generalizado, pois os dados da lógica nebulosa são aproximações de uma realidade. Portanto, a regra *modus ponens* generalizado ficaria da seguinte forma:

$$\frac{A \supset B}{A'} \quad \text{(Fórmula II.5)}$$

A expressão (II.5) mostra que a pergunta A' não é exatamente igual ao antecedente A do axioma ou regra, na verdade, a pergunta é “parecida” com o antecedente, o que nos leva então a concluir, que a resposta B' também será “parecida” com o conseqüente B da implicação. Um exemplo [7], empregando termos lingüísticos seria:

<p><i>PREMISSA</i> → O tomate está muito vermelho</p> <p><i>IMPLICAÇÃO</i> → Tomate vermelho é maduro</p> <p><i>CONCLUSÃO</i> → O tomate está muito maduro</p>
--

No caso de se ter várias regras com dois antecedentes, como no exemplo do sistema para achar um limite de velocidade, existem dois conjuntos nebulosos de entrada para comparar com cada um dos antecedentes das regras. No caso especial em que tais entradas fossem iguais aos termos das variáveis lingüísticas largura e trânsito, de alguma regra, então a resposta corresponderia ao conseqüente dessa mesma regra. Entretanto, o caso geral corresponde à situação em que tais entradas não são iguais aos antecedentes de nenhuma das

regras e sim uma boa aproximação dos antecedentes de alguma regra e uma má aproximação dos antecedentes das demais regras.

A aplicação da inferência *modus ponens* generalizado ao caso de várias regras é feita por intermédio de um cálculo [19] que tem início com a determinação do grau de aderência α que as entradas (pergunta) possuem com cada uma das regras. Essa aderência atenua ou não (dependendo do seu valor) a influência do conseqüente de cada regra com o resultado que, por sua vez, é uma combinação de todos os conseqüentes de todas as regras. Tal cálculo, apresentado detalhadamente em pseudo-código, segue os seguintes passos para o exemplo do limite de velocidade:

- (1) Sejam \tilde{L} e \tilde{T} dois conjuntos nebulosos representando uma largura e um trânsito de uma determinada estrada ε . Sejam também $\mu_{\tilde{L}}$ e $\mu_{\tilde{T}}$ as funções de pertinência de \tilde{L} e \tilde{T} respectivamente.
- (2) Para cada regra **Rn** Faça:
 - (2.1) Achar a função mínima entre a função de pertinência do antecedente Largura (ε) e $\mu_{\tilde{L}}$, nomeando-a de $f_{\min A1}$;
 - (2.2) Achar a função mínima entre a função de pertinência do antecedente Trânsito (ε) e $\mu_{\tilde{T}}$, nomeando-a de $f_{\min A2}$;
 - (2.3) $\alpha_n = \text{mínimo}(\text{supremo}(f_{\min A1}), \text{supremo}(f_{\min A2}))$;
 - (2.4) Achar a função mínima entre a função constante $f = \alpha_n$ e o conseqüente de **Rn**, nomeando-a de $f_{\min Cn}$;
- (3) Fim do Para Faça
- (4) Achar a função máxima entre todas as funções $f_{\min Cn}$ para n variando de 1 à quantidade de regras.

No exemplo considerado, para expor um método de inferência dos sistemas de lógica nebulosa, as duas entradas \tilde{L} e \tilde{T} seriam resultados de uma *fuzificação* de valores precisos da largura e do trânsito de uma dada estrada ε . A resposta para essas duas entradas seria um conjunto nebuloso com universo de discurso da variável lingüística velocidade e com função de pertinência obtida no passo (4). Procedendo-se então com a *desfuzificação*, de tal resposta, obter-se-ia finalmente um valor preciso de velocidade limite para a estrada ε com características de largura e trânsito \tilde{L} e \tilde{T} respectivamente.

II.3 Cartografia

Segundo a Associação cartográfica Internacional, a cartografia é definida como o conjunto de estudos e operações científicas, artísticas e técnicas, baseado nos resultados de observações diretas ou de análise de documentação, visando à elaboração de cartas, projetos e outras formas de expressão, bem como a sua utilização [20].

A terra, ao longo da história já foi apresentada de diferentes formas. No início, era considerada plana. Pitágoras e Aristóteles já a definiram esférica. Newton a definiu como elipsoidal e Gauss concluiu que a melhor forma seria a geoidal. A escolha da forma de representação da terra, na verdade leva em consideração o uso, pois, para cada aplicação uma determinada forma é considerada melhor do que a outra. A forma plana é a escolhida para uso dentro dos limites da topografia, já para cálculos astronômicos a melhor forma é a esférica. Quando o que levamos em consideração é a altitude, utilizamos o Geóide, que é uma superfície de nível esférica, onde a altitude é zero, normalmente utilizada para a projeção da superfície topográfica. O Geóide é uma superfície ondulada e não possui uma forma matemática ou geométrica definida e é usada como uma superfície de referência para altitudes. O Geóide é a superfície de nível usada para representar a forma da Terra. Ele é considerado como a superfície de nível de altitude igual a zero e coincidente com o nível médio dos mares.

Para uma medida precisa em cálculos de posicionamento, de distâncias, direções e outros elementos geométricos da cartografia, usamos a terra na forma do Elipsóide. O Elipsóide é uma figura relativamente simples e que se ajusta ao geóide com uma aproximação de primeira ordem. O elipsóide é formado a partir de uma elipse rotacionada em torno de seu semi-eixo menor (norte-sul).

II.4 Formação das Regiões para o Atendimento do Provedor

Atualmente no Brasil as ligações telefônicas são tarifadas de diversas formas segundo dados da Anatel [21]. Podemos citar:

- Áreas Conurbadas – também conhecidas como áreas metropolitanas, foram definidas através de um processo histórico, em que o desenvolvimento tecnológico foi fundamental, não havendo uma forma precisa para sua definição. Hoje existem 29 áreas conurbadas no Brasil. Para as ligações telefônicas entre localidades que estejam dentro de uma mesma área conurbada são cobradas tarifas locais. Como exemplo

podemos citar a área conurbada de Petrópolis, que é constituída pelas localidades de Petrópolis, Correias, Itaipava, Jardim Araras, Nogueira e Pedro do Rio.

- Áreas D1 – Quando possuem distâncias geodésicas inferior a 50Km, não se encontrando em uma mesma área conurbada. Para as ligações telefônicas dentro desta faixa a tarifa é em média quatro vezes maior que a tarifa de ligação local.
- Áreas D2 – Quando possuem distâncias geodésicas superior a 50Km e inferior a 100Km. Para as ligações telefônicas dentro desta faixa a tarifa é em média oito vezes maior que a tarifa de ligação local.
- Áreas D3 – Quando possuem distâncias geodésicas superior a 100Km e inferior a 300Km. Para as ligações telefônicas dentro desta faixa a tarifa é em média dez vezes maior que a tarifa de ligação local.
- Áreas D4 – Quando possuem distâncias geodésicas superior a 300Km. Para as ligações telefônicas dentro desta faixa a tarifa é em média quinze vezes maior que a tarifa de ligação local.

Diversas são as formas de conexão telefônica entre os usuários e os provedores de Internet atualmente. Algumas localidades são atendidas por serviços de acesso grátis através do uso de telefones de prefixo 1500 [22] que possuem cobrança de tarifa de ligação local. A Anatel atualmente trabalha na implantação do AICE (Acesso Individual Classe Especial) [23] para acesso a navegação na Internet, com tarifa de baixo custo.

II.4.1 O Sistema de Formação das Regiões de Atendimento Proposto

Para simplificar, este trabalho sugere e utiliza uma forma mais precisa e também capaz de atender aos interesses dos usuários, provedores e prestadoras de serviço de telefonia fixa. Usaremos o conceito de Regiões Limítrofes [24] que são aquelas compreendidas entre localidades pertencentes a regiões distintas, distantes entre si até 50Km geodésicos.

Desta forma, um acordo entre o provedor e as concessionárias das regiões afetadas

pode gerar um valor de cobrança diferenciado aos usuários do provedor que facilitará a sua implantação em um determinado município que poderá atender ainda a outros municípios, que tenham sua sede distante até 50Km da sede do município onde será implantado o provedor ou que neste trabalho é chamado de município base.

Uma ligação para acesso à Internet normalmente não é inferior a dez minutos, fazendo com que as concessionárias possam obter o seu lucro em função do tempo de acesso do usuário. As concessionárias poderiam tanto utilizar uma tarifa com desconto sobre a cobrada em uma ligação local quanto utilizar o mesmo valor das ligações locais, mas com um tempo para a contabilização dos pulsos maior, em vez dos quatro minutos normais utilizados pelas concessionárias.

II.4.2 O Cálculo da Distância Entre os Municípios

O cálculo da distância entre os municípios é fundamental ao sistema. O município escolhido, na verdade será apenas conhecido como visto anteriormente como município base.

Se a terra não fosse redonda, o cálculo da distância entre o município base e cada um dos demais seria bastante simplificado. Uma vez que possuímos as coordenadas de cada um dos municípios, bastaria uma simples triangulação para obtermos a distância entre eles.

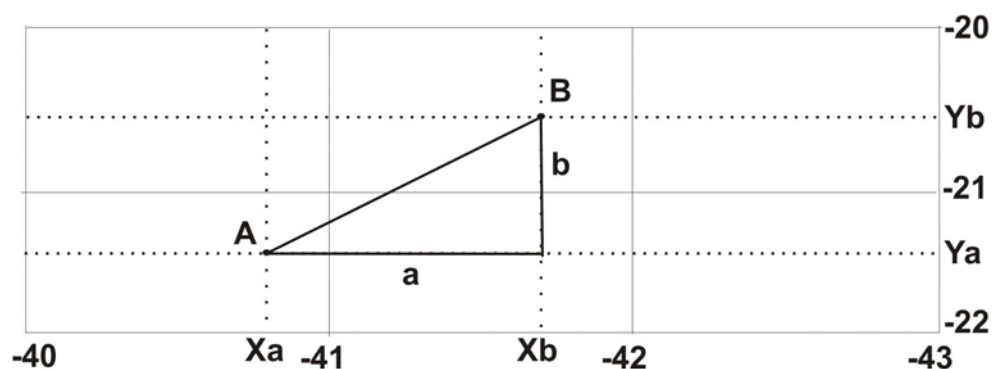


Figura II.6: Exemplo de triangulação entre dois pontos distintos.

Para obtermos a distância entre os pontos A e B, poderíamos então usar:

Pelo teorema de Pitágoras, podemos calcular a distancia entre os pontos A e B, como a hipotenusa do triângulo formado pelos catetos a e b.

$$h = \sqrt{a^2 + b^2} \quad (\text{Fórmula II.6})$$

Para obtermos os valores dos catetos a e b:

$$a = abs(X_a - X_b) \quad (\text{Fórmula II.7})$$

$$b = abs(Y_a - Y_b) \quad (\text{Fórmula II.8})$$

Assim, teríamos a distância entre os pontos A e B, onde os pontos A e B poderiam ser dois municípios distintos.

Mas, como a terra é redonda e não plana, a triangulação entre os pontos A e B não deve ser usada para o cálculo da distância entre os pontos A e B.

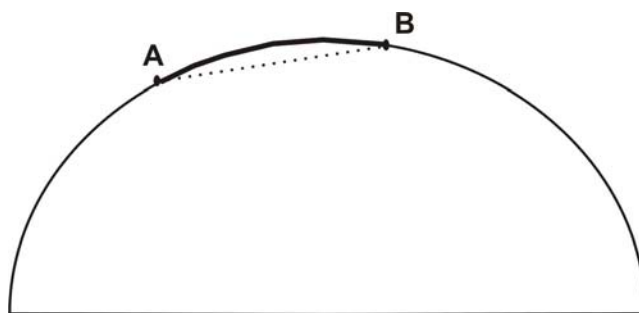


Figura II.7: Distância entre dois pontos na terra.

A distância entre os pontos A e B, não pode ser calculada por triangulação. A distância entre os pontos A e B é definida pelo arco formado entre A e B, não pela reta criada entre A e B. Uma vez que obtemos as coordenadas de latitude e longitude de cada município, essas então devem ser usadas em um sistema complexo de cálculo, que leve em consideração a curvatura da terra. Para isso, usaremos o método de Molodenski para o cálculo da distância entre dois pontos distintos da terra. Este cálculo da distância, será usado para calcularmos as distâncias entre o município base e cada um dos demais municípios. Assim, poderemos gerar uma lista dos municípios que se encontram a até 50Km de distância do município base, gerando assim uma região formada por estes municípios. Esta região é que será usada para avaliarmos a possibilidade de implantação do novo provedor.

Nosso objetivo é o cálculo da distância, obtido através da seguinte fórmula de Molodensky:

$$s = b[A \Delta\sigma + B \cos 2 \sigma_m \sin \Delta\sigma \frac{C}{2} \cos 4 \sigma_m \sin 2 \Delta\sigma + \frac{D}{3} \cos 6 \sigma_m \sin 2 \Delta\sigma] \quad (\text{Fórmula II.9})$$

A solução para o cálculo da distância entre duas coordenadas geodésicas pode ser iterativa. Para o início dos cálculos, temos que identificar o elipsóide a ser usado. Como as coordenadas dos pontos estão armazenadas de acordo com os dados obtidos no senso do IBGE, estas estão padronizadas segundo o elipsóide SAD-69 (IBGE). Podemos então definir os parâmetros do elipsóide.

Parâmetros do elipsóide (SAD-69):

Semi-eixo Maior do elipsóide : $a = 6378160$

: $df = 298.25$

Para melhor precisão, o valor de π será definido como:

$$\pi = 3.14159265358979$$

São necessários os seguintes cálculos auxiliares:

$$f = \frac{1}{df} \quad (\text{Fórmula II.10})$$

$$\text{Cálculo do Semi-eixo Menor} \quad b = a + (-af) \quad (\text{Fórmula II.11})$$

$$\text{Cálculo da Excentricidade} \quad e^2 = 2f - (f^2) \quad (\text{Fórmula II.12})$$

Dadas as latitudes e longitudes dos dois pontos, definidos por φ_a e λ_a para o ponto 1 e φ_b e λ_b para o ponto 2, com grandezas em graus, minutos e segundos, devemos então transformá-los para radianos, como no exemplo a seguir:

$$\varphi_a = \frac{\pi \varphi_a}{180} \quad (\text{Fórmula II.13})$$

$$\lambda_a = \frac{\pi \lambda_a}{180} \quad (\text{Fórmula II.14})$$

$$\varphi_b = \frac{\pi \varphi_b}{180} \quad (\text{Fórmula II.15})$$

$$\lambda_b = \frac{\pi \lambda_b}{180} \quad (\text{Fórmula II.16})$$

Para que possamos utilizar a fórmula final para o cálculo da distância (s), de acordo com as diretrizes de Molodensky, devemos efetuar uma série de cálculos intermediários e uma iteração. Os passos a seguir são capazes de nos orientar até estarmos aptos a efetuarmos o cálculo final.

- 1º passo: Cálculo das latitudes reduzidas para os pontos 1 e 2, para ψ_a e ψ_b , necessários para o cálculo de $\Delta\sigma$ no passo 2:

$$\psi_a = \arctan\left(\frac{b}{a} \tan(\varphi_a)\right) \quad (\text{Fórmula II.17})$$

$$\psi_b = \arctan\left(\frac{b}{a} \tan(\varphi_b)\right) \quad (\text{Fórmula II.18})$$

- Cálculo auxiliar da variação de Lambda $\Delta\lambda$:

$$\Delta\lambda = \lambda_b - \lambda_a \approx \Delta l_i \quad (\text{Fórmula II.19})$$

- Início da iteração, até a convergência de Δl_i em Δl_{i-1} :

REPITA:

$$\Delta l_{i-1} = \Delta l_i \quad (\text{Fórmula II.20})$$

- 2º passo: Cálculo da variação de Sigma, $\Delta\sigma$, necessário para o cálculo de ψ_0 no passo 4 e de Δl_i no passo 6:

$$\Delta\sigma = \arcsin \sqrt{(\sin \Delta l_i \cos \psi_b)^2 + (\sin \psi_b \cos \psi_a - \sin \psi_a \cos \psi_b \cos \Delta l_i)^2} \quad (\text{Fórmula II.21})$$

- 3º passo: Cálculo de α_{ab} , necessário para o cálculo de Δl_i no passo 5:

$$\alpha_{ab} = \arctan\left(\frac{\sin \Delta l_i \cos \psi_b}{\sin \psi_b \cos \psi_a - \sin \psi_a \cos \psi_b \cos \Delta l_i}\right) \quad (\text{Fórmula II.22})$$

$$\Delta longitude = \lambda_b - \lambda_a \approx \Delta\lambda \quad (\text{Fórmula II.23})$$

Para a definição do quadrante do azimute, usamos a tabela:

Tabela II.1: Determinação do sinal do azimute.

Sinal de $\Delta\lambda$	Sinal de $\text{tg } \alpha_{ab}$	Quadrante de α_{ab}
+	+	I
+	-	II
-	+	III
-	-	IV

α_{ab} deve então ser redefinido em função do seu quadrante:

SE ($\Delta\text{Longitude} > 0$ E $\text{tg } (\alpha_{ab}) < 0$) OU ($\Delta\text{Longitude} < 0$ E $\text{tg } (\alpha_{ab}) > 0$)

Então

$$\alpha_{ab} = \alpha_{ab} + \pi \quad (\text{Fórmula II.24})$$

FIM_SE;

- 4º passo - Cálculo de ψ_0 , necessário para o cálculo de Δl_i no passo 6:

$$\psi_0 = \arccos(\sin \alpha_{ab} \cos \psi_a) \quad (\text{Fórmula II.25})$$

- 5º passo - Cálculo de σ_m , necessário para o cálculo de Δl_i no passo 6:

$$\alpha_a = \arcsin(\cos \psi_a \sin \alpha_{ab}) \quad (\text{Fórmula II.26})$$

$$\sigma_a = \arcsin\left(\frac{\sin \psi_a}{\cos \alpha_e}\right) \quad (\text{Fórmula II.27})$$

$$\sigma_m = \sigma_a + \left(\frac{\Delta\sigma}{2}\right) \approx \frac{(\sigma_a + \sigma_b)}{2} \quad (\text{Fórmula II.28})$$

- 6º passo - Cálculo do novo Δl_i , utilizando os valores de $\Delta\sigma$, ψ_0 e σ_m , obtidos respectivamente nos passos 2, 4 e 5:

$$\Delta l_i = \Delta\text{longitude} + \frac{e^2}{2} \cos \psi_0 \left[A' \Delta\sigma + B' \cos 2 \sigma_m \sin \Delta\sigma + \frac{C'}{2} \cos 4 \sigma_m \sin 2 \Delta\sigma + \dots \right] \quad (\text{Fórmula II.29})$$

onde:

$$A' = 1.0 + \frac{e^2}{4} + \frac{e^4}{8} - \frac{e^2}{8} \sin^2 \psi_0 - \frac{e^4}{8} \sin^2 \psi_0 + \frac{3}{64} e^4 \sin^4 \psi_0 \quad (\text{Fórmula II.30})$$

$$B' = \frac{e^2}{8} \sin^2 \psi_0 - \frac{e^4}{8} \sin^2 \psi_0 + \frac{e^4}{16} \sin^4 \psi_0 \quad (\text{Fórmula II.31})$$

$$C' = \frac{e^4}{64} \sin^4 \psi_0 \quad (\text{Fórmula II.32})$$

- 7º passo: Uma vez obtido o valor novo de Δl_i , temos que compará-lo com o valor anteriormente calculado, armazenado em Δl_{i-1} . Se forem diferentes, ou seja, ainda não convergiram, devemos retornar ao passo 2.

ATÉ QUE: $(\Delta l_i - \Delta l_{i-1} < \varepsilon) \approx \Delta l_i = \Delta l_{i-1}$ (Fórmula II.33)

- 8º passo: Após as reiteraões convergirem, podemos finalmente calcular a distância s , após os seguintes cálculos intermediários:

$$K^2 = e^2 \sin^2 \psi_0 \quad (\text{Fórmula II.34})$$

$$A = 1.0 + \frac{k^2}{4} - \frac{3k^4}{64} + \frac{5k^6}{256} \quad (\text{Fórmula II.35})$$

$$B = -\frac{k^2}{4} + \frac{k^4}{16} - \frac{15k^6}{512} \quad (\text{Fórmula II.36})$$

$$C = -\frac{k^4}{64} + \frac{3k^6}{256} \quad (\text{Fórmula II.37})$$

$$D = -\frac{k^6}{512} \quad (\text{Fórmula II.38})$$

$$s = b[A \Delta \sigma + B \cos 2 \sigma_m \sin \Delta \sigma \frac{C}{2} \cos 4 \sigma_m \sin 2 \Delta \sigma + \frac{D}{3} \cos 6 \sigma_m \sin 2 \Delta \sigma] \quad (\text{Fórmula II.39})$$

O valor da distância (s) calculado é definido em metros, sendo necessária a divisão por 1000, para obtermos a distância final em Km.

$$\text{Distância Final (km)} = s / 1000 \quad (\text{Fórmula II.40})$$

II.5 Gerenciador de Banco de Dados

Os sistemas gerenciadores de banco de dados [25,26,27,28,29] possuem uma larga utilização dentro dos sistemas de informação. A necessidade cada vez maior de armazenamento de dados, para a posterior geração de informações, através de relatórios e/ou processamentos, nos faz recorrer às suas facilidades. Um Sistema Gerenciador de Banco de Dados (SGDB) é constituído por um conjunto de dados e um grupo de programas para acesso e recuperação destes dados. O principal objetivo de um SGBD é proporcionar um ambiente eficiente de armazenamento e recuperação de dados, oferecendo uma forma fácil de obtenção da informação.

Os sistemas gerenciadores de banco de dados são desenvolvidos para gerir grandes volumes de informações. O gerenciamento de informações implica na definição das estruturas de armazenamento, na definição dos mecanismos de manipulação dos dados para transformá-los em informação. Um SGBD deve ainda garantir a segurança dos dados armazenados, impedindo que usuários não autorizados façam acesso. Como os dados normalmente são acessados por diversos usuários simultaneamente, os SGBD's oferecem mecanismos de tratamento dos dados capazes de manter a integridade dos mesmos.

O uso de arquivos para o armazenamento dos dados implica em diversos problemas. Entre os objetivos principais de um SGBD está o de resolver esses problemas. Podemos então destacar os seguintes problemas resolvidos com o uso de um SGDB contra o uso de arquivos para o armazenamento dos dados.

- **Inconsistência e Redundância dos Dados** – Como arquivos e programas normalmente são criados e mantidos por pessoas diferentes, os arquivos possuem formatos diferentes e os programas muitas vezes são feitos em diferentes linguagens de programação. É comum que a mesma informação esteja em mais de um arquivo, aumentando o tamanho físico no disco para o armazenamento dos dados, dessa forma, tornando o acesso aos dados mais lentos e o que é pior, o programa pode fazer atualizações em apenas um arquivo que contenha os dados, deixando de atualizar as mesmas informações nos demais arquivos, o que também causa inconsistência nos dados. Atualizar os dados do endereço de um funcionário em um arquivo e não atualizar nos demais que contenham estes dados, fará com que o funcionário possua mais de um endereço, onde cada programa usará um endereço diferente e avisos de informes irão para diferentes endereços do mesmo

funcionário, dificultando a disseminação de informações pela empresa. Ao usar devidamente um SGDB, os dados são protegidos contra inconsistência e redundância.

- Dificuldade de Acesso aos Dados - Cada vez que necessitamos recuperar dados de uma nova forma, o programador é obrigado a fazer um novo programa, o que normalmente leva muito tempo. Pode ser importante em algum momento, saber quais são os clientes que mais comprou em nossa empresa, já em outro momento, pode ser importante saber quais são os clientes que moram perto de uma determinada filial. Um SGDB permite diversas formas de consulta, sem a necessidade do uso de programação específica, facilitando enormemente a recuperação dos dados das mais diversas formas possíveis, possibilitando uma recuperação extremamente rápida das informações.
- Isolamento de Dados – Como os dados podem estar dispersos em diferentes arquivos e estes podem estar em diferentes formatos, torna difícil escrever novos programas para que uma aplicação recupere de forma apropriada os dados. Se precisarmos saber as vendas para um determinado cliente em um determinado mês nos últimos três anos, o seu código estará em uma tabela e as vendas poderão estar em diferentes arquivos, será necessário escrever uma nova aplicação para solucionar este problema. Em um SGBD, os dados estão sempre armazenados em um mesmo formato e são recuperados seguindo sempre a mesma forma.
- Problemas de Integridade dos Dados – Os valores atribuídos aos dados devem satisfazer certas restrições para a manutenção da consistência desses dados. Um programa pode facilmente criticar e restringir os valores de um determinado dado, mas, isso implica em que todos os outros programas façam o mesmo, porque do contrário um valor indevido pode ser armazenado. Não podemos esquecer que restringir a entrada de um dado não é tão difícil, mas, quando se trata de uma grande quantidade de dados, isso fica difícil de gerenciar. Em um SGDB, as restrições são informadas quando da criação das tabelas e os dados só são armazenados uma vez que satisfizeram as restrições de validação, evitando que inconsistências sejam armazenadas.
- Problemas de Atomicidade – Um computador é uma máquina composta por partes mecânicas, elétricas e eletrônicas, assim como os equipamentos de comunicações através de redes, são todas sujeitas a falhas. Um sistema de arquivos pode realizar parte de uma transação, mas, não terminá-la. Assim, um caixa eletrônico pode

debitar de sua conta corrente e não lhe entregar o dinheiro. Uma transação deve ser única, ou seja, deve ser atômica de tal forma que a mesma ocorra por completo ou simplesmente não ocorra e gere uma mensagem de erro. Um SGDB deve realizar todo um conjunto de operações referentes a uma transação por completo e caso haja uma falha durante uma destas operações, todas estas devem voltar atrás, ou seja, deve desfazer as operações realizadas até o momento para essa transação e voltar ao estado inicial, como se a transação ainda não tivesse iniciado.

- Anomalias nos Acessos Concorrentes – Muitos sistemas necessitam de atualizações simultâneas nos dados, para aumento do seu desempenho e para melhoria do seu tempo de resposta. Neste tipo de ambiente, a interação entre atualizações concorrentes pode resultar em inconsistência dos dados. Um banco poderia ter um grande problema, caso um cliente com o cartão magnético retirasse todo o dinheiro de sua conta em um caixa eletrônico, enquanto o segundo titular da conta fizesse a mesma coisa, ao mesmo tempo em outro caixa. Os dois caixas poderiam receber o saldo e liberar o dinheiro ao mesmo tempo, desta maneira, o sistema faria o débito duas vezes do mesmo valor, permitindo que os correntistas retirassem mais dinheiro do que possuíam. Em um SGDB, cada transação fecharia temporariamente o acesso a esta conta e apenas liberaria o acesso, após a realização da transação por completo. Assim, caso o titular iniciasse a transação de retirada, o segundo titular só teria acesso após o término da primeira transação e neste caso, o saldo seria insuficiente para a sua retirada.
- Problemas de Segurança – Os usuários nem sempre estão autorizados a acessar todos os dados. Muitas vezes os usuários podem apenas consultar, mas não podem efetuar alterações ou até mesmo apagar registros. Em um banco, por exemplo, o caixa pode ter acesso ao seu saldo, mas pode não ter acesso às suas transações bancárias, assim como pode também não ter acesso aos dados do cadastro do cliente. Um SGBD deve prover segurança aos dados, através de restrições de acesso, impedindo que um usuário não autorizado acesse dados restritos para ele.

Por causa dos problemas acima relatados, um sistema gerenciador de banco de dados deve ser robusto em suas opções para atender a essas necessidades básicas e ágil o suficiente para efetuar tais tarefas em um curto espaço de tempo, para evitar filas de espera por uma resposta. Atualmente temos no mercado diversos SGBD's, onde cada um possui uma característica própria, sendo ela a capacidade de gerenciar enormes quantidades de dados e

usuários ou maior velocidade de resposta. Entre os mais utilizados atualmente, posso citar o Oracle, o SqlServer, o informix, DB2, entre outros.

Um SGBD é então composto por arquivos e programas inter-relacionados que permitem ao usuário o acesso para consultas e alterações desses dados. O grande benefício dos SGBD's está em proporcionar uma visão abstrata dos dados ao usuário. O SGBD oculta detalhes sobre a forma de armazenamento e manutenção dos dados, trazendo para o usuário uma forma simples de armazenar e recuperar os dados.

II.6 Sistemas de Informações Geográficas (SIG)

Sistemas de Informações Geográficas ou Sistemas de Informações Georeferenciadas [1,2,3] são normalmente usados para denominar a tecnologia que possui ferramental necessário para realizar análises com dados espaciais. Forma ao lado do Processamento Digital de Imagem (PDI) e da Geoestatística o chamado universo da Geotecnologia.

Um SIG é um sistema de computador capaz de fazer montagem, armazenamento e manipulação de informações referenciadas geograficamente.

O projeto de um Sistema de Informação Geográfica, em geral, lida com diversos quesitos dentre os quais podem-se destacar:

- Relacionamento entre informações de fontes diferentes;
- Aquisição de dados;
- Integração de dados;
- Projeções e registros;
- Estruturas de dados;
- Modelagem de dados.

O universo dos SIG's teve um aumento substancial de estudiosos e pesquisadores a partir da década de 80, onde universidades e empresas investiram numerosos esforços em busca de atender os mais diversos objetivos.

Para um melhor entendimento desta tecnologia, são necessários conhecimentos distintos que englobam tanto as ciências que estudam a terra, como conhecimentos de informática. Alguns conceitos passam a ser importantes e devem ser levados em conta, a partir do momento em que unimos essas ciências.

Dados Espaciais são elementos definidos pelas variáveis x , y e z , possuem localização no espaço e estão relacionados a determinados Sistemas de Coordenadas (um exemplo: Projeção de Mercator [3], latitude-longitude) e podem estar associadas a infinitos atributos ou características. Estão distribuídos pela superfície curva da terra. Já desenvolvemos muitas técnicas para projeção de mapas, mas, possuímos muito pouco conhecimento sobre métodos de análises de dados sobre uma esfera e ainda menos sobre como modelar processos sobre uma superfície curva. Os dados referentes ao mundo real podem ser divididos em três diferentes maneiras de se observar os fenômenos que ocorrem na superfície da terra. Cada maneira então descreve um tipo de dado espacial, divididos em três tipos:

- Espacial – quando a variação muda de lugar para lugar (declividade, altitude, profundidade do solo);
- Temporal – quando a variação muda com o tempo (densidade demográfica, ocupação do solo);
- Temáticos – quando as variações são detectadas através de mudanças de características (geologia, cobertura vegetal).

Dado e Informação – As descrições dos fenômenos relacionados ao mundo real podem ser arquivadas em duas formas distintas:

- Dado – corresponde a um conjunto de valores numéricos ou não, que corresponde à descrição de fatos do mundo real;
- Informação – é um conjunto de dados que possui um significado para um uso determinado ou aplicação em particular, ou seja, é o fato de termos agregado alguma forma de interpretação ao dado puro.

Um determinado dado espacial pode estar relacionado, além de sua forma de representação, que pode ser um ponto, um nó, uma linha ou arco, cadeias e polígonos, a outras informações, tais como:

- Suas Coordenadas;
- O Sistema de Coordenadas a qual suas coordenadas pertencem;
- Temperatura;

- Tipo de rocha;
- Tipo de solo;
- Vegetação;
- Profundidade;
- Etc.

Outra característica importante está no fato de que uma vez definida uma forma para representação de um dado espacial, esta forma pode se modificar ao mudarmos a escala. Uma área em uma determinada escala pode ser representada como um ponto em outra.

Os dados espaciais, mesmo definidos em diferentes escalas e resoluções, podem ser armazenados em duas formas:

- Analógica – as formas dos objetos são representadas em papel, formando assim os mapas;
- Digital - é a forma eletrônica, armazenadas no computador na forma binária. Podem ser divididas em duas maneiras de armazenamento:
 - Vector – forma vetorial, onde são definidos vetores que descreverão as formas e cores dos objetos no momento de sua representação no vídeo.
 - Raster – maneira matricial, onde descreverão os objetos através de matrizes de pontos, que através do uso de cores, descreverão as formas dos objetos no momento de sua representação no vídeo.

Os SIG's, para atingirem os objetivos esperados, necessitam de vários campos do conhecimento humano, entre eles podemos destacar:

- Ciência da Computação, que fornece os meios para capturar, manipular, armazenar e exibir os dados, sejam eles espaciais ou não;
- Gerenciamento de informações, que fornece ferramentas capazes de manipular bancos de dados de grandes proporções e as relações matemáticas que os unem;
- Cartografia, que fornece conjuntos de operações científicas, artísticas e técnicas obtidas a partir de resultados de operações diretas ou de exploração de documentação, tendo em vista a elaboração de cartas e plantas;
- Geodésia, que fornece técnicas de matemática aplicada, observa e mede tanto a forma, como o tamanho da terra, ou parte dela e determina a exata localização dos

pontos sobre sua superfície;

- Fotogrametria, que fornece formas de produção de medidas confiáveis através da fotografia;
- Topografia, que fornece técnicas para determinar a área de qualquer porção da superfície terrestre, além do comprimento e direção de linhas e o contorno de superfícies;
- Processamento Digital de Imagens, que fornece técnicas que tem como objetivos principais remover vários tipos de degradações e distorções inerentes aos processos de aquisição, transmissão e visualização das imagens coletadas, facilitando a extração de informações [30];
- Geografia, que fornece as disciplinas que estudam um ou mais objetos ou fenômenos em particular, que observam coisas através do tempo e que observam características dentro de seus contextos espaciais.

Sistema é o conjunto de elementos entre os quais existe uma relação. Informações correspondem à derivação da interpretação de dados aos quais são representações simbólicas de certas características [31]. Um Sistema de informações é uma cadeia de operações que nos remete a planejar a observação e a coleção de dados para armazená-los, analisá-los e usar as informações derivadas em algum processo de tomada de decisão [32].

Muitas são as definições de SIG, mas, podemos citar as mais comumente encontradas:

[33]: “um caso especial de sistemas de informações, no qual o banco de dados consiste em informações sobre características distribuídas espacialmente, atividades ou eventos, os quais são definidos no espaço como pontos, linhas ou áreas. Os SIGs manipulam os dados acerca destes pontos, linhas e áreas para estabelecer perguntas *ad hoc* e análises.”

[34]: “um elenco de funções automáticas que fornece aos profissionais, com avançada capacidade, o armazenamento, recuperação, manipulação e exibição de dados geograficamente localizados”.

[35]: “um poderoso elenco de ferramentas para colecionar, armazenar, recuperar, transformar e exibir dados espaciais referenciados ao mundo real”.

[36]: “uma forma de sistemas de gerenciamento de informações que permite exibir mapas de informações gerais”.

[37]: “um sistema basicamente preocupado em mais descrever a Terra do que analisá-la ou, se preferir, é a tradicional geografia do século XIX reinventada e vestida com a tecnologia digital do século XX”.

[38]: “um sistema para capturar, armazenar, checar, manipular, analisar e exibir dados, os quais são espacialmente referenciados à Terra”.

[39]: “um sistema de base de dados no qual a maioria dos dados está indexada espacialmente e sobre as quais um elenco de procedimentos é desencadeado com a finalidade de responder a perguntas sobre entidades espaciais”.

[40]: “uma tecnologia de informações que armazena, analisa e exibe tanto dados espaciais, quanto dados não espaciais”.

[41]: “um sistema que garante decisões envolvendo a integração de dados referenciados espacialmente em um ambiente específico”.

[42]: “um sistema combinado de computadores (equipamentos e aplicativos) e procedimentos, configurados para capturar, gerenciar, manipular, analisar, modelar e exibir dados espacialmente referenciados, para resolver problemas complexos de planejamento e gerenciamento”.

[43]: “um sistema que contém dados espacialmente referenciados que possam ser analisados e convertidos em informações para uso em um conjunto específico de finalidades. A característica principal de um SIG é analisar dados para gerar novas informações”.

[44]: “um SIG é qualquer sistema de gerenciamento de informações capaz de: coletar, armazenar e recuperar informações baseadas nas suas localizações espaciais; identificar locais dentro de um ambiente que tenha sido selecionado a partir de determinados critérios; explorar relações entre os dados de um certo ambiente; analisar os dados espaciais para subsidiar os critérios de formulação de decisões; facilitar a exportação de modelos analíticos capazes de avaliar alternativas de impactos no meio ambiente; exibir e selecionar áreas tanto graficamente como numericamente, antes e/ou depois das análises”.

[45]: “qualquer conjunto de procedimentos manual ou computacional usado para armazenar e manipular dados geograficamente referenciados”.

[46]: “uma entidade institucional, refletindo uma estrutura organizacional que integra a tecnologia com um banco de dados, especialistas e um contínuo suporte financeiro”.

[47]: “um sistema com capacitação avançada de geomodelamento”.

[48]: “um sistema que surgiu como um meio de reunir e analisar diversos dados espaciais”.

[49]: “é um banco de dados contendo uma discreta representação da realidade geográfica na forma estática de objetos geométricos, em duas dimensões, com seus atributos ou dados não espaciais associados, com uma funcionalidade grandemente limitada pelas operações geométricas primitivas para criar novos objetos ou para computar as relações entre objetos, ou para simples interrogáveis e descrições sumárias”.

Como podemos observar nas várias definições acima, que possuem muitos pontos em comum, posso então dizer que na minha concepção os SIG's são sistemas de informações aonde muitos de seus dados são geo-referenciados, usando na maioria das vezes os sistemas de computação para manipular a enorme massa de dados e gerar um conjunto de informações capazes de descrever uma dada região, segundo um determinado critério.

II.7 Desenvolvimento para a Web

O desenvolvimento de aplicações para a *Web* [26,50,51,52] pode seguir vários caminhos distintos. Podemos usar diversas tecnologias diferentes e muitas são as linguagens de *script's* disponíveis. Uma solução bastante versátil usa toda a potencialidade de uma linguagem de programação, em conjunto com uma linguagem de *script*. Uma solução bastante difundida é o uso do chamado *ActiveX*. Ao usarmos esta abordagem, temos a facilidade de uso da linguagem HTML, que pode ser executada em qualquer *browser*, com o uso de objetos criados em uma linguagem de programação. Deste modo, podemos criar uma aplicação para a *Web*, utilizando todos os benefícios de uma linguagem de programação e não apenas os

benefícios de uma linguagem de *script*.

O ponto alto desta tecnologia está no uso da linguagem HTML, para criação das interfaces com o usuário e para o uso da aplicação e de uma linguagem de programação para efetuar o processamento. Esta tecnologia necessita de uma linguagem de *script*, como interface entre a linguagem HTML e a linguagem de programação. A Microsoft fornece uma solução neste nível, através dos serviços IIS (*Internet Information Server*). Um servidor com esse serviço pode gerar para o usuário respostas dinâmicas, através do uso da linguagem de *script's* ASP (*Active Server Pages*), trabalhando com a arquitetura Cliente/Servidor. Tais páginas consistem em arquivos de extensão **.asp** que contêm combinações de *Server-Side scripts* e *tags* HTML.

As aplicações ASP podem instanciar objetos criados em alguma linguagem de programação e usar os métodos desses objetos. Assim, podemos criar uma DLL (*Dinamic Link Library*) e usá-la para efetuar o processamento com entrada de dados oriundos de uma página HTML. O processamento pode efetuar toda e qualquer forma de consulta disponíveis aos mais diversos gerenciadores de banco de dados encontrados no mercado. O processamento deve então ser executado em máquinas que possuam o IIS (*Internet Information Server* da Microsoft), para a geração de uma página HTML de resposta..

Todo o código de programação existente em páginas **Asp** é executado no servidor, e esse retorna ao cliente respostas em HTML montadas de acordo com a programação pré-definida para que as aplicações Asp possam ser acessadas por qualquer *browser* existente no mercado. Uma aplicação feita em Asp pode ainda conter linhas de *Client-Side script*, que serão executados na estação cliente. Essas páginas devem estar hospedadas em um servidor Microsoft *Internet Information Server*.

Aplicações cliente-servidor são definidas como uma divisão de processos entre estações clientes e servidores. A principal finalidade é a de buscar uma melhor performance, diminuindo assim o tempo de resposta e melhorando a facilidade no momento de efetuar a manutenção da aplicação. Este tipo de aplicação é dividido em duas partes. Uma parte é executada na máquina cliente e é chamada de *Cliente Side* e a outra é executada na máquina servidora, denominada *Server Side*.

Na máquina cliente são executados os chamados *Client Side scripts*. Esses se constituem de instruções de programa em linguagem de *script*, que é processado pela estação cliente. Nas aplicações desenvolvidas para a Internet, o código que é executado na máquina cliente não tem um processamento muito grande, normalmente esse cuida apenas da verificação de informações de vídeo e de validações durante o processo de entrada de dados.

Como estamos tratando de páginas para a *web*, os *client-side scripts* são processados por um *browser*. Um grande cuidado ao se utilizar este tipo de aplicação está no fato de existirem incompatibilidades na interpretação da linguagem de *scripts* entre os diferentes *browsers*. Como exemplo, podemos citar o Microsoft Internet Explorer, que é capaz de interpretar com facilidade o Visual Basic Script, outros *browser's*, como o Netscape não o fazem sem o auxílio de um *plug in*, que normalmente é desenvolvido por outros programadores, diferentes dos que desenvolveram o *browser*. Outro problema que podemos encontrar nos navegadores (*browser's*) está ligado às versões, que muitas vezes são antigas e não foram atualizadas. Isto impede que os navegadores possam interpretar convenientemente o *script* na máquina cliente.

Na máquina servidora é executado o chamado *Server Side scripts*. Estes se constituem de instruções de programa em linguagem de *script*, que são processados pela máquina servidora. Assim, não precisamos nos preocupar em que linguagem o código foi criado, o servidor é quem se encarrega de interpretá-lo e de devolver uma resposta formatada para o cliente. Em páginas Asp, são esses códigos os maiores responsáveis pelos resultados apresentados como resposta ao usuário e a linguagem *default* utilizada é o Visual Basic Script.

O Visual Basic Script ou simplesmente VBScript é uma das muitas possibilidades de linguagem *Script* que rodam em um servidor e, para o caso específico do IIS (*Internet Information Server*), ela é a linguagem *default* (padrão).

O VBScript é uma linguagem de *script's* similar ao VBA (*Visual Basic Application*), que por sua vez é baseado na linguagem de programação Visual Basic. Entre outras facilidades, o VBScript nos permite:

- A manipulação de dados como *string's*, datas e números;
- A utilização de quase todos os comandos do Visual Basic;
- A manipulação de banco de dados, que é feita através do objeto ADO da Microsoft. O VBScript cria instâncias deste objeto para que possamos manipular tabelas nos mais diferentes gerenciadores de Banco de Dados disponíveis no mercado;
- A comunicação com servidores de objetos COM. Entre eles podemos citar o Microsoft *Exchange Server*, Microsoft *Index Server* e o *Database Server*.

Uma página baseada em *script's* desenvolvida em VBScript é executada nativamente por um grande número de *browser's*. Caso um *browser* não tenha suporte à VBScript, podemos recorrer à instalação de um *plug in*.

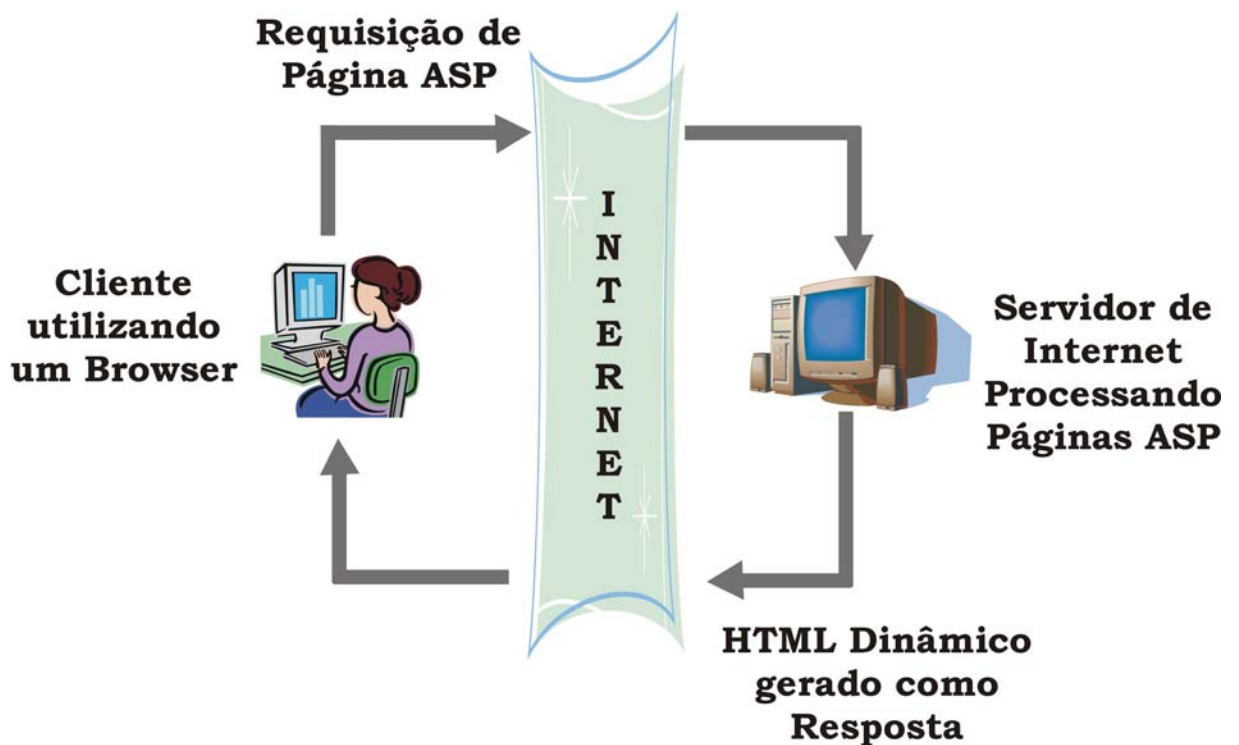


Figura II.8: Esquema de funcionamento de uma página Asp

Os arquivos escritos em Asp, são arquivos de texto comum (ASCII) e podem ser escritos em editores de texto comuns. Podemos usar o Notepad, por exemplo. Existem inúmeras ferramentas prontas capazes de ajudar no desenvolvimento de *script's* em Asp, entre elas, podemos citar o FrontPage, HomeSite, Visual InterDev, o DreamWiver, entre outras.

Para o funcionamento do esquema de execução de uma página Asp, necessitamos hospedar nossas páginas em um servidor *Web* da Microsoft, através do Windows NT, 2000 Server ou superior, com o IIS (*Internet Information Server*) instalado.

II.8 Comentários

Neste capítulo, procuramos expor, de maneira sintética, os principais conceitos da lógica nebulosa, que foram empregados no desenvolvimento do sistema. Expomos também a forma de agrupamento dos municípios, gerando as regiões de atuação para o provedor com os conceitos de cartografia necessários. Da mesma maneira foram apresentadas as fórmulas para o cálculo da distância entre dois pontos no globo terrestre e o uso dos Sistemas Gerenciadores de Banco de Dados (SGBD). Os conceitos necessários para o desenvolvimento de um Sistema de Informação geográfica (SIG) e as características do desenvolvimento de aplicações para a *Web* (www) igualmente foram abordados.

No próximo capítulo, teremos então a explanação propriamente do uso das diversas técnicas aqui descritas, de suas características e necessidades a serem empregadas no projeto.

CAPÍTULO III

Organização do SIGAI-API

III.1 Introdução

Neste capítulo, teremos uma explanação sobre as diversas técnicas empregadas no projeto e suas necessidades. Na seção III.2, veremos o uso da linguagem de hipertexto HTML para a interface com o usuário, tanto para a operação da aplicação, quanto para a apresentação da resposta ao usuário.

Na seção III.3, será descrito o uso da linguagem de *script's Active Server Pages* (ASP), necessária para a troca de dados entre a linguagem de hipertexto (HTML) e o objeto *ActiveX* desenvolvido para o sistema. Na seção III.4, será apresentado o funcionamento lógico do SIGAI-API e serão definidas as interfaces do usuário para a manipulação do sistema. São ainda apresentadas as interfaces do usuário, para entrada de dados e o recebimento da resposta gerada pelo sistema. São também mostradas as telas e os algoritmos necessários para as interfaces tanto em HTML, assim como em ASP para a instanciação do objeto *ActiveX*.

Na seção III.5, teremos a criação da DLL em *Visual Basic*, onde temos a aplicação em si, pois todas as técnicas empregadas foram produzidas na linguagem *Visual Basic* e a partir daí foi criado o objeto *ActiveX*, capaz de receber a requisição de entrada e gerar a resposta para o usuário. Na seção III.6, será definida a base de dados e serão descritas as tabelas da base de dados para uso do sistema. Nessa seção também serão informadas as formas de obtenção dos dados de cada tabela.

III.2 O uso do hipertexto (HTML)

Atualmente, uma máquina que não possua acesso a Internet, é uma máquina de pouco uso e limitada. É cada vez maior o número de usuários nesta grande rede, gerando inclusive um grande número de pessoas que não conseguem passar um longo período de tempo sem navegar pela Internet. Desta forma, o número de usuários é muito grande e como seu uso é muito difundido, o uso de navegadores (*browser's*) se tornou elementar. Sistemas que estão disponíveis na Internet facilitam a vida dos usuários. É normal hoje em dia, que

uma pessoa se esqueça do pagamento de uma conta. Muitas vezes esta conta não está em nosso poder, mas, podemos efetuar uma ligação telefônica para conseguirmos as informações necessárias e a partir daí, basta conseguirmos uma máquina com acesso à Internet e efetuarmos o pagamento em qualquer lugar do mundo. Também já é normal que esses pagamentos sejam feitos a partir de casa, sem necessitarmos de deslocamentos e mais ainda, os horários para pagamentos com *Internet banking* são mais flexíveis.

A funcionalidade de um sistema depende muito da interação com o usuário. É fundamental que o sistema seja intuitivo e não deixe o usuário em dúvida. O uso da linguagem de hipertexto facilita a criação das interfaces e deixa o usuário mais à vontade. Além disso, o projeto tem como principal característica, o uso da Internet. Assim, o uso da HTML para a criação das interfaces com o usuário, não só é intuitiva e de fácil manipulação, ela também permite que a aplicação seja usada em qualquer parte do mundo, sem a necessidade de instalações de *plug in's* ou qualquer outro *software*. O usuário cadastrado passa a usar o seu *login* e senha para obter acesso ao sistema.

Uma vez que o usuário tenha seu *login* e sua senha validados pelo sistema, serão ainda verificadas as suas permissões, para que esse possa utilizar o sistema. Após a tela de acesso, o usuário poderá escolher no mapa do estado do Rio de Janeiro, o município pelo qual tem interesse em montar um novo provedor de Internet, apenas clicando sobre o mesmo. Uma vez que o município foi escolhido, o usuário deverá aguardar a resposta do sistema, que informará a possibilidade de instalação de um novo provedor de Internet na região definida pelo município base.

III.3 O uso do ASP

O presente projeto baseia-se no desenvolvimento para a *Web*, mas, a linguagem HTML não tem como característica a manipulação de bases de dados, instanciação de objetos e tão pouco é capaz de realizar cálculos matemáticos avançados como os necessários ao sistema. Como a linguagem HTML é tida como estática, ela também não é capaz de gerar respostas, apenas pode apresentar respostas já previamente preparadas. Deste modo, somos obrigados a recorrer ao chamado HTML dinâmico (DHTML), que é composta pelo emprego da linguagem HTML tradicional com uma outra linguagem, no caso uma linguagem de *script*. O ASP (*Active Server Pages*) foi escolhido pela sua compatibilidade, similaridade e facilidade de uso em conjunto com a linguagem de programação *Visual Basic*, definida para o desenvolvimento da DLL responsável pela manipulação dos dados e processamentos.

A linguagem de *script's* ASP é capaz de manipular os dados, mas no caso deste projeto esta se torna pouco eficaz para o processamento dos cálculos e da criação do motor de inferência do sistema. A linguagem ASP foi então usada para servir de interface entre a linguagem HTML usada na confecção das interfaces com o usuário e os sistemas de manipulação de dados, de inferência *Fuzzy* e de cálculos cartográficos criados em *Visual Basic*, uma vez que a HTML não é capaz de instanciar objetos, diferentemente do ASP, que possui métodos próprios com essa finalidade. Outra necessidade está no fato de o HTML não ser capaz apenas de apresentar páginas prontas, o que não é suficiente para nós, pois, o sistema gerará respostas distintas para cada escolha do usuário e o ASP é capaz de receber uma resposta formatada em HTML e exibi-la ao usuário em seu *Browser*.

O projeto não está baseado na linguagem ASP, apenas a utiliza como ponte entre as linguagens HTML e *Visual Basic*. Em uma primeira abordagem o projeto chegou a utilizar para este fim a linguagem de *script's* *Cold Fusion* (concorrente da linguagem ASP), mas, em virtude da necessidade de instalação de um serviço extra, optou-se por retirá-lo, pois, ainda seria necessário o uso do IIS (*Internet Information Server*) de qualquer forma. Deste modo, eliminou-se um serviço e se diminuiu a dependência do sistema por outras aplicações.

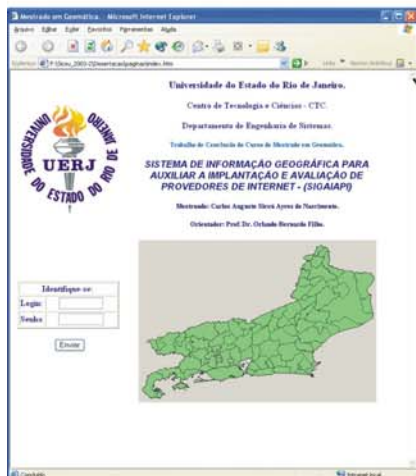
III.4 Descrição Lógica do Sistema

O sistema possui apenas três interfaces com o usuário, sendo duas para entrada e uma para saída. As duas primeiras são usadas para acesso ao sistema, onde o usuário deve se identificar para que seja validado e para o uso propriamente dito, onde o usuário faz a escolha do município base.

III.4.1 Funcionamento Básico do Sistema

O funcionamento básico do SIGAI-API é descrito pela figura III.1. Nela podemos observar que a primeira página (Interface 1) serve de acesso ao sistema, sendo restrita ao *login* do usuário. Com o usuário validado, passamos à página principal (Interface 2), responsável pelas consultas a DLL gerada em *Visual Basic*, responsável por todo o processamento da inferência, manipulação e geração da região. Por último, é exibida a página de resposta (Interface 3), onde o sistema mostra as respostas obtidas no processamento.

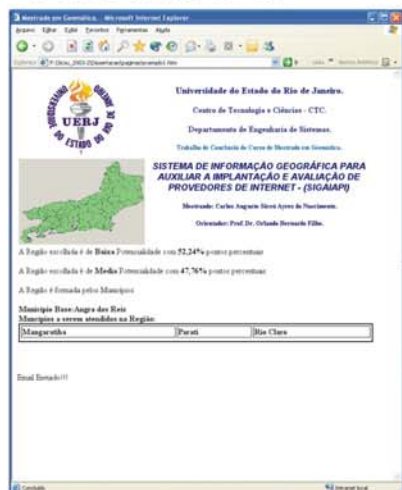
Interface 1



Interface 2



Interface 3



Sistema de
Verificação
de Usuário
(clsAcesso)

Motor de
Inferência e
Gerador de
Região
(clsFuzzy)

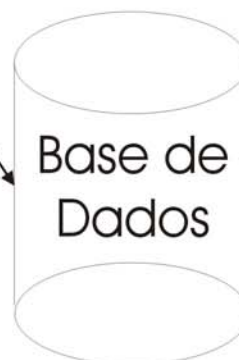


Figura III.1: Funcionamento básico do sistema.

III.4.2 Definição das Interfaces com o Usuário

Como visto antes, o sistema possui apenas três interfaces com o usuário, sendo elas:

III.4.2.1 A página *Index.html*

A primeira página do sistema é responsável pelo acesso do usuário (*index.html*). Nesta página o usuário deve informar seu nome de *login* e senha, para que o sistema possa então validar seus dados e liberar ou não o seu uso. Os dados inseridos nos campos *login* e senha são então passados à próxima página (*fuzzy.asp*) que é responsável pela verificação dos dados do usuário.

Descrição do funcionamento da página:

Index.html

Receber nomelogin e senha

Enviar nomelogin e senha para página Fuzzy.ASP

Fim-Index.html

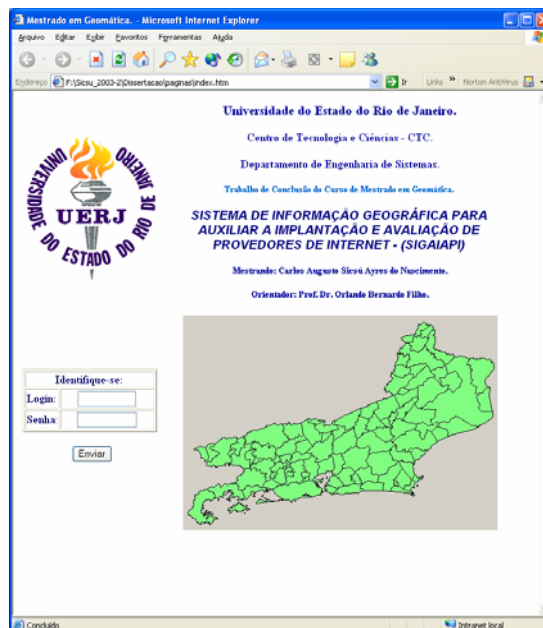


Figura III.2: Página de Login, para acesso ao sistema.

III.4.2.2 A página *Index.asp*

Nesta segunda página, ou Página principal do sistema, três interfaces diferentes podem aparecer. Através de programação, é selecionada a aparência da janela. Caso o usuário não seja válido, podemos ver a resposta do sistema pela figura III.3, informado que o usuário

não foi identificado. Já na figura III.4, temos a resposta do sistema informando que o usuário teve sua conta bloqueada e que o mesmo não pode acessar o SIGAIAPÍ.

Descrição do funcionamento da página:

Index.asp

Receber nomellogin e senha da página index.html

Verificar o status do usuário

Se usuário inválido

Então

Exibir janela formato usuário inválido

Senão

Se usuário Não-Autorizado

Então

Exibir janela formato usuário Não-Autorizado

Senão

Exibir janela formato interface de uso

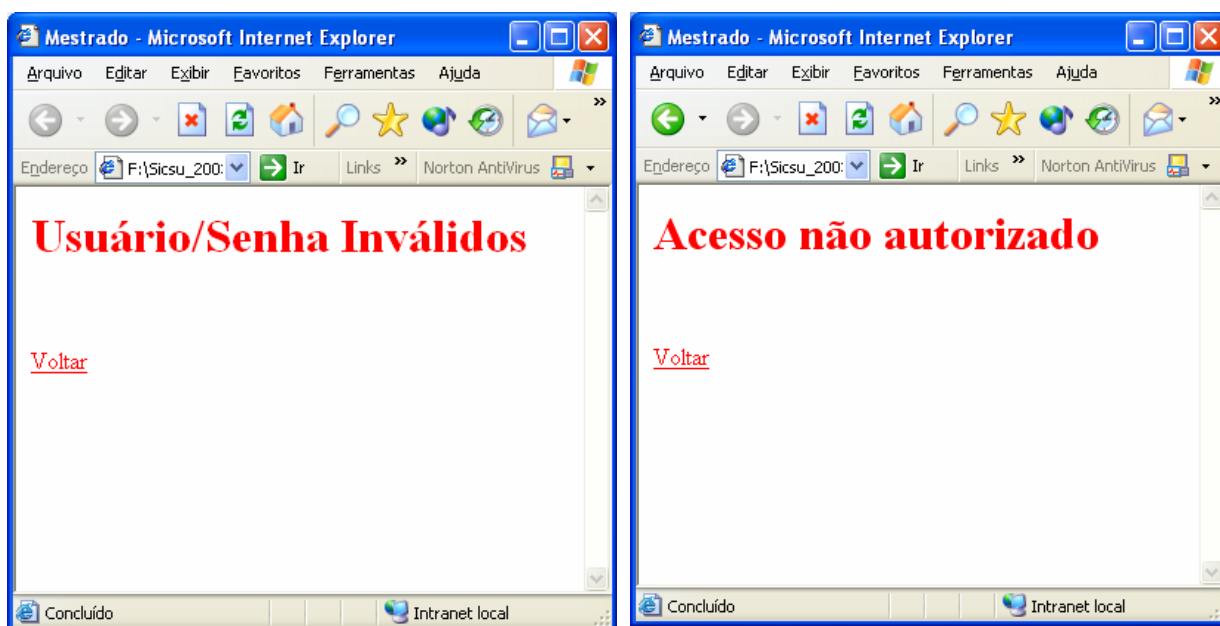
Receber escolha do usuário

Chamar a página Fuzzy.asp, passando o município escolhido

Fim-se;

Fim-se;

Fim-Index.asp



Figuras III.3 e III.4: Página de usuário inválido e página de usuário não autorizado.

Caso o usuário tenha sido validado, é apresentada a interface principal do sistema, onde o usuário poderá então escolher o município base, conforme as figuras III.5, III.6 e III.7. Esta configuração da página é grande em função da quantidade de municípios, necessitando assim efetuar um *scroll* na janela para podermos observar todos e fazermos nossa escolha.

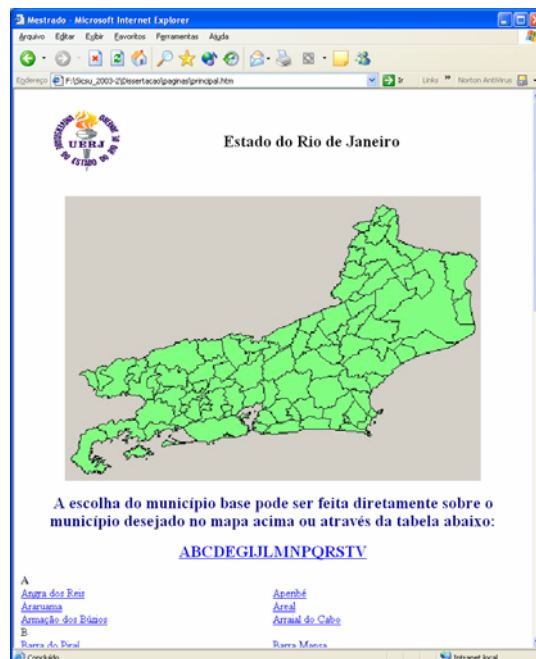
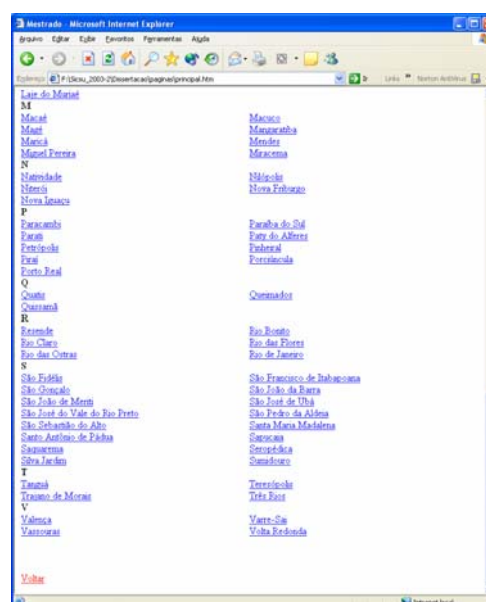


Figura III.5: Página de escolha do município base.



Figuras III.6 e III.7: Continuações da página de escolha do município base.

III.4.2.3 A página Fuzzy.asp

A última página é a de resposta à consulta do usuário. Esta página envia a solicitação do usuário ao sistema, instanciando a classe `clsFuzzy` e aguarda a resposta gerada para exibi-la, conforme a figura III.8. A resposta é montada on-line por programação dentro da classe `clsFuzzy`. Esta página exibe todas as informações resultantes do processamento. É mostrado o município base escolhido pelo usuário, os municípios que farão parte da região a ser atendida pelo provedor. O resultado da inferência é também mostrado, informando as

potencialidades da região gerada a partir do município base, informado ao sistema.

Descrição do funcionamento da página:

Fuzzy.asp

Instanciar objeto Fuzzy.clsFuzzy passando o município escolhido

Exibir resposta

Fim-Fuzzy.asp

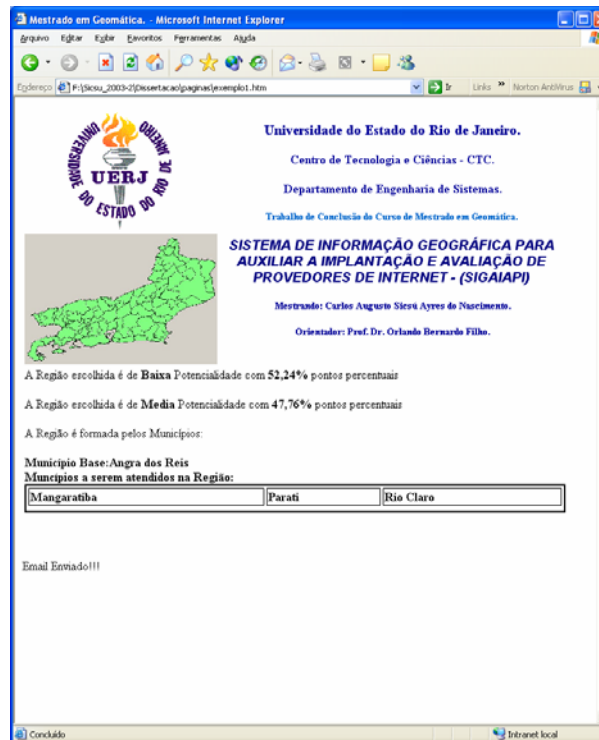


Figura III.8: Exemplo de resposta processada pelo sistema.

III.5 A Criação da DLL (*Dynamic Link Library*) Fuzzy

O projeto em questão baseia-se na *Web* e dessa forma precisamos integrar o ambiente de programação em *Visual Basic* com a Internet. Um sistema desenvolvido em linguagem de programação de alto nível não pode ser utilizado diretamente na Internet. Uma boa opção para esta integração é o uso do *ActiveX*. Assim, em vez de gerarmos um programa executável, geramos a DLL (*Dynamic Link Library*, biblioteca de vínculo dinâmico) chamada Fuzzy, para o *Windows*® e uma vez que esta DLL é registrada no sistema operacional, ela passa a estar à disposição de qualquer aplicação que queira usar seus métodos. Um aspecto que não podemos esquecer é que aplicações geradas como DLL não possuem interfaces com o usuário. A lógica de funcionamento dos módulos pode ser observado na figura III.9.

As DLL's podem apenas receber parâmetros e retornar respostas. A DLL criada

foi dividida em duas classes:

- clsAcesso, responsável apenas pela validação do usuário, retornando se o mesmo é válido, inválido ou não possui permissão.
- clsFuzzy, responsável por todo processamento do sistema em si.

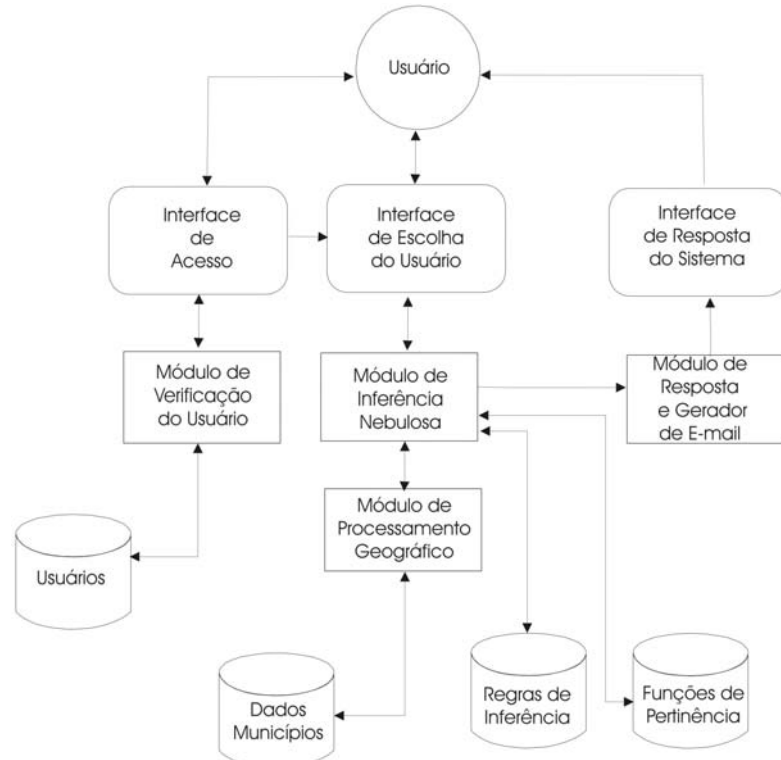


Figura III.9: Arquitetura do sistema.

III.5.1 A Classe clsAcesso

A classe clsAcesso recebe o nome de *login* do usuário e sua senha, através da Interface de Acesso (*Index.html*). É feita então uma pesquisa na tabela de usuários, através do módulo de verificação do usuário, que verifica se esse é válido e se sua senha confere. Verifica também o nível de acesso ao sistema, que se for igual a zero, indica que o usuário não está autorizado, valor maior que zero indica que o usuário pode utilizar o sistema.

Após essas verificações, é então retornada uma das respostas:

- Usuário inválido (não cadastrado);
- Usuário não autorizado (nível de autorização igual a zero);
- Usuário Válido (cadastrado e com autorização superior a zero).

Uma vez gerada a resposta do módulo de verificação do usuário, esta então é

repassada à interface de escolha do usuário, que possui um mecanismo de decisão capaz de apresentar uma tela entre três opções:

- Tela de usuário inválido, não dando possibilidade de continuação ao uso do sistema;
- Tela de usuário não autorizado, que também não dá possibilidade de continuação ao uso do sistema;
- Tela de escolha do usuário, que é a Interface de Escolha do Usuário.

III.5.2 A Classe clsFuzzy

A classe clsFuzzy é responsável pelos Módulos de Inferência Nebulosa, de Processamento Geográfico, de Resposta e Gerador de *E-mail*. Esta classe recebe o código do município base, a conexão com o banco de dados a ser usada e o nome do usuário. Através da função início, o sistema é controlado para gerar a resposta ao usuário.

Após o usuário ser validado, este deve escolher um município como base da região a ser atendida, através da Interface de Escolha do Usuário (*Index.asp*). O código do município base, a conexão com o banco de dados e o nome de *login* do usuário são passados para a Interface de resposta do Sistema, que instancia novamente a DLL, mas chamando o método Início. Com esses dados o sistema pode então realizar a inferência e retornar a resposta, além de enviar tal resposta por *e-mail* ao usuário.

O sistema irá então iniciar o processo de inferência. Neste processo, primeiro precisamos obter os dados de entrada a serem *fuzzyficados*. Estes dados são obtidos juntamente com a montagem da região proposta, onde teremos o agrupamento dos municípios que se encontram a até 50Km de distância do município base. Para montarmos a região, devemos calcular a distância do município base com os demais municípios do Estado do Rio de Janeiro e ao mesmo tempo acumularmos os dados parciais de entrada do sistema.

A obtenção desses dados segue a seguinte regra:

- Gerar a região, onde precisamos identificar todos os municípios que se encontram a até 50Km de distância do município base. Para cada município dentro do limite, deveremos adicioná-lo à região e acumular os valores das variáveis de entrada Ent1, Ent2, Ent3, Ent4 e Ent5.

- Ent1 – É o valor de entrada da variável lingüística *Conexão*. Identifica o tipo de banda encontrada na região e nos dá uma forma de quantificar a infra-estrutura do município para suportar o atendimento dos usuários. A conexão pode ser feita através de banda estreita ou de banda larga. O valor de entrada é definido multiplicando-se o número de habitantes de cada município pertencente à região pelas constantes de identificação de bandas definidas:
 - Se o valor na base de dados do município for 0, o número de habitantes é multiplicado por 64, identificando que o município possui banda estreita;
 - Se o valor na base de dados do município for 1, o número de habitantes é multiplicado por 256, identificando que o município possui banda larga telefônica;
 - Se o valor na base de dados do município for 2, o número de habitantes é multiplicado por 512, identificando que o município possui outras formas de banda larga além da telefônica;

Após o somatório de todas estas multiplicações, o total é dividido pelo número de habitantes da região, para termos a média por habitante da taxa de conexão, que é então usada pelo sistema.

- Ent2 – É o valor de entrada da variável lingüística *Urbanização*. Identifica o quanto o município é urbanizado, caracterizando sua aptidão para o uso do serviço de acesso à Internet. É medido em percentagem e é definido em função de 4 serviços de atendimento a população sobre cada município pertencente à região e somado, cada uma variando de 0 a 25%. Após o término do somatório, este é dividido pelo total de municípios da região, para então obtermos a urbanização média da região. Os percentuais são obtidos através da comparação do município com o município do Rio de Janeiro, tido como totalmente urbanizado para efeitos do sistema. Os critérios são:

- Percentual de urbanização calculado pelo número de agências bancárias por habitante no município. É calculado com a seguinte fórmula:

$$Urb_1 = \frac{NB}{Pop} * \frac{25}{Const_1} \quad \text{(Fórmula III.1)}$$

$$Const_1 = \frac{NBRJ}{PopRJ} = 0,0001615768 \quad (\text{Fórmula III.2})$$

Onde:

Urb₁ é o primeiro percentual de urbanização.

NB é o número de agência bancárias no município.

Pop é o total de habitantes do município.

Const₁ é utilizada para adequar o município em questão ao município do Rio de Janeiro, que possui valor máximo de 25%.

NBRJ é o número de agência bancárias no município do Rio de Janeiro.

PopRJ é o total de habitantes do município do Rio de Janeiro.

- Percentual de urbanização calculado pelo número de leitos hospitalares por habitante no município. É calculado com a seguinte fórmula:

$$Urb_2 = \frac{NL}{Pop} * \frac{25}{Const_2} \quad (\text{Fórmula III.3})$$

$$Const_2 = \frac{NLRJ}{PopRJ} = 0,00511 \quad (\text{Fórmula III.4})$$

Onde:

Urb₂ é o segundo percentual de urbanização.

NL é o número de leitos no município.

Pop é o total de habitantes do município.

Const₂ é utilizada para adequar o município em questão ao município do Rio de Janeiro, que possui valor máximo de 25%.

NLRJ é o número de leitos no município do Rio de Janeiro.

PopRJ é o total de habitantes do município do Rio de Janeiro.

- Percentual de urbanização calculado pelo número de empresas não agropecuárias por habitante no município. É calculado com a seguinte fórmula:

$$Urb_3 = \frac{(NE - NEA)}{Pop} * \frac{25}{Const_3} \quad (\text{Fórmula III.5})$$

$$Const_3 = \frac{NERJ - NEARJ}{PopRJ} = 0,024086 \quad (\text{Fórmula III.6})$$

Onde:

Urb_3 é o terceiro percentual de urbanização.

NE é o número de empresas atuantes no município.

NEA é o número de empresas agropecuárias atuantes no município.

Pop é o total de habitantes do município.

$Const_3$ é utilizada para adequar o município em questão ao município do Rio de Janeiro, que possui valor máximo de 25%.

NERJ é o número de empresas atuantes no município do Rio de Janeiro.

NEARJ é o número de empresas agropecuárias atuantes no município do Rio de Janeiro.

PopRJ é o total de habitantes do município do Rio de Janeiro.

- Percentual de urbanização calculado pelo número de estabelecimentos educacionais por habitante no município. É calculado com a seguinte fórmula:

$$Urb_4 = \frac{(NEP + NEF + NEM)}{Pop} * \frac{25}{Const_4} \quad (\text{Fórmula III.7})$$

$$Const_4 = \frac{NEPRJ + NEFRJ + NEMRJ}{PopRJ} = 0,000802259 \quad (\text{Fórmula III.8})$$

Onde:

Urb_4 é o quarto percentual de urbanização.

NEP é o número de estabelecimentos de ensino pré-escolar no município

NEF é o número de estabelecimentos de ensino fundamental no município

NEM é o número de estabelecimentos de ensino médio no município

Pop é o total de habitantes do município

$Const_4$ é utilizada para adequar o município em questão ao município do Rio de Janeiro, que possui valor máximo de 25%.

NEPRJ é o número de estabelecimentos de ensino pré-escolar no município do Rio de Janeiro.

NEFRJ é o número de estabelecimentos de ensino fundamental no município do Rio de Janeiro.

NEMRJ é o número de estabelecimentos de ensino médio no município do Rio de Janeiro.

PopRJ é o total de habitantes do município do Rio de Janeiro.

- A urbanização total (Urb) do município é dada pela soma:

$$Urb = Urb_1 + Urb_2 + Urb_3 + Urb_4 \quad (\text{Fórmula III.9})$$

- Ent3 – É o valor de entrada da variável lingüística *População*. Quantifica o número de habitantes. É a variável responsável pelo total da população dos municípios que compõe a região. O total da população usado no sistema é definido pelo total de habitantes do município base somado ao total de habitantes de todos os municípios que se encontram dentro do limite de 50Km;
- Ent4 – É o valor de entrada da variável lingüística *Concorrência*. Identifica o número médio de concorrentes na região. É a variável responsável pelo total da concorrência. É definido pelo total de provedores existentes em todos os municípios da região, dividido pelo número de municípios pertencentes à região;

- Ent5 – É o valor de entrada da variável lingüística *Renda*. Quantifica a renda da população. É a variável responsável pela renda da região. É definido pelo somatório dos depósitos bancários, somado ao montante dos valores aplicados pelos habitantes em cada município que pertence à região, dividido pela população dessa região.

De posse dos dados de entrada, iniciamos o processo de inferência, que irá carregar os antecedentes e o conseqüente, *fuzzyficar* os dados de entrada, carregar as regras e fazer a inferência para cada regra.

Após a inferência, devemos *defuzzyficar* a saída obtida pela inferência, montar a resposta e enviar o *e-mail* com a resposta para o usuário.

III.6 A Base de Dados

O SGBD escolhido foi o *SQL-Server*® da *Microsoft Corporation*. Qualquer outro gerenciador poderia ter sido usado, tais como *Oracle*®, *Paradox*®, ou até mesmo o *Access*®. Os dados necessários ao sistema tiveram diversas fontes, sendo cada uma delas informada quando da descrição das tabelas a seguir.

O sistema possui cinco tabelas com os dados a serem consultados. Não foram criados módulos de inclusão, alteração ou exclusão de registros, pois, seus dados são obtidos de diversas fontes apenas para consulta, não necessitando nenhum tipo de manutenção pelo sistema. As alterações na base de dados são feitas externamente ao sistema, pois o mesmo importa as informações já atualizadas para a utilização em seu processamento.

III.6.1 A Tabela Usuários

A tabela usuários contém os dados dos usuários autorizados a usar o sistema. Tal tabela possui como campos **ID**, **Login**, **Senha**, **IndAtivo**, **Acesso** e **Mail**. A seguir, podemos acompanhar a descrição desses campos na tabela.

Tabela III.1: Tabela de usuários do sistema.

Campo	Descrição	Tipo	Tamanho
ID	Identificação	Int	4
Login	Nome de login do usuário	Varchar	10
Senha	Senha do usuário	Varchar	10
IndAtivo	Indicativo de usuário ativo	bit	1
Acesso	Nível de acesso do usuário ao sistema	Int	4
Mail	E-Mail do usuário	Varchar	50

III.6.2 A Tabela Funções

Contém as partes de todas as funções de pertinência necessárias para o módulo *fuzzy* do sistema. Seus dados foram obtidos a partir das funções de pertinência criadas para o SIGIAPI. Os cálculos que geraram estes dados se encontram no capítulo IV, seção 3. Lá são descritas detalhadamente não só as fórmulas, mas também os valores definidos para o sistema. Qualquer necessidade de alteração ou qualquer necessidade de ajuste pode ser feita diretamente sobre esta tabela, podendo-se assim ajustar o sistema a diferentes realidades.

Tabela III.2: Tabela de funções de pertinência do sistema.

Campo	Descrição	Tipo	Tamanho
ID	Identificação	Int	4
Antecedente	Nome do Antecedente	Varchar	20
FuncaoParte	Número da Parte	Inteiro	4
TermoLing	Nome do Termo Lingüístico	Varchar	20
CoefAng	Valor do Coeficiente Angular	Float	8
CoefLin	Valor do Coeficiente Linear	Int	4
IntervInic	Valor Inicial do Intervalo	Int	4
IntervFin	Valor final do Intervalo	Int	4

III.6.3 A Tabela Regras

Contém as regras do módulo *fuzzy* do sistema. Seus dados foram gerados a partir da combinação de todas as variáveis lingüísticas. As variáveis definidas para o sistema encontram-se descritas no capítulo IV, seção 4, onde também são apresentadas como exemplo algumas regras usadas no sistema e são identificados os termos lingüísticos. Nesta tabela temos as combinações dos diversos termos lingüísticos dos antecedentes e o termo lingüístico do conseqüente para todas as regras. Qualquer necessidade de ajuste em alguma regra pode

ser feita diretamente sobre esta tabela, facilitando o ajuste do sistema.

Tabela III.3: Tabela de regras do sistema.

Campo	Descrição	Tipo	Tamanho
ID	Identificação	Int	4
Conexão	Identifica o termo lingüístico da Conexão	Varchar	20
Urbanização	Identifica o termo lingüístico da Urbanização	Varchar	20
População	Identifica o termo lingüístico da População	Varchar	20
Concorrência	Identifica o termo lingüístico da Concorrência	Varchar	20
Renda	Identifica o termo lingüístico da Renda	Varchar	20
Potencialidade	Identifica o termo lingüístico da Potencialidade	Varchar	20

III.6.4 A Tabela Municípios

Contém os dados necessários de cada município para as entradas das variáveis lingüísticas Conexão e Concorrência no sistema. Estes dados foram obtidos em pesquisas na Internet, no *site* de buscas *Google*®, no site do IBGE e nos *sites* das operadoras que prestam serviço de conexão em banda larga. As prestadoras pesquisadas foram: *Virtua*®, *AJato*® e *Velox*®.

O campo Provedores foi preenchido da seguinte forma:

- Os municípios que não possuem provedores foram obtidos no *site* do IBGE e para eles foi fornecido o valor 0 (zero);
- Os municípios que possuem provedores tiveram suas quantidades obtidas a partir de consultas ao *site Google*.

O campo Banda foi preenchido da seguinte forma:

- Municípios que não possuem banda larga foram preenchidos com 0 (zero);
- Municípios que possuem banda larga foram preenchidos com 1 (um);
- Municípios que possuem banda larga acima de 256 Kb foram preenchidos com 2 (dois);

Tabela III.4: Tabela de municípios do sistema.

Campo	Descrição	Tipo	Tamanho
Código	Identificação do Município (código IBGE)	Varchar	6
Município	Nome do Município (Nome IBGE)	Varchar	35
Provedores	Número de Provedores	Int	4
Banda	Tipo de Banda (0,1, ou 2)	Int	4

III.6.5 A Tabela Síntese

Esta tabela foi importada do CD da Base de Informações Municipais do IBGE, 2ª Edição de 2000 [53]. Contém os dados dos municípios necessários para os cálculos de distância, para as entradas das variáveis lingüísticas de População, Urbanização e Renda no sistema.

Fontes de informação para o sistema:

- Contagem da População de 1996;
- Cadastro Central de Empresas de 1996;
- Malha Municipal Oficial de 1997;
- Censo Educacional de 1997;
- Departamento de Informática do DATASUS;
- Banco Central do Brasil de 1998.

Tabela III.5: Descrição da tabela de síntese dos municípios – (IBGE Senso 2000).

Campo	Descrição	Tipo	Tamanho
Código	Identificação do Município (código IBGE)	Varchar	6
N1	População Total	Float	8
N35	Número de estabelecimentos de ensino pré-escolar	Float	8
N36	Número de estabelecimentos de ensino fundamental	Float	8
N37	Número de estabelecimentos de ensino médio	Float	8
N39	Número de Leitos Hospitalares	Float	8
N54	Número de Empresas com CGC Atuentes	Float	8
N77	Número de estabelecimentos Agropecuários	Float	8
N87	Número de Agências Bancárias	Float	8
N88	Total em Depósitos Bancários	Float	8
N89	Total em Aplicações Bancárias	Float	8
N101	Nome do Município (Nome IBGE)	Varchar	35
Latitude	Latitude do Município	Float	8
Longitude	Longitude do Município	Float	8

III.7 Comentários

Neste capítulo, observamos não só o funcionamento lógico do sistema, mas também conhecemos as linguagens e as ferramentas necessárias para o seu sucesso. Vimos também como os dados foram obtidos, tratados e manipulados de forma a gerar como informação os valores de entrada do motor de inferência. No próximo capítulo, teremos a implementação do sistema.

CAPÍTULO IV

Implantação do SIGAIPI

IV.1 Introdução

Este projeto foi desenvolvido utilizando em conjunto as facilidades de uso da linguagem HTML para a criação de interfaces de entradas e de apresentação da resposta gerada pelo usuário e a tecnologia *Active Server Pages* (ASP) capaz de oferecer a interface entre a HTML e o objeto *Active X* criado em Visual Basic que compõe as classes capazes de processar a escolha do usuário e de manipular os dados oriundos de um SGBD. Para o armazenamento dos dados foi utilizado o SQL Server.

Neste capítulo, veremos como foram utilizados os conceitos e as técnicas mostradas nos capítulos anteriores para o desenvolvimento do trabalho. Na seção IV.2, é apresentada uma descrição da solução empregada no desenvolvimento do SIGAIPI.

As seções IV.3 e IV.4 mostram como foram implementadas as técnicas da lógica nebulosa para o sistema. Na seção IV.3, as variáveis lingüísticas são apresentadas e descritas, assim como os termos dessas variáveis e suas respectivas funções de pertinência. Na seção IV.4, são apresentadas e definidas as regras de inferência que são usadas pelo sistema de inferência criado.

Na seção IV.5, são exibidos os métodos das classes criadas para a realização dos cálculos cartográficos para o acesso à base de dados e para o sistema de inferência projetado, suas descrições e em alguns casos seus respectivos algoritmos. Finalmente a seção IV.6 mostrará os testes realizados com o sistema pronto.

IV.2 Definição do Projeto

O sistema foi concebido para atender aos pequenos ou pouco desenvolvidos municípios do estado do Rio de Janeiro, pois os maiores municípios como Rio de Janeiro e Niterói se encontram bem atendidos na questão de provedores de Internet.

Como um provedor de Internet segundo a proposta desse trabalho pode atender a um grupo de municípios, que estejam a até 50Km de distância. Foi decidido que seria melhor

agrupar o município base aos municípios que se encontram a até uma distância de 50Km deste. Desse modo, o investimento para a implantação não seria em virtude de se atender a apenas um município, mas sim, a um conjunto de municípios, formando uma região, com maior potencialidade de crescimento do negócio e conseqüente retorno para o investidor. A região não é delimitada e sim gerada pelo sistema, a partir do chamado município base, onde será implantada a infra-estrutura do provedor, para acesso e atendimento aos clientes da região gerada pelo conjunto dos municípios que se encontram a até uma distância de 50km.

A implantação do SIGAPI se inicia pela tela de acesso, onde o usuário deve se identificar e inserir sua respectiva senha. Uma vez que o sistema valide o usuário, esse deverá fazer a escolha de um município do Estado do Rio de Janeiro. A partir da escolha do município base será formada a região. Uma vez obtida a relação dos municípios que formarão a região, os dados desses municípios serão combinados, de modo que o sistema os trate de forma única, definindo conexão, urbanização, população, concorrência e renda em função da região e não do município base.

As entradas para o sistema de inferência serão então os dados definidos pela região gerada. O motor de inferência do sistema receberá esses dados como parâmetros e iniciará o processo de inferência. Após a avaliação executada pelo sistema na região definida a partir do município base, o usuário receberá a resposta do sistema, contendo o resultado da inferência. A resposta gerada pelo sistema será então enviada ao usuário de duas maneiras distintas:

1. Uma na tela será gerada para o *browser* do usuário contendo a resposta do sistema de forma imediata em formato HTML;
2. Um *E-mail* será gerado e enviado ao usuário, contendo a resposta do sistema.

Em ambos os casos, o usuário receberá junto com a resposta um relatório contendo o município escolhido, os municípios que compõem a região, os parâmetros de entrada e os resultados.

IV.3 Aplicação da Teoria dos Conjuntos Nebulosos ao Problema

A implementação do motor de inferência do SIGAPI teve início com a escolha das variáveis lingüísticas do Sistema de Lógica Nebulosa (SLN) e com a definição dos seus respectivos termos. O SLN, responsável por indicar a potencialidade de cada região para se

implantar os provedores de acesso à Internet, tem 450 regras de inferência as quais por sua vez possuem cinco antecedentes e um conseqüente.

Todos os termos de todas as variáveis lingüísticas do SLN desenvolvido têm suas funções de pertinência montadas com segmentos de reta, definindo funções lineares, na maioria das vezes apresentando perfis triangulares. Assim, devemos calcular o coeficiente linear e o coeficiente angular para cada parte de cada função de pertinência. Como cada função de reta é definida através da fórmula:

$$y = Mx + Q \quad (\text{Fórmula IV.1})$$

Onde:

$M \rightarrow$ é o Coeficiente Angular;

$Q \rightarrow$ é o Coeficiente Linear.

Dessa forma, como podemos obter os valores de y, x e Q diretamente sobre o gráfico, basta calcularmos o Coeficiente Angular (M), através da fórmula:

$$M = \frac{y - Q}{x} \quad (\text{Fórmula IV.2})$$

Uma vez que os coeficientes lineares e os coeficientes angulares foram calculados para todas as partes de todas as funções de pertinência, todas as partes das funções de pertinência encontram-se codificadas em um banco de dados de modo que basta atualizar o banco para se alterar, incluir ou excluir uma parte de uma função de pertinência, sem que haja a necessidade de se modificar o código interno do programa. Desta forma, podemos facilmente alterar o número de partes de uma função de pertinência ou mesmo os intervalos iniciais e finais, assim como o universo de discurso U de uma variável lingüística.

A seguir são apresentadas as variáveis lingüísticas dos cinco antecedentes e seus respectivos termos definidas pela quintupla $(x, T(x), U, G, \tilde{M})$. Também serão apresentadas as respectivas funções de pertinência dos termos das variáveis lingüísticas definidas como antecedentes e como conseqüente.

- Para o primeiro antecedente. Variável lingüística: **Conexão**.

Tabela IV.1: Descrição da variável lingüística: **Conexão**.

$x=$	Conexão
$T(x)=$	$\{Estreita, Larga\}$
$U=$	[0kbps, 512kbps]
$M:$	$M(Estreita) = \{(u, \mu_{Estreita}(u)) \mid u \in [0kbps, 512kbps]\}$ $M(Larga) = \{(u, \mu_{Larga}(u)) \mid u \in [0kbps, 512kbps]\}$

$$\circ M(Estreita) = \{(u, \mu_{Estreita}(u)) \mid u \in [0kbps, 512kbps]\}$$

Tabela IV.2: Constantes das partes da função de pertinência do termo Estreita.

Parte	Intervalo inicial	Intervalo Final	Coefficiente Linear	Coefficiente Angular
1	0	96	1	-0,0104
2	96	512	0	0

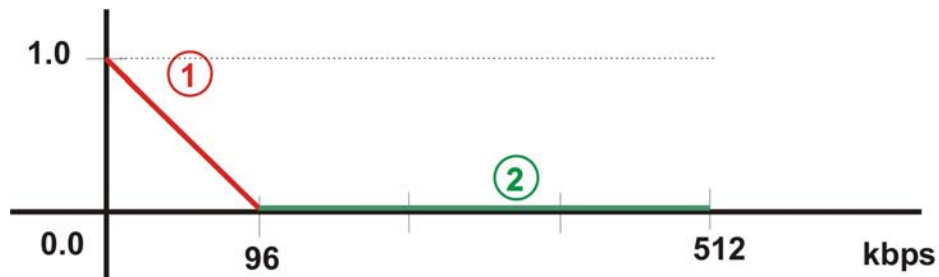


Figura IV.1: Função de pertinência do termo Estreita (variável **Conexão**).

$$\circ M(Larga) = \{(u, \mu_{Larga}(u)) \mid u \in [0kbps, 512kbps]\}$$

Tabela IV.3: Constantes das partes da função de pertinência do termo Larga.

Parte	Intervalo inicial	Intervalo Final	Coefficiente Linear	Coefficiente Angular
1	0	96	0	0,0104
2	96	512	1	0

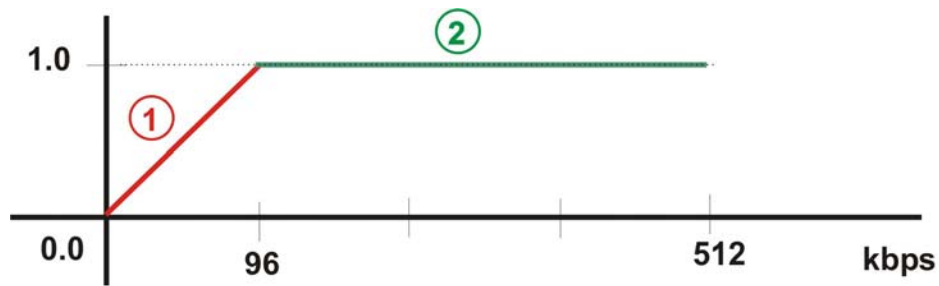


Figura IV.2: Função de pertinência do termo Larga (variável *Conexão*).

- Para o segundo antecedente. Variável lingüística: *Urbanização*.

Tabela IV.4: Descrição da variável lingüística: *Urbanização*.

$x=$	<i>Urbanização</i>
$T(x)=$	$\{Levemente, Pouco, Urbanizada, Muito, Totalmente\}$
$U=$	$[0\%,100\%]$
$M:$	$M(Levemente) = \{(u, \mu_{Levemente}(u)) \mid u \in [0\%, 100\%]\}$ $M(Pouco) = \{(u, \mu_{Pouco}(u)) \mid u \in [0\%, 100\%]\}$ $M(Urbanizada) = \{(u, \mu_{Urbanizada}(u)) \mid u \in [0\%, 100\%]\}$ $M(Muito) = \{(u, \mu_{Muito}(u)) \mid u \in [0\%, 100\%]\}$ $M(Totalmente) = \{(u, \mu_{Totalmente}(u)) \mid u \in [0\%, 100\%]\}$

$$\circ \quad M(Levemente) = \{(u, \mu_{Levemente}(u)) \mid u \in [0\%, 100\%]\}$$

Tabela IV.5: Constantes das partes da função de pertinência do termo Levemente.

Parte	Intervalo inicial	Intervalo Final	Coeficiente Linear	Coeficiente Angular
1	0	25	1	-0,04
2	25	100	0	0

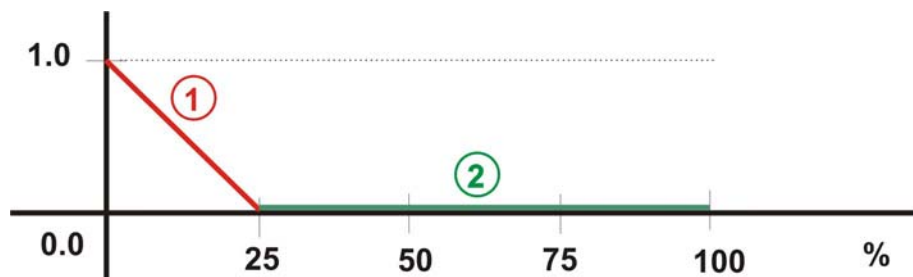


Figura IV.3: Função de pertinência do termo Levemente (variável *Urbanização*).

$$\circ \quad M(\text{Pouco}) = \{(u, \mu_{\text{Pouco}}(u)) \mid u \in [0\%, 100\%]\}$$

Tabela IV.6: Constantes das partes da função de pertinência do termo Pouco.

Parte	Intervalo inicial	Intervalo Final	Coefficiente Linear	Coefficiente Angular
1	0	25	0	0,04
2	25	50	2	-0,04
3	50	100	0	0

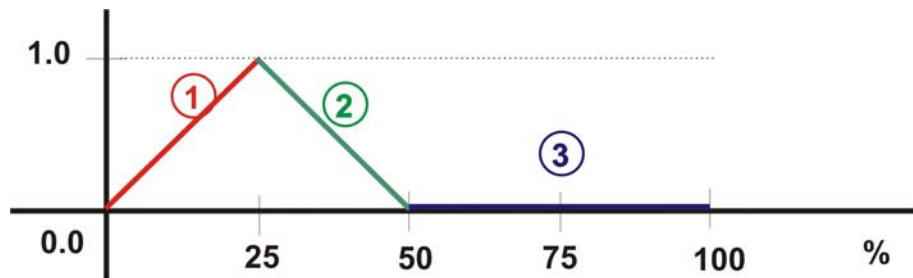


Figura IV.4: Função de pertinência do termo Pouco (variável *Urbanização*).

$$\circ \quad M(\text{Urbanizada}) = \{(u, \mu_{\text{Urbanizada}}(u)) \mid u \in [0\%, 100\%]\}$$

Tabela IV.7: Constantes das partes da função de pertinência do termo Urbanizada.

Parte	Intervalo inicial	Intervalo Final	Coefficiente Linear	Coefficiente Angular
1	0	25	0	0
2	25	50	-1	0,04
3	50	75	3	-0,04
4	75	100	0	0

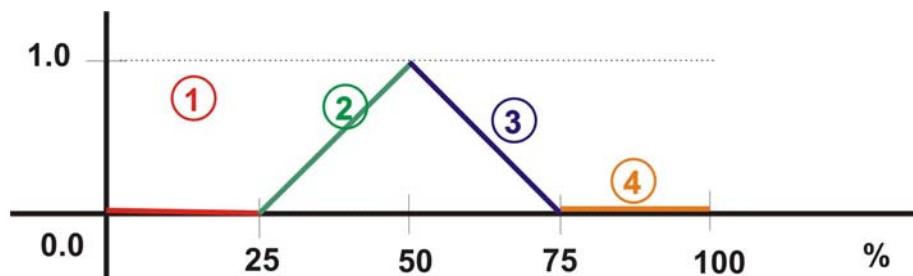


Figura IV.5: Função de pertinência do termo Urbanizada (variável *Urbanização*).

$$\circ \quad M(\text{Muito}) = \{(u, \mu_{\text{Muito}}(u)) \mid u \in [0\%, 100\%]\}$$

Tabela IV.8: Constantes das partes da função de pertinência do termo Muito.

Parte	Intervalo inicial	Intervalo Final	Coefficiente Linear	Coefficiente Angular
1	0	25	0	0
2	25	50	-2	0,04
3	50	75	4	-0,04

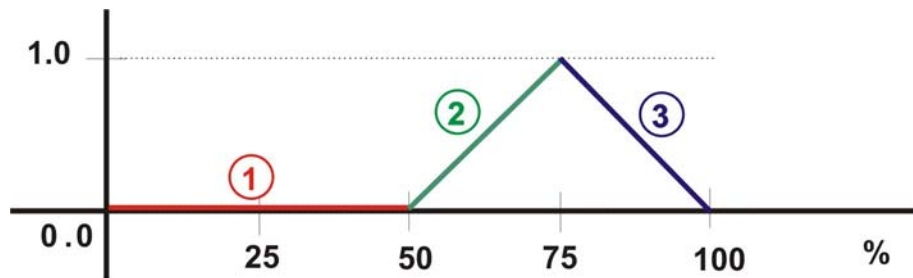


Figura IV.6: Função de pertinência do termo Muito (variável *Urbanização*).

$$\circ \quad M(\text{Totalmente}) = \{(u, \mu_{\text{Totalmente}}(u)) \mid u \in [0\%, 100\%]\}$$

Tabela IV.9: Constantes das partes da função de pertinência do termo Totalmente.

Parte	Intervalo inicial	Intervalo Final	Coefficiente Linear	Coefficiente Angular
1	0	25	0	0
2	25	50	-3	0,04

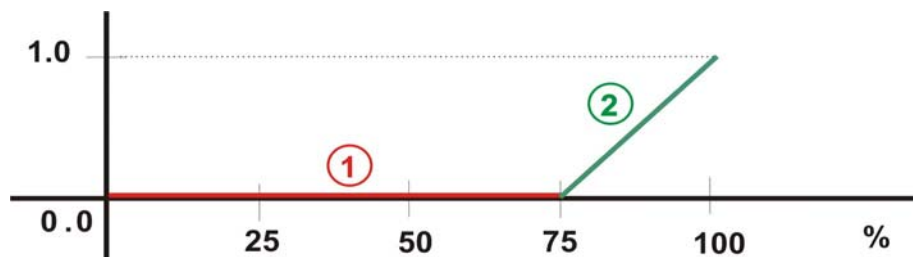


Figura IV.7: Função de pertinência do termo Totalmente (variável *Urbanização*).

- Para o terceiro antecedente. Variável lingüística: **População**.

Tabela IV.10: Descrição da variável lingüística: **População**.

$x=$	População
$T(x)=$	$\{Levemente, Pouco, Populada, Muito, Extremamente\}$
$U=$	$[0,400.000]$
$M:$	$M(Levemente) = \{(u, \mu_{Levemente}(u)) \mid u \in [0,400.000]\}$ $M(Pouco) = \{(u, \mu_{Pouco}(u)) \mid u \in [0,400.000]\}$ $M(Populada) = \{(u, \mu_{Populada}(u)) \mid u \in [0,400.000]\}$ $M(Muito) = \{(u, \mu_{Muito}(u)) \mid u \in [0,400.000]\}$ $M(Extremamente) = \{(u, \mu_{Extremamente}(u)) \mid u \in [0,400.000]\}$

$$\circ M(Levemente) = \{(u, \mu_{Levemente}(u)) \mid u \in [0,100.000]\}$$

Tabela IV.11: Constantes das partes da função de pertinência do termo Levemente.

Parte	Intervalo inicial	Intervalo Final	Coeficiente Linear	Coeficiente Angular
1	0	100.000	1	-0,00001
2	100.000	400.000	0	0

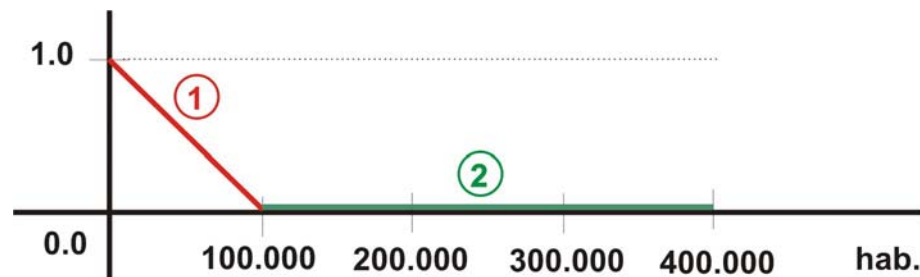


Figura IV.8: Função de pertinência do termo Levemente (variável **População**).

$$\circ M(Pouco) = \{(u, \mu_{Pouco}(u)) \mid u \in [0,100.000]\}$$

Tabela IV.12: Constantes das partes da função de pertinência do termo Pouco.

Parte	Intervalo inicial	Intervalo Final	Coeficiente Linear	Coeficiente Angular
1	0	100.000	0	0,00001
2	100.000	200.000	2	-0,00001
3	200.000	400.000	0	0

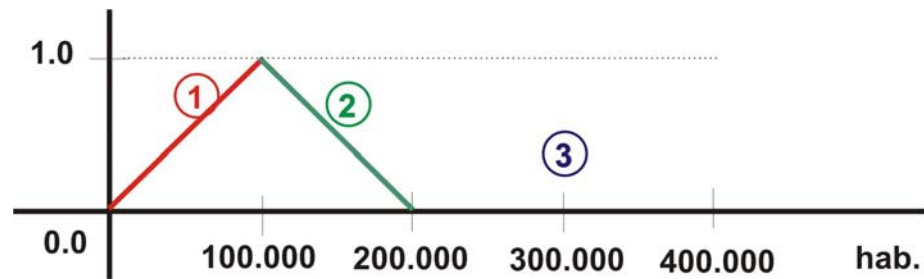


Figura IV.9: Função de pertinência do termo Pouco (variável *População*).

$$\circ \quad M(\text{Populada}) = \{(u, \mu_{\text{Pouco}}(u)) \mid u \in [0, 100.000]\}$$

Tabela IV.13: Constantes das partes da função de pertinência do termo Populada.

Parte	Intervalo inicial	Intervalo Final	Coefficiente Linear	Coefficiente Angular
1	0	100.000	0	0
2	100.000	200.000	-1	0,00001
3	200.000	300.000	3	-0,00001
4	300.000	400.000	0	0

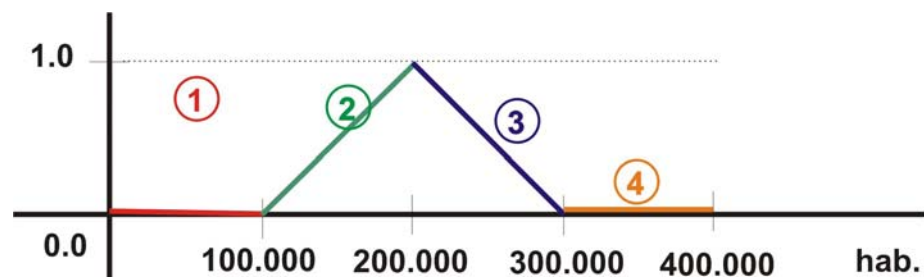


Figura IV.10: Função de pertinência do termo Populada (variável *População*).

$$\circ \quad M(\text{Muito}) = \{(u, \mu_{\text{Muito}}(u)) \mid u \in [0, 100.000]\}$$

Tabela IV.14: Constantes das partes da função de pertinência do termo Muito.

Parte	Intervalo inicial	Intervalo Final	Coefficiente Linear	Coefficiente Angular
1	0	200.000	0	0
2	200.000	300.000	-2	0,00001
3	300.000	400.000	4	-0,00001

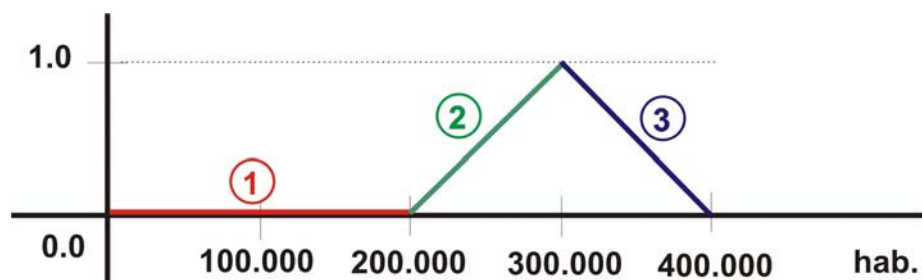


Figura IV.11: Função de pertinência do termo Muito (variável *População*).

$$\circ \quad M(\text{Extremamente}) = \{(u, \mu_{\text{Extremamente}}(u)) \mid u \in [0, 100.000]\}$$

Tabela IV.15: Constantes das partes da função de pertinência do termo Extremamente.

Parte	Intervalo inicial	Intervalo Final	Coefficiente Linear	Coefficiente Angular
1	0	100.000	0	0
2	100.000	400.000	-3	0,00001

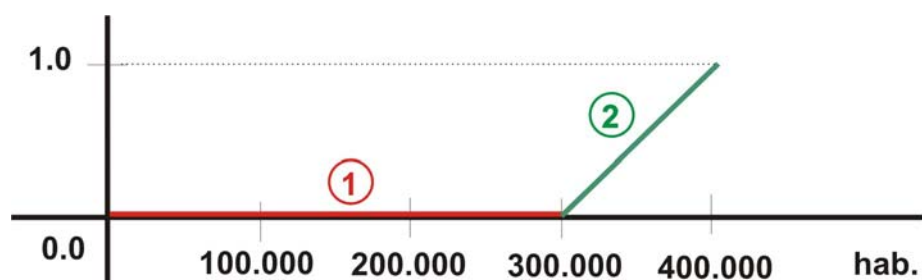


Figura IV.12: Função de pertinência do termo Extremamente (variável *População*).

- Para o quarto antecedente. Variável lingüística: *Concorrência*.

Tabela IV.16: Descrição da variável lingüística: *Concorrência*.

$x =$	<i>Concorrência</i>
$T(x) =$	<i>{Sem Concorrência, Com Concorrência, Extremamente Concorrente}</i>
$U =$	[0,10]
$M:$	$M(\text{Sem Concorrência}) = \{(u, \mu_{\text{Sem Concorrência}}(u)) \mid u \in [0, 10]\}$ $M(\text{Com Concorrência}) = \{(u, \mu_{\text{Com Concorrência}}(u)) \mid u \in [0, 10]\}$ $M(\text{Extremamente Concorrente}) = \{(u, \mu_{\text{Extremamente Concorrente}}(u)) \mid u \in [0, 10]\}$

$$\circ \quad M(\text{Sem Concorrência}) = \{(u, \mu_{\text{Sem Concorrência}}(u)) \mid u \in [0,10]\}$$

Tabela IV.17: Constantes das partes da função de pertinência do termo Sem Concorrência.

Parte	Intervalo inicial	Intervalo Final	Coefficiente Linear	Coefficiente Angular
1	0	5	1	-0,2
2	5	10	0	0

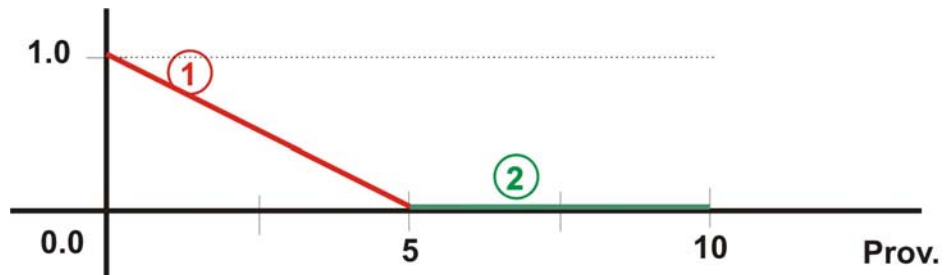


Figura IV.13: Função de pertinência do termo Sem Concorrência (variável *Concorrência*).

$$\circ \quad M(\text{Com Concorrência}) = \{(u, \mu_{\text{Com Concorrência}}(u)) \mid u \in [0,10]\}$$

Tabela IV.18: Constantes das partes da função de pertinência do termo Com Concorrência.

Parte	Intervalo inicial	Intervalo Final	Coefficiente Linear	Coefficiente Angular
1	0	5	0	0,2
2	5	10	2	-0,2

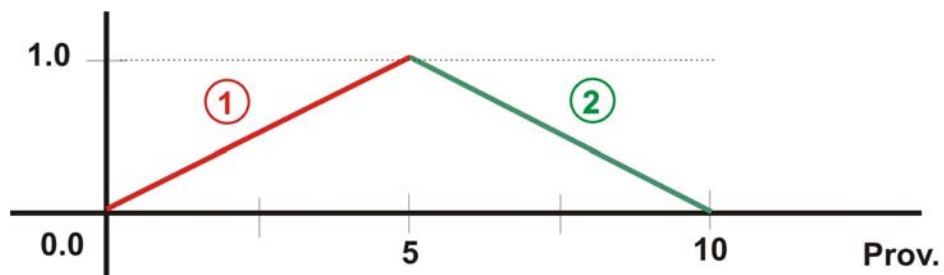


Figura IV.14: Função de pertinência do termo Com Concorrência (variável *Concorrência*).

- $M(\text{Extremamente Concorrente}) = \{(u, \mu_{\text{Extremamente Concorrente}}(u)) \mid u \in [0, 10]\}$

Tabela IV.19: Constantes das partes da função de pertinência do termo Extremamente Concorrente.

Parte	Intervalo inicial	Intervalo Final	Coefficiente Linear	Coefficiente Angular
1	0	5	0	0
2	5	10	-1	0,2

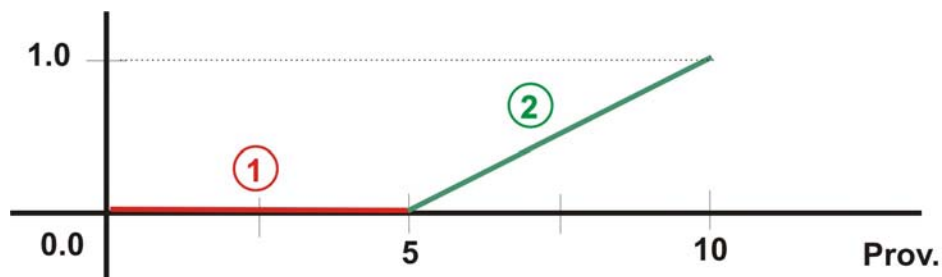


Figura IV.15: Função de pertinência do termo Extremamente Concorrente (variável *Concorrência*).

- Para o quinto antecedente. Variável lingüística: **Renda**.

Tabela IV.20: Descrição da variável lingüística: **Renda**.

$x =$	Renda
$T(x) =$	$\{Baixa, Média, Alta\}$
$U =$	$[R\$0,00, R\$4.000,00]$
$M:$	$M(Baixa) = \{(u, \mu_{Baixa}(u)) \mid u \in [0, 4.000]\}$ $M(Média) = \{(u, \mu_{Média}(u)) \mid u \in [0, 4.000]\}$ $M(Alta) = \{(u, \mu_{Alta}(u)) \mid u \in [0, 4.000]\}$

$$\circ M(\text{Baixa}) = \{(u, \mu_{\text{Baixa}}(u)) \mid u \in [0, 4.000]\}$$

Tabela IV.21: Constantes das partes da função de pertinência do termo Baixa.

Parte	Intervalo inicial	Intervalo Final	Coefficiente Linear	Coefficiente Angular
1	0	1.000	1	0
2	1.000	2.000	2	-0,001
3	2.000	4.000	0	0

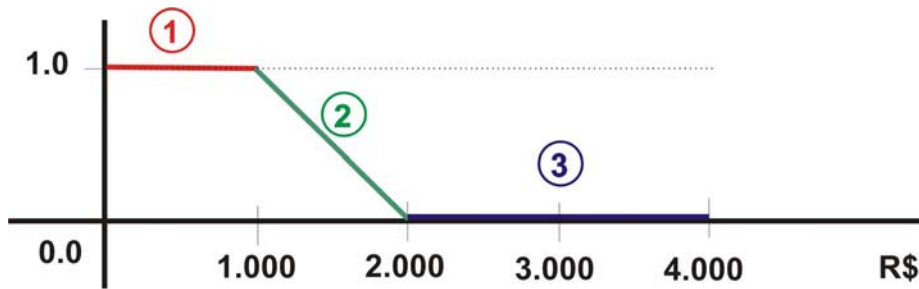


Figura IV.16: Função de pertinência do termo Baixa (variável *Renda*).

$$\circ M(\text{Média}) = \{(u, \mu_{\text{Média}}(u)) \mid u \in [0, 4.000]\}$$

Tabela IV.22: Constantes das partes da função de pertinência do termo Média.

Parte	Intervalo inicial	Intervalo Final	Coefficiente Linear	Coefficiente Angular
1	0	1.000	0	0
2	1.000	2.000	-1	0,001
3	2.000	3.000	3	-0,001
4	3.000	4.000	0	0

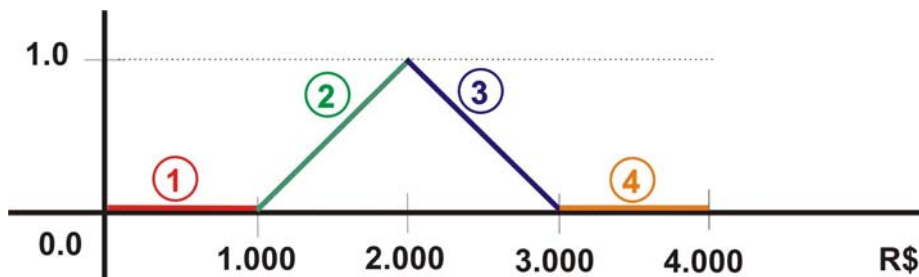


Figura IV.17: Função de pertinência do termo Média (variável *Renda*).

$$\circ \quad M(\text{Alta}) = \{(u, \mu_{\text{Alta}}(u)) \mid u \in [0, 4.000]\}$$

Tabela IV.23: Constantes das partes da função de pertinência do termo Alta.

Parte	Intervalo inicial	Intervalo Final	Coefficiente Linear	Coefficiente Angular
1	0	2.000	0	0
2	2.000	3.000	-2	0,001
3	3.000	4.000	1	0

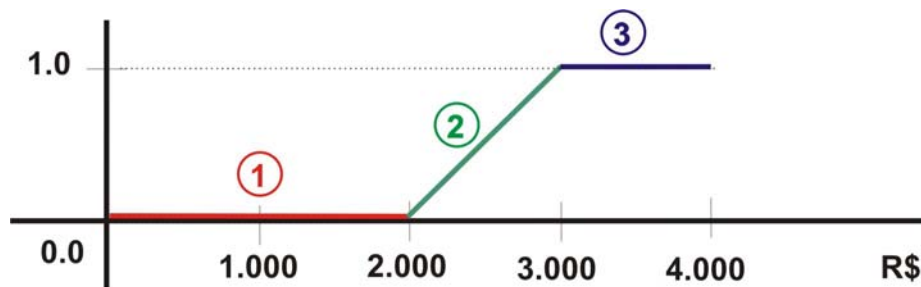


Figura IV.18: Função de pertinência do termo Alta (variável *Renda*).

A variável lingüística do conseqüente caracteriza a resposta do serviço consultado acerca do potencial de uma dada região, logo, essa variável recebe o nome de *Potencialidade* e seus termos são apresentados a seguir.

- Para o conseqüente. Variável lingüística: ***Potencialidade***.

Tabela IV.24: Descrição da variável lingüística: ***Potencialidade***.

$x=$	<i>Potencialidade</i>
$T(x)=$	$\{Sem, Baixa, Média, Boa, Ótima\}$
$U=$	$[0\%, 100\%]$
$M:$	$M(Sem) = \{(u, \mu_{Sem}(u)) \mid u \in [0\%, 100\%]\}$ $M(Baixa) = \{(u, \mu_{Baixa}(u)) \mid u \in [0\%, 100\%]\}$ $M(Média) = \{(u, \mu_{Média}(u)) \mid u \in [0\%, 100\%]\}$ $M(Boa) = \{(u, \mu_{Boa}(u)) \mid u \in [0\%, 100\%]\}$ $M(Ótima) = \{(u, \mu_{Ótima}(u)) \mid u \in [0\%, 100\%]\}$

$$\circ M(\text{Sem}) = \{(u, \mu_{\text{Sem}}(u)) \mid u \in [0\%, 100\%]\}$$

Tabela IV.25: Constantes das partes da função de pertinência do termo Sem.

Parte	Intervalo inicial	Intervalo Final	Coefficiente Linear	Coefficiente Angular
1	0	25	1	-0,04
2	25	100	0	0

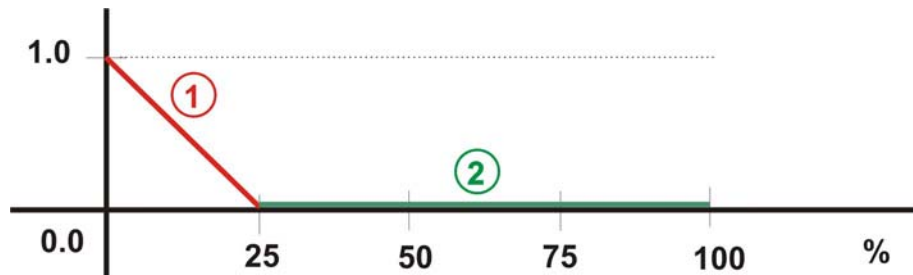


Figura IV.19: Função de pertinência do termo Sem (variável *Potencialidade*).

$$\circ M(\text{Baixa}) = \{(u, \mu_{\text{Baixa}}(u)) \mid u \in [0\%, 100\%]\}$$

Tabela IV.26: Constantes das partes da função de pertinência do termo Baixa.

Parte	Intervalo inicial	Intervalo Final	Coefficiente Linear	Coefficiente Angular
1	0	25	0	0,04
2	25	50	2	-0,04
3	50	100	0	0

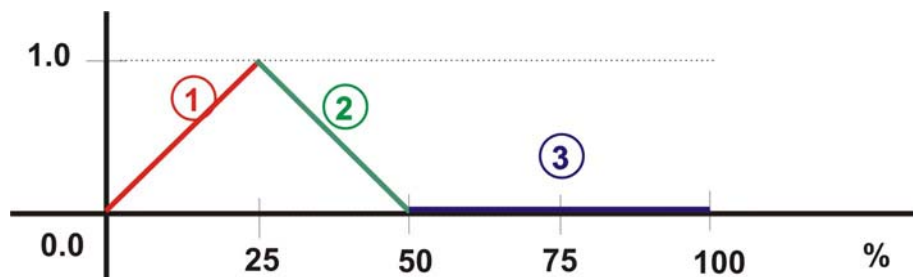


Figura IV.20: Função de pertinência do termo Baixa (variável *Potencialidade*).

$$\circ \quad M(\text{Média}) = \{(u, \mu_{\text{Média}}(u)) \mid u \in [0\%, 100\%]\}$$

Tabela IV.27: Constantes das partes da função de pertinência do termo Média.

Parte	Intervalo inicial	Intervalo Final	Coefficiente Linear	Coefficiente Angular
1	0	25	0	0
2	25	50	-1	0,04
3	50	75	3	-0,04
4	75	100	0	0

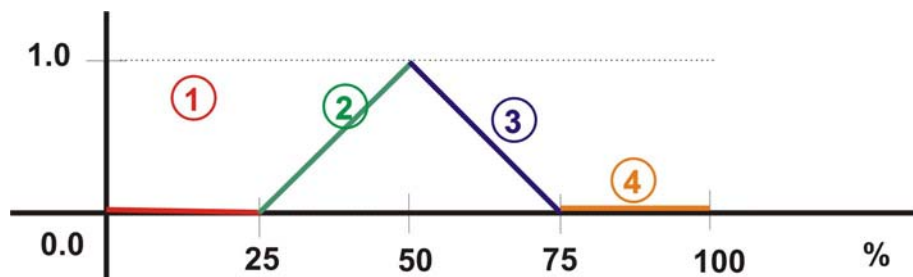


Figura IV.21: Função de pertinência do termo Média (variável *Potencialidade*).

$$\circ \quad M(\text{Boa}) = \{(u, \mu_{\text{Boa}}(u)) \mid u \in [0\%, 100\%]\}$$

Tabela IV.28: Constantes das partes da função de pertinência do termo Boa.

Parte	Intervalo inicial	Intervalo Final	Coefficiente Linear	Coefficiente Angular
1	0	25	0	0
2	25	50	-2	0,04
3	50	75	4	-0,04

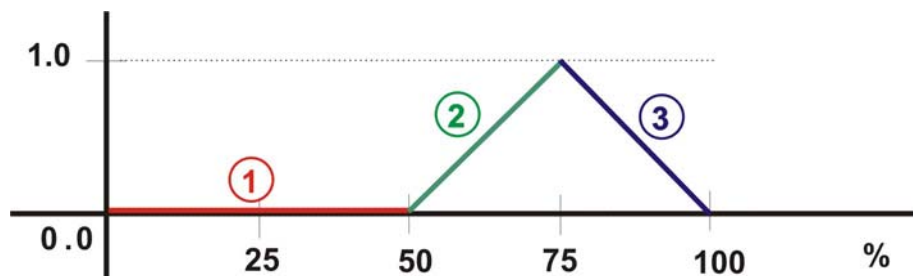


Figura IV.22: Função de pertinência do termo Boa (variável *Potencialidade*).

$$\circ M(\text{Ótima}) = \{(u, \mu_{\text{Ótima}}(u)) \mid u \in [0\%, 100\%]\}$$

Tabela IV.29: Constantes das partes da função de pertinência do termo Ótima.

Parte	Intervalo inicial	Intervalo Final	Coefficiente Linear	Coefficiente Angular
1	0	25	0	0
2	25	50	-3	0,04

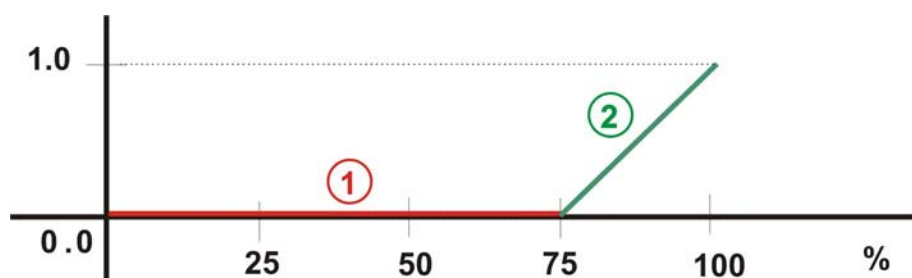


Figura IV.23: Função de pertinência do termo Ótima (variável *Potencialidade*).

IV.4 Definição das Regras de Inferência

As regras de inferência são fundamentais para a obtenção de uma resposta satisfatória. O sistema utiliza a combinação de todas as regras para inferir e gerar uma resposta com a melhor precisão possível. Podemos obter o total de regras do sistema, através da multiplicação da quantidade de termos encontrados em cada uma das variáveis lingüísticas, através de um arranjo. Através de um arranjo foram então geradas todas as regras, necessitando então adicionar aos termos da variável lingüística Potencialidade para cada regra. Posteriormente todas foram incluídas na tabela Regras do sistema, descrita no apêndice A.

Tabela IV.30: Variáveis e quantidade de termos lingüísticos.

Variável Lingüística	Número de Termos Lingüísticos
<i>Conexão</i>	2
<i>Urbanização</i>	5
<i>População</i>	5
<i>Concorrência</i>	3
<i>Renda</i>	3
Número Total de Regras	450

A seguir, são mostradas algumas das 450 regras de inferência do SLN. Tais regras encontram-se codificadas em um banco de dados de modo que basta atualizar o banco para se alterar, incluir ou excluir uma regra, sem que haja a necessidade de se modificar o código interno do programa.

Agrupando as regras por potencialidade, veremos então três exemplos de regras. No apêndice A poderemos ver todas as regras usadas pelo sistema.

- Potencialidade: M(Ótima) – Total de Regras: 3.

○ **Regra: 399**

SE	Conexão (α) é Larga	E	Urbanização (α) é Muito
E	População (α) é Extremamente	E	Concorrência (α) é Sem Concorrença
E	Renda (α) é Alta	ENTÃO	Potencialidade é Ótima

○ **Regra: 435**

SE	Conexão (α) é Larga	E	Urbanização (α) é Totalmente
E	População (α) é Muito	E	Concorrência (α) é Sem Concorrença
E	Renda (α) é Alta	ENTÃO	Potencialidade é Ótima

○ **Regra: 444**

SE	Conexão (α) é Larga	E	Urbanização (α) é Totalmente
E	População (α) é Extremamente	E	Concorrência (α) é Sem Concorrença
E	Renda (α) é Alta	ENTÃO	Potencialidade é Ótima

- Potencialidade: M(Boa) – Total de Regras: 21.

○ **Regra: 210**

SE	Conexão (α) é Estreita	E	Urbanização (α) é Totalmente
E	População (α) é Muito	E	Concorrência (α) é Sem Concorrença
E	Renda (α) é Alta	ENTÃO	Potencialidade é Boa

○ **Regra: 264**

SE	Conexão (α) é Larga	E	Urbanização (α) é Levemente
E	População (α) é Extremamente	E	Concorrência (α) é Sem Concorrença
E	Renda (α) é Alta	ENTÃO	Potencialidade é Boa

○ **Regra: 402**

SE	Conexão (α) é Larga	E	Urbanização (α) é Muito
E	População (α) é Extremamente	E	Concorrência (α) é Com Concorrença
E	Renda (α) é Alta	ENTÃO	Potencialidade é Boa

○ **Regra: 434**

SE	Conexão (α) é Larga	E	Urbanização (α) é Totalmente
E	População (α) é Muito	E	Concorrência (α) é Sem Concorrença
E	Renda (α) é Média	ENTÃO	Potencialidade é Boa

- Potencialidade: M(Média) – Total de Regras: 184.

○ **Regra: 66**

SE	Conexão (α) é Estreita	E	Urbanização (α) é Pouco
E	População (α) é Populada	E	Concorrência (α) é Sem Concorrença
E	Renda (α) é Alta	ENTÃO	Potencialidade é Média

○ **Regra: 135**

SE	Conexão (α) é Estreita	E	Urbanização (α) é Urbanizada
E	População (α) é Extremamente	E	Concorrência (α) é Extremamente
E	Renda (α) é Alta	ENTÃO	Potencialidade é Média

○ **Regra: 371**

SE	Conexão (α) é Larga	E	Urbanização (α) é Muito
E	População (α) é Pouco	E	Concorrência (α) é Sem Concorrença
E	Renda (α) é Média	ENTÃO	Potencialidade é Média

○ **Regra: 384**

SE	Conexão (α) é Larga	E	Urbanização (α) é Muito
E	População (α) é Populada	E	Concorrência (α) é Com Concorrença
E	Renda (α) é Alta	ENTÃO	Potencialidade é Média

- Potencialidade: M(Baixa) – Total de Regras: 234.

○ **Regra: 65**

SE	Conexão (α) é Estreita	E	Urbanização (α) é Pouco
E	População (α) é Populada	E	Concorrência (α) é Sem Concorrença
E	Renda (α) é Média	ENTÃO	Potencialidade é Baixa

○ **Regra: 224**

SE	Conexão (α) é Estreita	E	Urbanização (α) é Totalmente
E	População (α) é Extremamente	E	Concorrência (α) é Extremamente
E	Renda (α) é Média	ENTÃO	Potencialidade é Baixa

○ **Regra: 226**

SE	Conexão (α) é Larga	E	Urbanização (α) é Levemente
E	População (α) é Levemente	E	Concorrência (α) é Sem Concorrença
E	Renda (α) é Baixa	ENTÃO	Potencialidade é Baixa

○ **Regra: 382**

SE	Conexão (α) é Larga	E	Urbanização (α) é Muito
E	População (α) é Populada	E	Concorrência (α) é Com Concorrença
E	Renda (α) é Baixa	ENTÃO	Potencialidade é Baixa

- Potencialidade: M(Sem) – Total de Regras: 8.

○ **Regra: 4**

SE	Conexão (α) é Estreita	E	Urbanização (α) é Levemente
E	População (α) é Levemente	E	Concorrência (α) é Com Concorrença
E	Renda (α) é Baixa	ENTÃO	Potencialidade é Sem

○ **Regra: 16**

SE	Conexão (α) é Estreita	E	Urbanização (α) é Levemente
E	População (α) é Pouco	E	Concorrência (α) é Extremamente
E	Renda (α) é Baixa	ENTÃO	Potencialidade é Sem

○ **Regra: 52**

SE	Conexão (α) é Estreita	E	Urbanização (α) é Pouco
E	População (α) é Levemente	E	Concorrência (α) é Extremamente
E	Renda (α) é Baixa	ENTÃO	Potencialidade é Sem

○ **Regra: 97**

SE	Conexão (α) é Estreita	E	Urbanização (α) é Urbanizada
E	População (α) é Levemente	E	Concorrência (α) é Extremamente
E	Renda (α) é Baixa	ENTÃO	Potencialidade é Sem

IV.5 Definição das Classes e Métodos

Para o processamento principal do sistema foi desenvolvida uma DLL com o Visual Basic 6.0. A DLL implementada foi então registrada no servidor e é instanciada pela interface com o usuário, pela codificação em linguagem ASP, caracterizando uma aplicação *ActiveX*. A codificação completa do sistema encontra-se no apêndice B.

Na DLL, foram desenvolvidas duas classes, uma para validação do acesso do usuário e outra responsável pelo acesso aos dados das tabelas, pela definição da região, pelo processamento da inferência, pela geração da resposta ao usuário e pelo envio da resposta gerada via e-mail para o usuário.

IV.5.1 A Classe clsAcesso

A primeira classe definida, foi chamada de clsAcesso. Esta classe recebe o nome de *login*, a senha do usuário e passa a *string* de conexão, responsável pela conexão com o gerenciador de banco de dados. Esta classe pode retornar até quatro respostas diferentes, todas definidas através de um numero inteiro podendo variar de acordo com a tabela a seguir:

Tabela IV.31: Possíveis retornos da classe clsAcesso.

Se o usuário for inválido	O número retornado será -1
Se o usuário não tiver direito de acesso	O número retornado será 0
Se o usuário tiver direito de acesso	O número retornado será maior que 0 (zero)

- A classe `clsAcesso` é definida pela declaração das seguintes variáveis:

`ERRO_INVALIDO = -1`

`ERRO_SEMACCESSO = 0`

O método `ValidaLogin` é responsável pela verificação do usuário junto à tabela de usuários, como resposta podemos ter um número inteiro positivo maior que 1 e até 5, indicando o seu nível de acesso ao sistema. Caso o sistema responda com valor igual ou menor que 0 (zero), indica que o usuário possui algum tipo de problema, podendo ser:

`ERRO_INVALIDO = -1` ➔ indica que o usuário e a senha não foram encontrados;

`ERRO_SEMACCESSO = 0` ➔ indica que o usuário em questão não possui acesso.

Descrição do método:

Este método recebe como parâmetros o nome de *login* do usuário (`Login`), a senha do usuário (`senha`) e a *String* de conexão com a base de dados (`StringConexao`)

Saída: Inteiro variando entre -2 e 5.

`ValidaLogin (Login, Senha, StringConexao) as Integer`

Abrir a conexão com a base de dados;

Criar o `RecordSet Rs` para receber os resultados das consultas;

`Rs` ← Resultado da consulta de `Login` e `Senha`;

Se Número de registros > 0

Então

Mover para 1º registro;

`ValidaLogin` ← valor de acesso; // Podendo ser 0 (zero) sem acesso.

Senão

`ValidaLogin` ← `ERRO_INVALIDO`;

Fim_se;

Fim-`ValidaLogin`.

IV.5.2 A Classe `clsFuzzy`

Esta é a segunda e última classe definida, ela possui a declaração das variáveis globais necessárias aos métodos e uma série de métodos divididos em 4 partes distintas:

IV.5.2.1 - Métodos responsáveis pelo acesso às tabelas e pela montagem do grupo de regras e das funções de pertinência

IV.5.2.1.1 – Método CarregarAntecedentes()

Este método é responsável por montar o universo de discurso dos antecedentes e do conseqüente. É criado um vetor contendo todos os antecedentes e outro contendo o conseqüente, indicando a descrição, o valor inicial do intervalo e o valor final do intervalo do universo de discurso de cada variável lingüística.

Descrição do método:

CarregarAntecedentes() As Integer

Para cada Antecedente

Carregar na estrutura: Descrição, Intervalo inicial e Intervalo final;

Fim-para;

Carregar na estrutura conseqüente: Descrição, Intervalo inicial e Intervalo final;

Fim-CarregarAntecedentes.

IV.5.2.1.2 – Método CarregarRegras()

Este método é responsável por acessar a tabela de regras e transferi-la para o vetor de regras. Cada regra é montada a partir da tabela de regras no vetor Regras, este método é importante para aumentar a performance do sistema e permitir que novas regras sejam adicionadas, retiradas ou alteradas, bastando que para isso seja feita uma manutenção na tabela de regras, pois estas são sempre carregadas quando do momento do processamento. Retorna 1 caso tenha obtido sucesso.

Descrição do método:

CarregarRegras() As Integer

Contador \leftarrow 0;

Enquanto (não final de Regras)

Acrescentar elemento à estrutura de regras;

Para cada antecedente

Regras[Contador].Antecedente[antecedente] \leftarrow Termo_lingüístico_Antecedente;

Fim-para;

Regras[Contador]Conseqüente \leftarrow Termo_lingüístico_Conseqüente;

Contador \leftarrow Contador + 1;

Mover P/ próximo registro;

Fim_Enquanto;

NRegras \leftarrow Contador; 'Número de Regras

Fim-CarregarRegras.

IV.5.2.1.3 – Método MontaRegra()

Este método é responsável por montar a regra a ser usada na inferência com as partes dos termos. Acessa a tabela de funções de pertinência e carrega as partes referentes a cada termo da regra a ser montada. Como as funções estão em uma tabela, o sistema facilita qualquer alteração que necessite ser feita em qualquer parte de qualquer termo. Recebe como parâmetro o número da regra a ser montada e retorna 1 caso tenha obtido sucesso.

Descrição do método:

MontaRegra(RegraNumero As Integer) As Integer

Para cada Antecedente

Selecionar as funções de pertinência do termo lingüístico em questão;

Enquanto Houver função de pertinência

Adicionar elemento a estrutura do Antecedente;

Estrutura Antecedente \leftarrow (Descricao,Termoling,CoefAng,CoefLin,IntervInic,IntervFim)

Mover próximo registro

Fim-Enquanto;

Fim-Para;

Selecionar as funções de pertinência do termo lingüístico em questão do Conseqüente;

Enquanto Houver função de pertinência

Adicionar elemento à estrutura do Conseqüente;

Estrutura Conseqüente \leftarrow (Descricao,Termoling,CoefAng,CoefLin,IntervInic,IntervFim)

Mover próximo registro

Fim-Enquanto;

Fim-MontaRegra.

IV.5.2.2 - Métodos responsáveis pelos cálculos cartográficos e geração da região a partir do município base

IV.5.2.2.1 – Método GeraRegiao()

Este método é responsável por gerar a região, obtendo os municípios até 50km de distância, gerando a população, renda, conexão, urbanização e concorrência para a região. Recebe como parâmetro o código do município e retorna uma *String* contendo todos os municípios que fazem parte da região.

Descrição do método:

GeraRegiao(Municipio As String) As String

Regiao \leftarrow Municipio;

LatMun \leftarrow latitude Municipio;

LongMun \leftarrow longitude municipio;

```

Tpopulacao ← população município;
Trenda ← renda município;
Tconexao ← conexao município;
Turbanizacao ← urbanizacao município;
Tconcorrenca ← concorrencia município;
NumeroMunicipios ← 1;
Enquanto houver município
  Se código município <> município base
    Então
      Distancia ← CalculaDistancia(LatMunAtual, LongMunAtual, LatMun, LongMun);
      Se distancia <= 50 km
        Então
          Regiao ← região + Municipio;
          Tpopulacao ← Tpopulacao + população município atual;
          Trenda ← Trenda + renda município atual;
          Tconexao ← Tconexao + conexao município atual;
          Turbanizacao ← Turbanizacao + urbanizacao município atual;
          Tconcorrenca ← TConcorrenca + concorrencia município atual;
          NumeroMunicipios ← NumeroMunicipios + 1;
        Fim-se;
      Fim-se;
    Fim-enquanto;
  TRenda ← TRenda / TPopulacao;
  TConcorrenca ← TConcorrenca;
  TConexao ← TConexao / TPopulacao;
  TURbanizacao ← TURbanizacao / ContMunic;
  Ent2 ← TConexao;
  Ent3 ← TURbanizacao;
  Ent4 ← TPopulacao;
  Ent5 ← TConcorrenca;
  Ent6 ← TRenda;
  GeraRegiao ← GRegiao;
Fim-GeraRegiao;

```

IV.5.2.2.2 – Método CalculaDistancia(Lat1 As Double, Long1 As Double, Lat2 As Double, Long2 As Double)

Este método é responsável por calcular a distância entre o município base e os demais, para identificar aqueles que possuam distância inferior a 50Km para que possa gerar a região. Recebe como parâmetros as latitudes e longitudes de dois municípios (local da sede da prefeitura) e retorna a distância entre eles.

Descrição do método:

```

CalculaDistancia(Lat1 As Double, Long1 As Double, Lat2 As Double, Long2 As Double) As Double
  Pi ← 3.14159265358979;

```

Carregar Parâmetros do elipsoide (SAD-69);
 Efetuar Cálculos Auxiliares de f , $seMenor$ e $e2$;
 Carregar Latitudes e Longitudes (dos dois) pontos em FiA , FiB , $LambA$ e $LambB$;
 Transformar Graus para Radianos FiA , FiB , $LambA$ e $LambB$;
 Efetuar Passo 1 para cálculo de: $PsiA$, $PsiB$ e dl ;
 Repita
 $dl2 \leftarrow dl$;
 Efetuar Passo 2 - Cálculo de $dSigma$;
 Efetuar Passo3 - Cálculo de $AlfaAB$;
 Verificar Quadrante de $AlfaAB$;
 Efetuar Passo 4 - Cálculo de $Psi0$;
 Efetuar Passo 5 - Cálculo de $SigmaM$;
 Efetuar Passo 6 - Cálculo de dl ;
 Até que ($dl = dl2$) ' // retorna ao passo 2, se o dl antigo diferente do novo.
 Calcular a Convergência;
 Calcular a Distância s ;
 Transformar a distância s de metros para Km;
 Fim-CalculaDistancia;

IV.5.2.3 - Métodos responsáveis por formatar a resposta e enviar os resultados por e-mail

IV.5.2.3.1 – Método MontarResposta()

Este método é responsável por formatar a saída do sistema, gerando uma resposta em linguagem HTML para ser exibida pelo *browser* do usuário. Recebe como parâmetro a resposta da inferência *desfuzzyficada* e retorna uma *String* contendo a resposta formatada em HTML.

Descrição do método:

```

MontarResposta(d As Double) As String
  Para x de 0 até Total_Termos_Consequente - 1
    Para y de x + 1 To Total_Termos_Consequente
      Se PercentuaisResp(x) < PercentuaisResp(y)
        Então
          Temp1 ← PercentuaisResp(x);
          PercentuaisResp(x) ← PercentuaisResp(y);
          PercentuaisResp(y) ← Temp1;
          Temp2 ← ConsequentesResp(x);
          ConsequentesResp(x) ← ConsequentesResp(y);
          ConsequentesResp(y) ← Temp2;
        Fim-Se;
      Fim-Para;
    Fim-Para;
  
```

```

For x de 0 até UBound(ConsequentesResp)
    StrResult ← StrResult + "<p>" + "A Região escolhida é de " + ConsequentesResp(x) +
    " Potencialidade com " + CStr(PercentuaisResp(x)) + " pontos percentuais" + Chr(13)
Fim-Para;
StrResult ← StrResult + "<p>A Região é formada pelos Municípios: </p>" + RegiaoBase;
Resp ← EnviaMail("StrResult");
MontarResposta ← StrResult;
Fim-MontarResposta;

```

IV.5.2.3.2 – Método EnviaMail()

Este método é responsável por enviar para o usuário um *e-mail* contendo a resposta do sistema para que o usuário possa armazenar e consultar quando quiser. Recebe como parâmetros o conteúdo do *e-mail* e o endereço eletrônico do usuário e retorna 1 caso tenha obtido sucesso.

Descrição do método:

```

EnviaMail (StrMail As String, MailTo As String) As Integer
    Criar objeto myMail como novo ("CDONTS.NewMail");
    myMail.From ← sicsu@globo.com;
    myMail.To ← MailTo;
    myMail.CC ← orlando@eng.uerj.br;
    myMail.Subject ← "Resultado gerado pelo Sistema SIGAPI.";
    myMail.Body ← StrMail;
    myMail.BodyFormat ← 0;
    myMail.MailFormat ← 0;
    myMail.Importance ← 2;
    myMail.Value("Reply-To") ← "Sicsú<sicsu@globo.com>";
    myMail.Send;
    Destruir Objeto myMail;
    EnviaMail ← 1;
Fim-EnviaMail;

```

IV.5.2.4 - Métodos responsáveis por efetuar a inferência (Motor de Inferência do Sistema)

IV.5.2.4.1 – Método Inicio()

Este método é responsável por controlar a execução do sistema. A partir desse método todos os outros métodos são então instanciados. Recebe como parâmetros o código do município, a *String* de conexão com a base de dados e a identificação do usuário. Este método retorna ao servidor ASP a saída formatada do resultado da inferência.

Descrição do método:

```
Inicio(IdMunicipio As String, StringConexao As String, User as String) As String
  Criar objetos de conexão ao banco de dados;
  Usuário ← User;
  Gerar a Região a partir do município base;
  Inicio ← FazerInferencia;
Fim-Inicio.
```

IV.5.2.4.2 – Método FazerInferencia()

Este método é responsável por controlar a execução do sistema. A partir desse método todos os outros métodos são então instanciados. Recebe como parâmetros o código do município, a *String* de conexão com a base de dados e a identificação do usuário. Este método retorna ao servidor ASP a saída formatada do resultado da inferência.

Descrição do método:

```
FazerInferencia() As String
  Resp ← CarregarAntecedentes();
  Resp ← FuzzificarEntradas(Ent2, Ent3, Ent4, Ent5, Ent6);
  Resp ← CarregarRegras();
  Resp ← MontaRegra(0); 'Montar primeira regra
  Se Resp = 1
    Então
      Para x de 1 até Total_de_antecedentes
        Ent(x) ← FuncaoMinima(Antecedente, Entrada);
      Fim-para;
      Alpha ← Minimo(Minimo(Supremo(fAux2), Minimo(Supremo(fAux3),
        Supremo(fAux4))), Minimo(Supremo(fAux5), Supremo(fAux6)));
      FuncaoAlpha(0).IntervInic ← Consequente(0).IntervInic;
      FuncaoAlpha(0).IntervFim ← Consequente(UBound(Consequente)).IntervFim;
      FuncaoAlpha(0).CoefAng ← 0;
      FuncaoAlpha(0).CoefLin ← Alpha;
      Resp ← FuncaoMinima(FuncaoAlpha, Consequente);
      Para x de 1 até Total_de_Regras - 1
        Resp ← MontaRegra(x); 'Monta próxima regra
        Se Resp = 1
          Então
            Para x de 1 até Total_de_antecedentes
              Ent(x) ← FuncaoMinima(Antecedente, Entrada);
            Fim-para;
            Alpha ← Minimo(Minimo(Supremo(fAux2), Minimo(Supremo(fAux3),
              Supremo(fAux4))), Minimo(Supremo(fAux5), Supremo(fAux6)));
            FuncaoAlpha(0).CoefAng ← 0;
            FuncaoAlpha(0).CoefLin ← Alpha;
            fAux2 ← Resposta;
            Resp ← FuncaoMinima(FuncaoAlpha, Consequente);
            fAux1 ← Resposta;
            Resp ← FuncaoMaxima(fAux1, fAux2);
```

```

    Fim-se;
    Fim-para;
    Fim-se;
    RespDesFuzzi  $\leftarrow$  DesFuzzificar(Resposta);
    FazerInferencia  $\leftarrow$  MontarResposta(RespDesFuzzi);
    Fim-FazerInferencia.

```

IV.5.2.4.3 – Método FuzzyficarEntradas()

Este método é responsável por preparar as entradas para o motor de inferência do sistema. Recebe como parâmetros os cinco valores das entradas do sistema de inferência e as converte em funções de pertinência e retorna 1 caso tenha obtido sucesso.

Descrição do método:

```

FuzzificarEntradas(e2 As Double, e3 As Double, e4 As Double, e5 As Double, e6 As Double)
As Integer
    Para x de 2 até 6
        Estrutura_Intervalo_inicial  $\leftarrow$  e(x);
        Estrutura_Intervalo_final  $\leftarrow$  e(x);
        Estrutura_Coeficiente_Linear  $\leftarrow$  1;
        Estrutura_Coeficiente_Angular  $\leftarrow$  0;
    Fim-para;
    FuzzificarEntradas  $\leftarrow$  1;
    Fim- FuzzificarEntradas.

```

IV.5.2.4.4 – Método AvaliaFuncao()

Este método é responsável por avaliar a função de pertinência em um determinado ponto. Recebe como parâmetros a função a ser avaliada e o valor do ponto a avaliar e retorna o valor obtido na avaliação.

Descrição do método:

```

AvaliaFuncao(f() As FuncaoAuxiliar, Ponto As Double) As Double
    Para x de 0 até Quantidade de Partes da Função
        Se (Ponto >= f(x).IntervInic E Ponto <= f(x).IntervFim)
            Então
                AvaliaFuncao  $\leftarrow$  f(x).CoefAng * Ponto + f(x).CoefLin;
            Fim-se;
        Fim-Para;
    Fim-AvaliaFuncao.

```

IV.5.2.4.5 – Método Minimo()

Este método é responsável por verificar o menor entre dois valores. Recebe como parâmetros dois valores e n1 e n2 e retorna o valor do menor.

IV.5.2.4.6 – Método Maximo()

Este método é responsável por verificar o maior entre dois valores. Recebe como parâmetros dois valores e n1 e n2 e retorna o valor do maior.

IV.5.2.4.8 – Método Supremo()

Este método retorna o valor supremo de uma função de pertinência. Recebe como parâmetro uma função de pertinência e retorna o valor supremo da função.

Descrição do método:

Supremo(F1() As FuncaoAuxiliar) As Double

NTermos \leftarrow Tamanho_Estrutura(F1);

Se NTermos > 0

Então

Para Contador de 0 até NTermos

Supremo \leftarrow Maximo(Supremo, Maximo(F1(Contador).CoefAng *

F1(Contador).IntervInic + F1(Contador).CoefLin, F1(Contador).CoefAng *

F1(Contador).IntervFim + F1(Contador).CoefLin));

Fim-Para;

Senão

Supremo \leftarrow F1(0).CoefLin;

Fim-Se;

Fim-Supremo;

IV.5.2.4.9 – Método DesFuzzyficar()

Este método *desfuzzyfica* a saída gerada pelo motor de inferência do sistema, através do método do centróide. Recebe como parâmetro uma função de pertinência e retorna o valor *desfuzzyficado*.

IV.5.2.4.10 – Métodos: FuncaoMinima() e FuncaoMaxima()

O procedimento FuncaoMinima inicialmente verifica se um dos parâmetros de entrada (f1 ou f2) é constituído de um ponto isolado, pois, nesse caso, a resposta é trivial, isto é, a função mínima também será uma função de ponto isolado, cujo valor é o mínimo entre as avaliações de f1 e f2 nesse ponto isolado. A FuncaoMaxima faz o mesmo, sendo que fornece como resposta o máximo entre as avaliações de f1 e f2.

Excluindo-se o caso trivial, o procedimento FuncaoMinima testa uma parte de cada uma das entradas f1 e f2 por vez. Como f1 e f2 são diferentes, obviamente não possuem a mesma quantidade de partes e essas partes não ocupam o mesmo intervalo. As entradas f1 e f2 possuem apenas o mesmo universo de discurso o que leva também a função mínima a estar definida dentro desse mesmo universo de discurso.

A função mínima é montada parte a parte pelo procedimento FuncaoMinima dentro de um *loop* que tem início comparando-se as primeiras partes de f1 e f2. A primeira parte da função mínima é a menor das primeiras partes de f1 e f2 (menor intervalo), a segunda parte seria o restante da maior primeira parte entre f1 e f2 e assim por diante. Para esclarecer melhor como funciona o procedimento, veja a figura IV.24.

A figura IV.24-a mostra um exemplo da função de entrada f1, enquanto a figura IV.24-b mostra um exemplo da função de entrada f2. A função f1 possui 3 partes, a primeira com equação de reta y11 (intervalo de 0 a 3), a segunda com equação de reta y12 (intervalo de 3 a 8) e a terceira com a equação y13 (intervalo de 8 a 10). A função f2 também tem 3 partes, ou seja, a primeira com equação de reta y21 (intervalo de 0 a 2), a segunda com equação y22 (intervalo de 2 a 4) e a terceira com equação y23 (intervalo de 4 a 10).

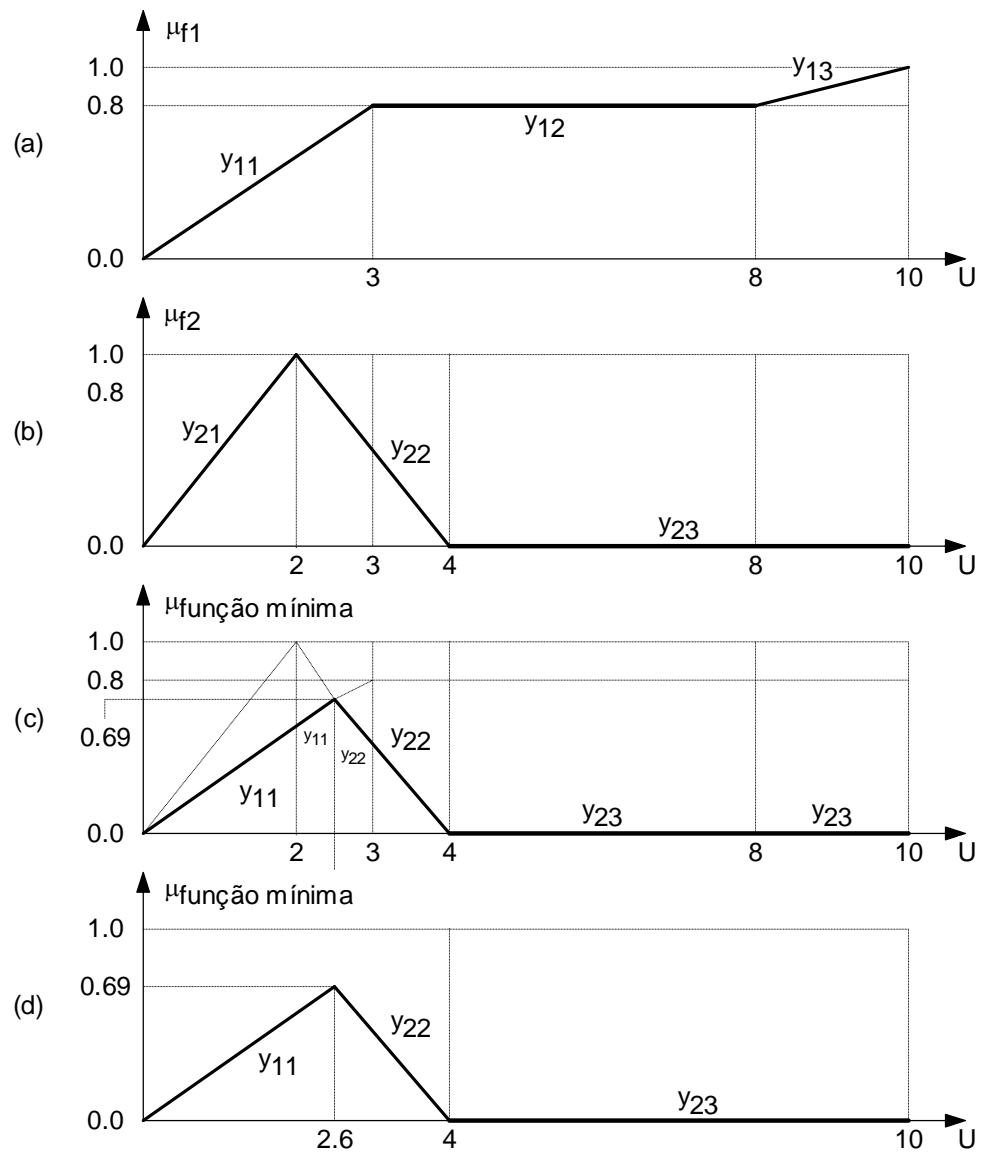


Figura IV.24: Um exemplo de funções de entrada e saída para o procedimento FuncaoMinima.

A primeira parte da função mínima é dada pelo intervalo de 0 a 2, pois esse primeiro intervalo de f_2 é menor do que o primeiro intervalo de f_1 (0 a 3); entretanto a equação do primeiro segmento de reta da função mínima será y_{11} de f_1 pois no intervalo de 0 a 2, $y_{11} < y_{21}$ (vide figura IV.4-c). O programa escolhe qual das duas equações de reta será usada, testando o valor das duas funções nos extremos do intervalo. Nessa primeira parte, ambas as funções são zero no ponto zero do universo de discurso; e no ponto 2, y_{11} é igual a 0,533, enquanto y_{21} é igual a 1,0 e, portanto, $y_{11} < y_{21}$. A figura IV.25 resume todos os casos possíveis de comparação de segmentos de reta em um determinado intervalo.

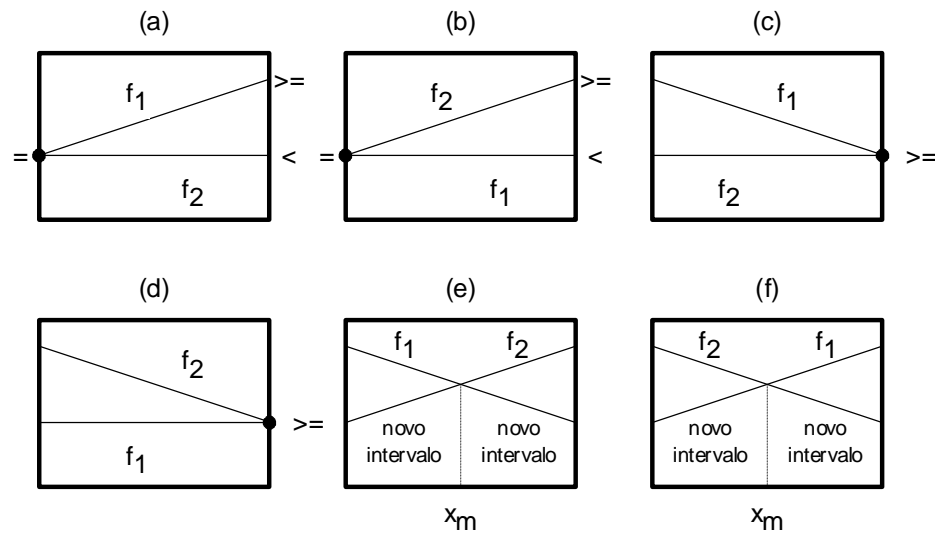


Figura IV.25: Todos os casos possíveis de comparação entre os seguimentos de reta de um dado intervalo das funções de entrada dos procedimentos FuncaoMinima e FuncaoMaxima.

Os casos (e) e (f) da figura IV.25 retratam uma situação em que um dado intervalo escolhido pelo procedimento FuncaoMinima é quebrado em dois, já que a função mínima deve ter a contribuição das duas retas. No caso (e), por exemplo, a equação da reta seria a de f_1 até o ponto x_m e, depois, a equação de f_2 . O intervalo da segunda parte do exemplo da figura IV.24 (2 a 3) recai nesse caso, pois $y_{11}(2) < y_{22}(2)$ e $y_{11}(3) > y_{22}(3)$, caracterizando portanto, um cruzamento que ocorre no ponto 2,6. Sendo assim, a segunda parte da função mínima teria intervalo de 2 até 2,6 e a terceira parte teria intervalo de 2,6 a 3.

A quarta parte da função mínima teria intervalo de 3 a 4 com a equação y_{22} de f_2 , a quinta parte teria intervalo de 4 a 8 com a equação y_{23} de f_2 e a sexta parte teria também a equação y_{23} de f_2 . Como pode ser observado pela figura IV.24-c, ao longo da definição das equações das partes da função mínima ocorrem as mesmas escolhas de equações em alguns intervalos adjacentes. Para que não existam diversos intervalos iguais desnecessariamente, o procedimento FuncaoMinima agrupa automaticamente em uma única parte os intervalos adjacentes com a mesma equação. A figura IV.24-d mostra a função mínima obtida após terem sido agrupados os intervalos iguais.

O método FuncaoMaxima tem um algoritmo bem semelhante ao do método FuncaoMinima. A diferença surge nos momentos em que é escolhida qual equação será definida para uma parte da resposta, pois no caso da função máxima toma-se a maior função ao invés da menor.

Ao observar o código do programa, mostrado no apêndice B, dos métodos

FuncaoMinima e FuncaoMaxima, nota-se que algumas das avaliações de valores ou comparações, comentadas aqui, não são feitas de maneira direta, mas sim através de referência a valores pequenos, por exemplo, ao invés de se comparar $f1e = f2e$, faz-se o valor absoluto de $(f1e - f2e)$ e verifica-se é menor que $1.0E-9$. Isso se deve ao fato dos cálculos com números reais apresentarem erro de truncamento.

IV.6 Testes Realizados

O procedimento de testes foi feito utilizando o sistema pronto. Foram selecionados os municípios de Saquarema e Três Rios, pelas características das respostas recebidas. No primeiro caso obtivemos uma resposta única e no segundo caso, obtivemos uma resposta composta, com duas potencialidades distintas e suas respectivas percentagens.

IV.6.1 Teste com o Município de Saquarema

- Município base escolhido: Saquarema.
- Número de municípios adicionados à região gerada: 9 municípios.
- Entradas:

Tabela IV.32: Valores de Entrada no sistema (Saquarema).

Variável Lingüística	Valor de Entrada
Conexão (Ent1)	171 kbps
Urbanização (Ent2)	57 %
População (Ent3)	507228 habitantes
Concorrência (Ent4)	1,4 provedores / município
Renda (Ent5)	R\$ 670,00

- Termos lingüísticos gerados (com pertinência acima de 0 (zero)):

Tabela IV.33: Termos Lingüísticos (Saquarema).

Variável Lingüística	Termos Lingüísticos
Conexão	Larga
Urbanização	Urbanizada / Muito
População	Extremamente
Concorrência	Sem Concorrência / Com Concorrência
Renda	Baixa

- Regras sensibilizadas:

Tabela IV.34: Pertinências das regras sensibilizadas (Saquarema).

Id	Conexão	Urbanizacao	Populacao	Concorrencia	Renda	Potencialidade
352	Larga (1.0)	Urbanizada (0.72)	Extremamente (1.0)	Sem Conc. (0.72)	Baixa (1.0)	Media (0.72)
397	Larga (1.0)	Muito (0.28)	Extremamente (1.0)	Sem Conc. (0.72)	Baixa (1.0)	Media (0.28)
355	Larga (1.0)	Urbanizada (0.72)	Extremamente (1.0)	Com Conc. (0.28)	Baixa (1.0)	Media (0.28)
400	Larga (1.0)	Muito (0.28)	Extremamente (1.0)	Com Conc. (0.28)	Baixa (1.0)	Media (0.28)

Como podemos observar a regra mais sensibilizada foi a regra 352 com pertinência igual a 0.72, seguida das demais regras, com pertinência igual a 0.28, gerando como resposta a potencialidade Média com 100%.

A resposta apresentada pelo sistema pode ser vista na figura IV.26 a seguir, onde pode ser constatado que de fato a potencialidade Média ficou com 100%.

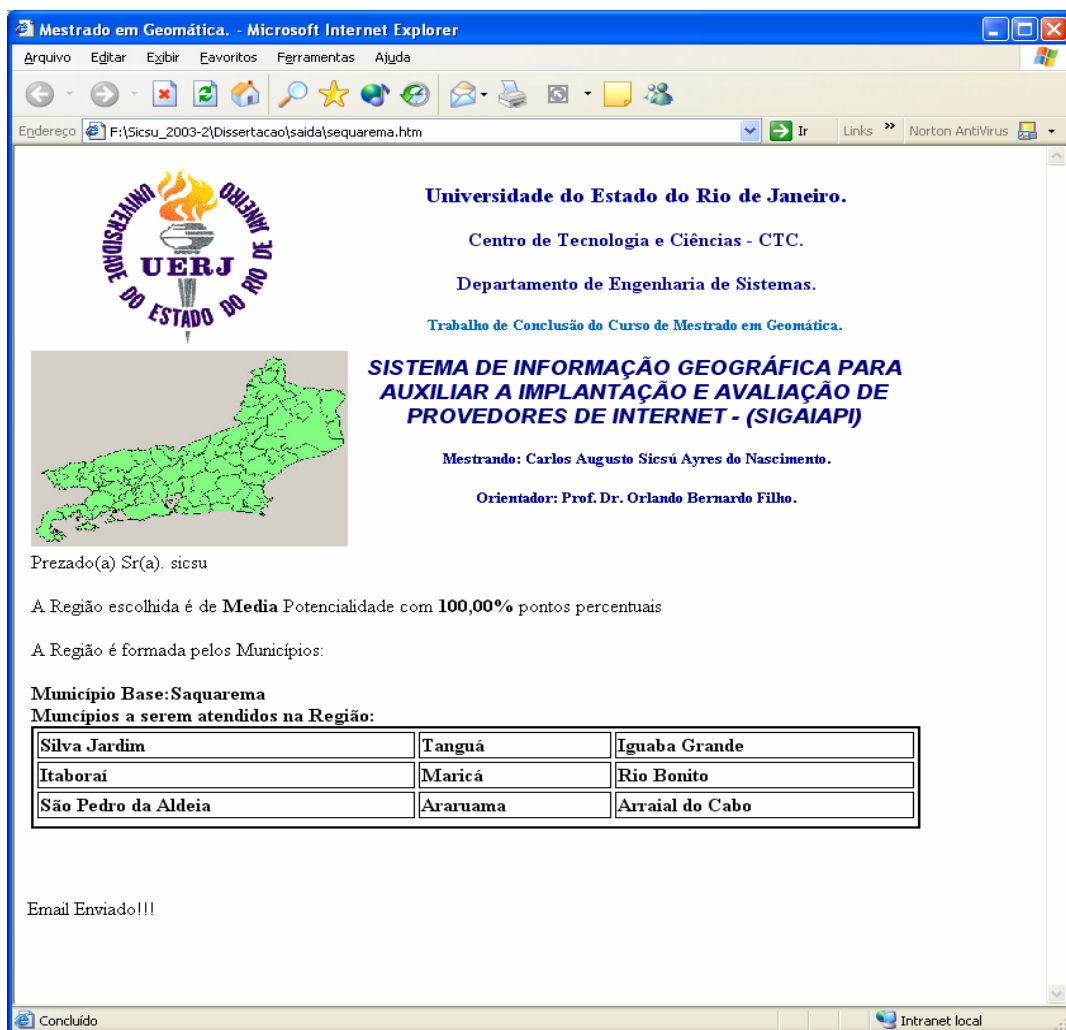


Figura IV.26: Resposta do SIGAI-API para o teste IV.6.1.

IV.6.2 Teste com o Município de Três Rios

- Município base escolhido: Três Rios.
- Número de municípios adicionados á região gerada: 10 municípios.
- Entradas:

Tabela IV.35: Valores de Entrada no sistema (Três Rios).

Variável Lingüística	Valor de Entrada
Conexão (Ent1)	308 kbps
Urbanização (Ent2)	69 %
População (Ent3)	512942 habitantes
Concorrência (Ent4)	2,45 provedores / município
Renda (Ent5)	R\$ 1.831,00

- Termos lingüísticos gerados (com pertinência acima de 0 (zero)):

Tabela IV.36: Termos Lingüísticos (Três Rios).

Variável Lingüística	Termos Lingüísticos
Conexão	Larga
Urbanização	Urbanizada / Muito
População	Extremamente
Concorrência	Sem Concorrência / Com Concorrência
Renda	Baixa / Média

- Regras sensibilizadas:

Tabela IV.37: Pertinências das regras sensibilizadas (Três Rios).

Id	Conexão	Urbanizacao	População	Concorrencia	Renda	Potencialidade
397	Larga (1.0)	Muito (0.76)	Extremamente (1.0)	Sem Conc. (0.51)	Baixa (0.17)	Media (0.17)
398	Larga (1.0)	Muito (0.76)	Extremamente (1.0)	Sem Conc. (0.51)	Media (0.83)	Boa (0.51)
400	Larga (1.0)	Muito (0.76)	Extremamente (1.0)	Com Conc. (0.49)	Baixa (0.17)	Media (0.17)
401	Larga (1.0)	Muito (0.76)	Extremamente (1.0)	Com Conc. (0.49)	Media (0.83)	Media (0.49)
352	Larga (1.0)	Urbanizada (0.24)	Extremamente (1.0)	Sem Conc. (0.51)	Baixa (0.17)	Media (0.17)
353	Larga (1.0)	Urbanizada (0.24)	Extremamente (1.0)	Sem Conc. (0.51)	Media (0.83)	Media (0.24)
355	Larga (1.0)	Urbanizada (0.24)	Extremamente (1.0)	Com Conc. (0.49)	Baixa (0.17)	Media (0.17)
356	Larga (1.0)	Urbanizada (0.24)	Extremamente (1.0)	Com Conc. (0.49)	Media (0.83)	Media (0.24)

Como podemos observar a regra mais sensibilizada foi a regra 398 com pertinência igual a 0.51, seguida da regra 401 com pertinência igual a 0.49, gerando como respostas possíveis a potencialidade respectivamente Boa(~51%) e Média(~49%).

A resposta apresentada pelo sistema encontra-se na figura IV.27 a seguir.

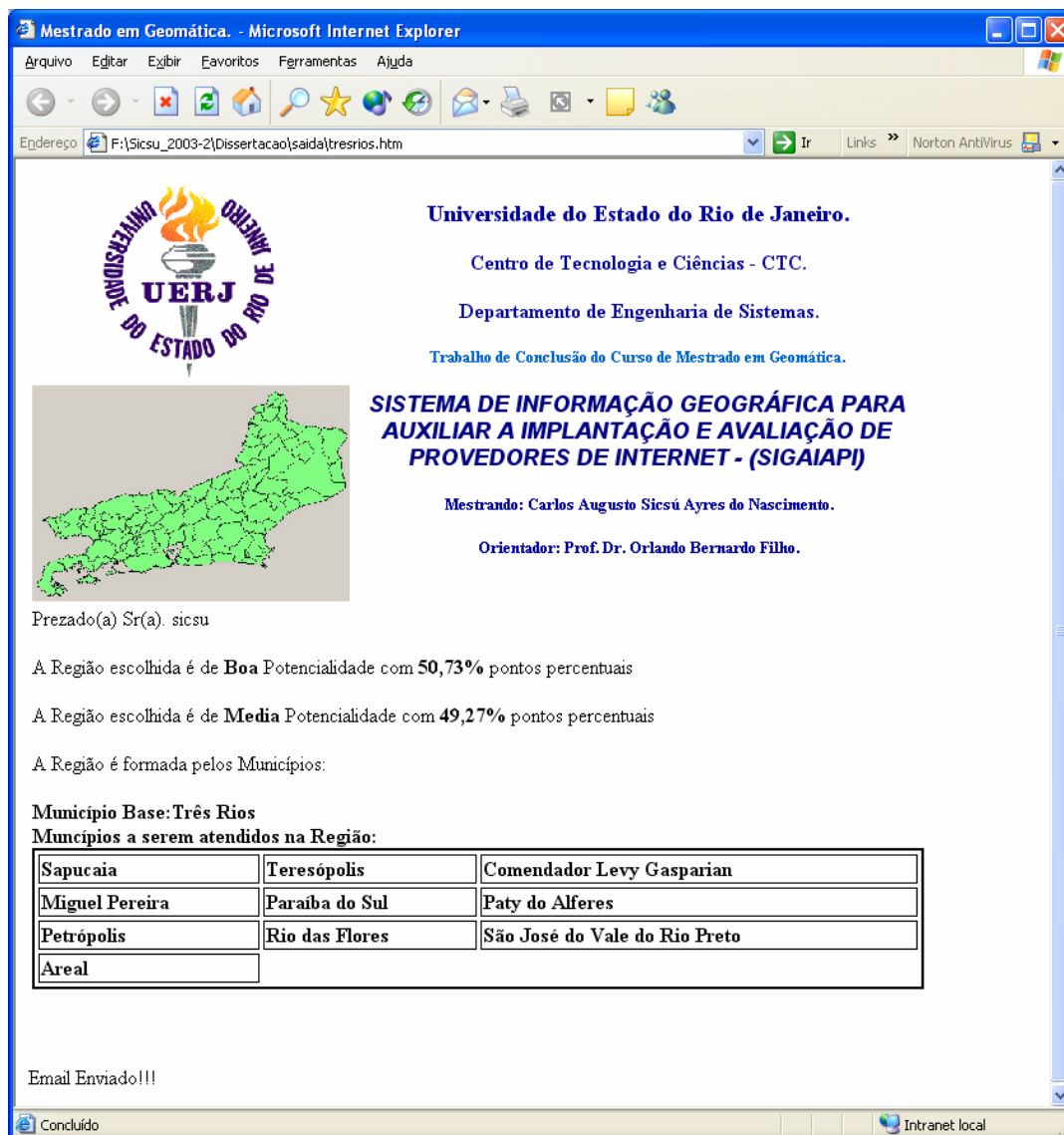


Figura IV.27: Resposta do SIGAI-API para o teste IV.6.2.

IV.7 Comentários

Neste capítulo, observamos a implementação do sistema. Vimos como foram montadas as partes das funções de pertinência e as regras. Foram também apresentadas as soluções para codificação do sistema. Por final foram realizados testes que comprovaram o funcionamento correto do SIGAI-API. Comprovamos também que as respostas geradas estão corretas e de acordo com o sistema implementado.

CAPÍTULO V

Conclusões

O SIGAI-API, da maneira como foi idealizado, possui uma boa funcionalidade, no entanto, o mesmo poderá contar com mais temas para compor as informações geográficas, de modo a obter uma base de dados mais sólida, mais próxima ainda da realidade do mercado do estado, o que irá favorecer uma análise mais criteriosa e confiável.

Os objetivos esperados foram alcançados, uma vez que o sistema mostrou-se bastante eficaz dentro do escopo em que foi especificado. Todas as técnicas e tecnologias necessárias para o seu desenvolvimento foram aplicadas e atenderam ao que era esperado. A técnica de uso do *ActiveX* foi dominada, permitindo que novas aplicações destinadas ao uso com a Internet sejam desenvolvidas com maior facilidade.

O sistema mostrou-se capaz de auxiliar a tomada de decisão no processo de implantação de novos provedores de Internet, informando se o local escolhido como base (município base) não apenas possui ou não potencialidade na região, mas também é capaz de quantificar as potencialidades. Como apresentado no capítulo anterior, os testes provaram a eficácia do sistema e do motor de inferência criado. As respostas geradas pelo sistema foram conferidas passo a passo e verificamos que as mesmas encontram-se corretas de acordo com as especificações do sistema e os resultados obtidos eram os resultados esperados.

O sistema mostrou-se também muito flexível, pois, alguns ajustes nas funções de pertinência foram necessários. Por exemplo, a variável conexão foi alterada. Para ajustá-la à realidade do estado do Rio de Janeiro, foi necessário diminuir não só o número de termos lingüísticos, mas, também foi necessário alterar o universo de discurso dessa variável. Com o uso de uma tabela para armazenar as partes das funções, o sistema não precisou ser reescrito, bastou apenas manipular a tabela funções do banco de dados para efetuar estes ajustes. O mesmo procedimento pode ser feito quanto à necessidade de ajuste de qualquer variável lingüística.

Os ajustes no sistema podem ser feitos também sobre as regras, se em algum momento for necessário alterar alguma regra, basta alterá-la na tabela regras.

A base de dados do censo do IBGE e o emprego da lógica nebulosa podem servir para se desenvolver outros serviços de implantação de outros negócios no estado do Rio de Janeiro e não apenas para esta aplicação de implantação e avaliação de provedores de Internet.

Uma vez que os dados sobre a população do Rio de Janeiro e das características

do estado são bastantes gerais, é possível aproveitar boa parte da implementação do SIGAI-API para rapidamente elaborar um protótipo de outra aplicação, usando a análise com inferência nebulosa para dar suporte à decisão de investimentos no estado do Rio de Janeiro.

Tal núcleo de implementação pode vir a ser usado até mesmo como documentação da realidade do mercado do estado do Rio de Janeiro, se o mesmo for periodicamente atualizado.

O sistema pode ser facilmente alterado para atender a qualquer tipo de investimento a ser feito em um município ou região. O poder público, quando do momento de novos investimentos em infra-estrutura, saúde, educação entre outras necessidades poderia usar o sistema com poucas alterações para a implantação de novas escolas, hospitais, postos de saúde, estradas, postos policiais entre outros.

Apesar de o sistema apresentar as respostas previstas, poderíamos fazer algumas alterações. As sugestões seriam:

1. Uma delas seria trocar a forma de uso do sistema. Em vez de escolhermos o município base e esperarmos a sua potencialidade, poderíamos inverter a forma de uso e ao executarmos o sistema, esperar que o mesmo informe o município base e a região que possui a maior potencialidade neste momento. Dessa forma, poderíamos obter a região mais adequada à implantação de um novo provedor de Internet no Estado do Rio de Janeiro.
2. Outra alteração possível seria a criação de novos sistemas bastante semelhantes que poderiam informar o lugar mais carente (município ou região) para a implantação de novos hospitais, postos de saúde, escolas, entre outros.

A grande dificuldade encontrada no desenvolvimento do sistema foi a obtenção dos dados para a base dos municípios. Foram necessárias várias buscas e montagens de tabelas a partir de diversas fontes distintas e em formatos distintos. Se tivéssemos um acesso mais fácil a estes dados como a disponibilidade desses dados via internet, o sistema poderia então ser modificado para o uso de banco de dados heterogêneos. Com o uso desta tecnologia e disponibilidade dos dados na Internet, o sistema poderia obter os dados *on-line*, sendo desnecessária a atualização dos mesmos. Assim o sistema poderia atender melhor e com maior eficácia aos seus usuários.

Outra oportunidade seria o armazenamento pelo sistema dos dados antigos, de sensores passados. Deste modo, o sistema poderia ser alterado para não só a indicar uma região com potencialidade neste momento, mas ajudar a tomar decisões de implantações futuras, informando as regiões que possam vir a ter potencialidade.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] FERRARI, R., *Viagem ao SIG – Planejamento Estratégico, Implantação e Gerenciamento de Sistemas de Informação Geográfica*, SAGRES EDITORA, 1997.
- [2] ROCHA, C. H. B., *Geoprocessamento: Tecnologia Transdisciplinar*, ED. DO AUTOR, 2000.
- [3] SILVA, A. de B., *Sistemas de Informações Geográficas Geo-referenciadas: Conceitos e Fundamentos*, EDITORA DA UNICAMP, 1999.
- [4] FILHO, O. B., NASCIMENTO, C. A. S. A. do , RIBEIRO J. A., *Proceedings of the VI Brazilian Conference on Neural Networks - VI Congresso Brasileiro de Redes Neurais pp. 61–66*, 2003.
- [5] BOMFORD, G., *Geodesy*, Oxford University Press., 3^a Ed., 1977.
- [6] BURROUGH, P. A., *Principles of Geographical Information Systems for Land Resources Assessment*, Oxford University Press., 1986.
- [7] ZIMMERMANN, H. J., (1985). *Fuzzy Set Theory - and Its Applications*, Kluwer-Nijhoff Publishing, 1985.
- [8] KANDEL, A., *Fuzzy Mathematical Techniques with Applications*, Addison-Wesley Publishing Co., 1986.
- [9] MENDEL, J. M., *Fuzzy Logic Systems for Engineering: A Tutorial*, Proceedings of the IEEE, vol. 83, nº 3, março, 1995.
- [10] ZADEH, L. A., *Fuzzy Sets*, Information and Control, 1965.
- [11] TANSCHKEIT, R., *Controle Nebuloso*, Anais do 9º Congresso Brasileiro de Automática, setembro, 1992.

- [12] ZADEH, L. A., *Fuzzy Sets as a Basis for a Theory of Possibility*, Fuzzy Sets and Systems 1, 1978.
- [13] DUBOIS, D. e PRADE, H., *Fuzzy Sets and Systems, Theory and Applications*, Academic Press, Orlando, 1980.
- [14] TERANO, T., ASAI, K. e SUGENO, M., *Fuzzy Theory and its Applications*, Academic Press, Boston, 1987.
- [15] LEE, C. C., *Fuzzy Logic in Control Systems: Fuzzy Logic Controller*, parts I & II, IEEE Transactions on Systems, Man and Cybernetics, vol. 20, nº 2, 1990.
- [16] ZADEH, L. A., *The Role of Fuzzy Logic in the Management of Uncertainty in Expert Systems*, Fuzzy Sets and Systems 11, 1983.
- [17] ZADEH, L. A., *Fuzzy Logic*, Computer, 1988.
- [18] FILHO, O. B., *Verificação de Protocolos de Comunicação com Lógica Nebulosa*. Tese de Doutorado, Programa de Engenharia Elétrica, COPPE/UFRJ, 1999.
- [19] SILVA, A. P. da, *Escalamento Heurístico Baseado em Lógica Difusa*, Tese de Mestrado, Programa de Engenharia Elétrica, COPPE/UFRJ, 1995.
- [20] OLIVEIRA, C., *Dicionário Cartográfico*, FIBGE, 1993.
- [21] http://www.anatel.gov.br/Tools/frame.asp?link=/biblioteca/releases/2003/anexo_ii_release_27_06_2003.pdf, consultado em 02/12/2003.
- [22] <http://acessogratis.com.br/internet-gratis/qos-provedores-gratuitos.asp>, consultado em 02/12/2003.
- [23] http://www.anatel.gov.br/Tools/frame.asp?link=/biblioteca/releases/2003/release_05_09_2003_3.pdf, consultado em 04/12/2003.

- [24] http://www.anatel.gov.br/Tools/frame.asp?link=/biblioteca/regulamentos/1998/regulamento_resolucao85_1998.pdf, consultado em 10/12/2003.
- [25] JEPSON, B., *Programando Banco de Dados em Java*, MAKRON BOOKS, 1997.
- [26] MARCON, A. M. e NEVES, D., *Aplicações e Banco de Dados para Internet*, EDITORA ÉRICA, 1999.
- [27] FREEZE, W. S., *SQL Guia de Referência do Programador*, EDITORA CIÊNCIA MODERNA, 1998.
- [28] CÔRTEZ, P. L. e SHIRAISHI, K., *MS-SQL 7.0 Server e Visual Basic 6.0*, EDITORA ÉRICA, 3ª Ed., 2002.
- [29] PRADO N. DEL P., *Acessando SQL Server com Visual Basic 6.0*, ADVANCED EDITORA, 1998.
- [30] CRÓSTA, A. P., *Processamento Digital de Imagens de Sensoriamento Remoto*, Instituto de Geociências, UNICAMP, 1992.
- [31] BENYON, D., *Information and Data Modelling*, Blackweell Scientific Pub. OXFORD, 1990.
- [32] CALKINS, H. W. e TOMLINSON, R. F., *Geographic Information System: Methods and Requirements for Landuse Planning*, Program USGS, RESTON, 1977.
- [33] DUEKER, K. J., *Land Resource Information Systems: A Review of Fifteen Years of Experience*, Geoprocessing 1, pp. 105-108, 1979.
- [34] OZEMOY, V. M., EMITH, D. R. e SICHERMAN, A., *Evaluating Computerized Geographic Information Systems Using Decision Analysis*, Intefaces 11, pp. 92-98, 1981.

- [35] BURROUGH, P. A., *Principles of Geographical Information System: Methods and Requirements for Landuse Planning.*, Clarendon, OXFORD, 1986.
- [36] DEVINE, H. A. e FIELD, R. C., *The Gist of GIS.*, Journal of Forestry, 08, pp. 17-22, 1986.
- [37] OPERSHAW S., *Guest Editorial: An Automated Geographical Analysis System.*, Env. And Plan. A, vol 19, pp. 431-436, 1987.
- [38] DEPARTMANET OF ENVIRONMENT (DOE), *Handling Geographic Information.*, HMSO, London, 1987.
- [39] SMITH, T. R., MENON, S., STAR, J. L. E ESTES, J. E., *Requirements and Principals for the Implementation and Construction of Large-Scale Geographic Information Systems.*, Int. Jour. of Geog. Inf Sys. 1:13-31, 1987.
- [40] PARKER, H. D., *The unique Qualities of a Geographic Information Systems: A Commentary.*, Protog. Eng. and Rem. Sen. 54(11):1547-1549, 1988.
- [41] COWEN, D. J., *GIS versus CAD versus DBMS: What are the differences.*, Photog. Eng. and Rem. Sen. 54: 1551-4, 1988.
- [42] FICC – TECNHNOLGY WORKING GROUP, *A Process for evaluating Geographic Information Systems.*, Technical Report 1, USGS Open-File Report, pp. 88-105, 1988.
- [43] PARENT, P. J., *Geographic Information System: Evolution Academic Involvement and Issus Arising from proliferation of Information.*, Master Teses University of California, 1988.
- [44] HANIGAN, F., *GIS by any other name is still...* The GIS Forum 1: 6, 1988.
- [45] ARONOFF, S., *Geographic Information Systems: a Management Perspective*, WDL Publications, Ottawa, Canadá , 1989.

- [46] CARTER, J. R., *On Defining the Geographic Information System. In: Ripple (ed.). Fundamentals of Geographical Information Systems: a Compendium*, ASPRS/ACSM, Falls Church, Virginia, 1989.
- [47] KOSHKARIOV, A. V., TIKUNOV, V. S. e TROKIMOV, A. M., *The current state and main trends in the development of Geographical Information Systems in the USSR*, Int. Jour. Of Geog. Inf. Sys. 3(3): 257-272, 1989.
- [48] STAR, J. L. e ESTES J. E., *Geographic Information Systems: An Introduction.*, Prentice Hall, 1990.
- [49] GOODCHILD, M. F., *Spatial Analysis with GIS: Problems and Prospects GIS/LIS*, The Inforum Atlanta, Georgia, pp. 40-48, 1991.
- [50] KAUFMAN, S. Jr., PERKINS J. e FLEET D., *Programação Activex – Aprenda em 21 Dias.*, EDITORA CAMPUS, 1997.
- [51] APPLEMAN D., *Visual Basic 5.0 – Desenvolvendo Componentes ActiveX*, MAKRON BOOKS, 1999.
- [52] RAMALHO, J. A., *HTML Avançado*, MAKRON BOOKS, 1997.
- [53] IBGE, *Base de Informações Municipais 2ª Edição*, IBGE, 2000.

APÊNDICES

Apêndice A – Regras

NRegra	Conexão	Urbanização	População	Concorrência	Renda	Potencialidade
1	Estreita	Levemente	Levemente	Sem Concorrença	Baixa	Baixa
2	Estreita	Levemente	Levemente	Sem Concorrença	Media	Baixa
3	Estreita	Levemente	Levemente	Sem Concorrença	Alta	Baixa
4	Estreita	Levemente	Levemente	Com Concorrença	Baixa	Sem
5	Estreita	Levemente	Levemente	Com Concorrença	Media	Baixa
6	Estreita	Levemente	Levemente	Com Concorrença	Alta	Baixa
7	Estreita	Levemente	Levemente	Extremamente	Baixa	Sem
8	Estreita	Levemente	Levemente	Extremamente	Media	Sem
9	Estreita	Levemente	Levemente	Extremamente	Alta	Baixa
10	Estreita	Levemente	Pouco	Sem Concorrença	Baixa	Baixa
11	Estreita	Levemente	Pouco	Sem Concorrença	Media	Baixa
12	Estreita	Levemente	Pouco	Sem Concorrença	Alta	Media
13	Estreita	Levemente	Pouco	Com Concorrença	Baixa	Baixa
14	Estreita	Levemente	Pouco	Com Concorrença	Media	Baixa
15	Estreita	Levemente	Pouco	Com Concorrença	Alta	Baixa
16	Estreita	Levemente	Pouco	Extremamente	Baixa	Sem
17	Estreita	Levemente	Pouco	Extremamente	Media	Baixa
18	Estreita	Levemente	Pouco	Extremamente	Alta	Baixa
19	Estreita	Levemente	Populada	Sem Concorrença	Baixa	Baixa
20	Estreita	Levemente	Populada	Sem Concorrença	Media	Baixa
21	Estreita	Levemente	Populada	Sem Concorrença	Alta	Media
22	Estreita	Levemente	Populada	Com Concorrença	Baixa	Baixa
23	Estreita	Levemente	Populada	Com Concorrença	Media	Baixa
24	Estreita	Levemente	Populada	Com Concorrença	Alta	Baixa
25	Estreita	Levemente	Populada	Extremamente	Baixa	Sem
26	Estreita	Levemente	Populada	Extremamente	Media	Baixa
27	Estreita	Levemente	Populada	Extremamente	Alta	Baixa
28	Estreita	Levemente	Muito	Sem Concorrença	Baixa	Baixa
29	Estreita	Levemente	Muito	Sem Concorrença	Media	Baixa
30	Estreita	Levemente	Muito	Sem Concorrença	Alta	Media
31	Estreita	Levemente	Muito	Com Concorrença	Baixa	Baixa
32	Estreita	Levemente	Muito	Com Concorrença	Media	Baixa
33	Estreita	Levemente	Muito	Com Concorrença	Alta	Baixa
34	Estreita	Levemente	Muito	Extremamente	Baixa	Baixa
35	Estreita	Levemente	Muito	Extremamente	Media	Baixa
36	Estreita	Levemente	Muito	Extremamente	Alta	Baixa
37	Estreita	Levemente	Extremamente	Sem Concorrença	Baixa	Baixa
38	Estreita	Levemente	Extremamente	Sem Concorrença	Media	Media
39	Estreita	Levemente	Extremamente	Sem Concorrença	Alta	Media
40	Estreita	Levemente	Extremamente	Com Concorrença	Baixa	Baixa
41	Estreita	Levemente	Extremamente	Com Concorrença	Media	Baixa
42	Estreita	Levemente	Extremamente	Com Concorrença	Alta	Media
43	Estreita	Levemente	Extremamente	Extremamente	Baixa	Baixa
44	Estreita	Levemente	Extremamente	Extremamente	Media	Baixa
45	Estreita	Levemente	Extremamente	Extremamente	Alta	Baixa
46	Estreita	Pouco	Levemente	Sem Concorrença	Baixa	Baixa
47	Estreita	Pouco	Levemente	Sem Concorrença	Media	Baixa
48	Estreita	Pouco	Levemente	Sem Concorrença	Alta	Media
49	Estreita	Pouco	Levemente	Com Concorrença	Baixa	Baixa
50	Estreita	Pouco	Levemente	Com Concorrença	Media	Baixa
51	Estreita	Pouco	Levemente	Com Concorrença	Alta	Baixa
52	Estreita	Pouco	Levemente	Extremamente	Baixa	Sem
53	Estreita	Pouco	Levemente	Extremamente	Media	Baixa
54	Estreita	Pouco	Levemente	Extremamente	Alta	Baixa
55	Estreita	Pouco	Pouco	Sem Concorrença	Baixa	Baixa
56	Estreita	Pouco	Pouco	Sem Concorrença	Media	Baixa

NRegra	Conexão	Urbanização	População	Concorrência	Renda	Potencialidade
57	Estreita	Pouco	Pouco	Sem Concorrência	Alta	Media
58	Estreita	Pouco	Pouco	Com Concorrência	Baixa	Baixa
59	Estreita	Pouco	Pouco	Com Concorrência	Media	Baixa
60	Estreita	Pouco	Pouco	Com Concorrência	Alta	Baixa
61	Estreita	Pouco	Pouco	Extremamente	Baixa	Sem
62	Estreita	Pouco	Pouco	Extremamente	Media	Baixa
63	Estreita	Pouco	Pouco	Extremamente	Alta	Baixa
64	Estreita	Pouco	Populada	Sem Concorrência	Baixa	Baixa
65	Estreita	Pouco	Populada	Sem Concorrência	Media	Baixa
66	Estreita	Pouco	Populada	Sem Concorrência	Alta	Media
67	Estreita	Pouco	Populada	Com Concorrência	Baixa	Baixa
68	Estreita	Pouco	Populada	Com Concorrência	Media	Baixa
69	Estreita	Pouco	Populada	Com Concorrência	Alta	Baixa
70	Estreita	Pouco	Populada	Extremamente	Baixa	Baixa
71	Estreita	Pouco	Populada	Extremamente	Media	Baixa
72	Estreita	Pouco	Populada	Extremamente	Alta	Baixa
73	Estreita	Pouco	Muito	Sem Concorrência	Baixa	Baixa
74	Estreita	Pouco	Muito	Sem Concorrência	Media	Media
75	Estreita	Pouco	Muito	Sem Concorrência	Alta	Media
76	Estreita	Pouco	Muito	Com Concorrência	Baixa	Baixa
77	Estreita	Pouco	Muito	Com Concorrência	Media	Baixa
78	Estreita	Pouco	Muito	Com Concorrência	Alta	Media
79	Estreita	Pouco	Muito	Extremamente	Baixa	Baixa
80	Estreita	Pouco	Muito	Extremamente	Media	Baixa
81	Estreita	Pouco	Muito	Extremamente	Alta	Baixa
82	Estreita	Pouco	Extremamente	Sem Concorrência	Baixa	Baixa
83	Estreita	Pouco	Extremamente	Sem Concorrência	Media	Media
84	Estreita	Pouco	Extremamente	Sem Concorrência	Alta	Media
85	Estreita	Pouco	Extremamente	Com Concorrência	Baixa	Baixa
86	Estreita	Pouco	Extremamente	Com Concorrência	Media	Baixa
87	Estreita	Pouco	Extremamente	Com Concorrência	Alta	Media
88	Estreita	Pouco	Extremamente	Extremamente	Baixa	Baixa
89	Estreita	Pouco	Extremamente	Extremamente	Media	Baixa
90	Estreita	Pouco	Extremamente	Extremamente	Alta	Baixa
91	Estreita	Urbanizada	Levemente	Sem Concorrência	Baixa	Baixa
92	Estreita	Urbanizada	Levemente	Sem Concorrência	Media	Baixa
93	Estreita	Urbanizada	Levemente	Sem Concorrência	Alta	Media
94	Estreita	Urbanizada	Levemente	Com Concorrência	Baixa	Baixa
95	Estreita	Urbanizada	Levemente	Com Concorrência	Media	Baixa
96	Estreita	Urbanizada	Levemente	Com Concorrência	Alta	Baixa
97	Estreita	Urbanizada	Levemente	Extremamente	Baixa	Sem
98	Estreita	Urbanizada	Levemente	Extremamente	Media	Baixa
99	Estreita	Urbanizada	Levemente	Extremamente	Alta	Baixa
100	Estreita	Urbanizada	Pouco	Sem Concorrência	Baixa	Baixa
101	Estreita	Urbanizada	Pouco	Sem Concorrência	Media	Baixa
102	Estreita	Urbanizada	Pouco	Sem Concorrência	Alta	Media
103	Estreita	Urbanizada	Pouco	Com Concorrência	Baixa	Baixa
104	Estreita	Urbanizada	Pouco	Com Concorrência	Media	Baixa
105	Estreita	Urbanizada	Pouco	Com Concorrência	Alta	Baixa
106	Estreita	Urbanizada	Pouco	Extremamente	Baixa	Baixa
107	Estreita	Urbanizada	Pouco	Extremamente	Media	Baixa
108	Estreita	Urbanizada	Pouco	Extremamente	Alta	Baixa
109	Estreita	Urbanizada	Populada	Sem Concorrência	Baixa	Baixa
110	Estreita	Urbanizada	Populada	Sem Concorrência	Media	Media
111	Estreita	Urbanizada	Populada	Sem Concorrência	Alta	Media
112	Estreita	Urbanizada	Populada	Com Concorrência	Baixa	Baixa
113	Estreita	Urbanizada	Populada	Com Concorrência	Media	Baixa
114	Estreita	Urbanizada	Populada	Com Concorrência	Alta	Media
115	Estreita	Urbanizada	Populada	Extremamente	Baixa	Baixa
116	Estreita	Urbanizada	Populada	Extremamente	Media	Baixa
117	Estreita	Urbanizada	Populada	Extremamente	Alta	Baixa
118	Estreita	Urbanizada	Muito	Sem Concorrência	Baixa	Baixa
119	Estreita	Urbanizada	Muito	Sem Concorrência	Media	Media
120	Estreita	Urbanizada	Muito	Sem Concorrência	Alta	Media

NRegra	Conexão	Urbanização	População	Concorrência	Renda	Potencialidade
121	Estreita	Urbanizada	Muito	Com Concorrência	Baixa	Baixa
122	Estreita	Urbanizada	Muito	Com Concorrência	Media	Baixa
123	Estreita	Urbanizada	Muito	Com Concorrência	Alta	Media
124	Estreita	Urbanizada	Muito	Extremamente	Baixa	Baixa
125	Estreita	Urbanizada	Muito	Extremamente	Media	Baixa
126	Estreita	Urbanizada	Muito	Extremamente	Alta	Baixa
127	Estreita	Urbanizada	Extremamente	Sem Concorrência	Baixa	Media
128	Estreita	Urbanizada	Extremamente	Sem Concorrência	Media	Media
129	Estreita	Urbanizada	Extremamente	Sem Concorrência	Alta	Media
130	Estreita	Urbanizada	Extremamente	Com Concorrência	Baixa	Baixa
131	Estreita	Urbanizada	Extremamente	Com Concorrência	Media	Baixa
132	Estreita	Urbanizada	Extremamente	Com Concorrência	Alta	Media
133	Estreita	Urbanizada	Extremamente	Extremamente	Baixa	Baixa
134	Estreita	Urbanizada	Extremamente	Extremamente	Media	Baixa
135	Estreita	Urbanizada	Extremamente	Extremamente	Alta	Media
136	Estreita	Muito	Levemente	Sem Concorrência	Baixa	Baixa
137	Estreita	Muito	Levemente	Sem Concorrência	Media	Baixa
138	Estreita	Muito	Levemente	Sem Concorrência	Alta	Media
139	Estreita	Muito	Levemente	Com Concorrência	Baixa	Baixa
140	Estreita	Muito	Levemente	Com Concorrência	Media	Baixa
141	Estreita	Muito	Levemente	Com Concorrência	Alta	Baixa
142	Estreita	Muito	Levemente	Extremamente	Baixa	Baixa
143	Estreita	Muito	Levemente	Extremamente	Media	Baixa
144	Estreita	Muito	Levemente	Extremamente	Alta	Baixa
145	Estreita	Muito	Pouco	Sem Concorrência	Baixa	Baixa
146	Estreita	Muito	Pouco	Sem Concorrência	Media	Media
147	Estreita	Muito	Pouco	Sem Concorrência	Alta	Media
148	Estreita	Muito	Pouco	Com Concorrência	Baixa	Baixa
149	Estreita	Muito	Pouco	Com Concorrência	Media	Baixa
150	Estreita	Muito	Pouco	Com Concorrência	Alta	Media
151	Estreita	Muito	Pouco	Extremamente	Baixa	Baixa
152	Estreita	Muito	Pouco	Extremamente	Media	Baixa
153	Estreita	Muito	Pouco	Extremamente	Alta	Baixa
154	Estreita	Muito	Populada	Sem Concorrência	Baixa	Baixa
155	Estreita	Muito	Populada	Sem Concorrência	Media	Media
156	Estreita	Muito	Populada	Sem Concorrência	Alta	Media
157	Estreita	Muito	Populada	Com Concorrência	Baixa	Baixa
158	Estreita	Muito	Populada	Com Concorrência	Media	Baixa
159	Estreita	Muito	Populada	Com Concorrência	Alta	Media
160	Estreita	Muito	Populada	Extremamente	Baixa	Baixa
161	Estreita	Muito	Populada	Extremamente	Media	Baixa
162	Estreita	Muito	Populada	Extremamente	Alta	Baixa
163	Estreita	Muito	Muito	Sem Concorrência	Baixa	Media
164	Estreita	Muito	Muito	Sem Concorrência	Media	Media
165	Estreita	Muito	Muito	Sem Concorrência	Alta	Media
166	Estreita	Muito	Muito	Com Concorrência	Baixa	Baixa
167	Estreita	Muito	Muito	Com Concorrência	Media	Baixa
168	Estreita	Muito	Muito	Com Concorrência	Alta	Media
169	Estreita	Muito	Muito	Extremamente	Baixa	Baixa
170	Estreita	Muito	Muito	Extremamente	Media	Baixa
171	Estreita	Muito	Muito	Extremamente	Alta	Media
172	Estreita	Muito	Extremamente	Sem Concorrência	Baixa	Media
173	Estreita	Muito	Extremamente	Sem Concorrência	Media	Media
174	Estreita	Muito	Extremamente	Sem Concorrência	Alta	Boa
175	Estreita	Muito	Extremamente	Com Concorrência	Baixa	Baixa
176	Estreita	Muito	Extremamente	Com Concorrência	Media	Media
177	Estreita	Muito	Extremamente	Com Concorrência	Alta	Media
178	Estreita	Muito	Extremamente	Extremamente	Baixa	Baixa
179	Estreita	Muito	Extremamente	Extremamente	Media	Baixa
180	Estreita	Muito	Extremamente	Extremamente	Alta	Media
181	Estreita	Totalmente	Levemente	Sem Concorrência	Baixa	Baixa
182	Estreita	Totalmente	Levemente	Sem Concorrência	Media	Media
183	Estreita	Totalmente	Levemente	Sem Concorrência	Alta	Media
184	Estreita	Totalmente	Levemente	Com Concorrência	Baixa	Baixa

NRegra	Conexão	Urbanização	População	Concorrência	Renda	Potencialidade
185	Estreita	Totalmente	Levemente	Com Concorrência	Media	Baixa
186	Estreita	Totalmente	Levemente	Com Concorrência	Alta	Media
187	Estreita	Totalmente	Levemente	Extremamente	Baixa	Baixa
188	Estreita	Totalmente	Levemente	Extremamente	Media	Baixa
189	Estreita	Totalmente	Levemente	Extremamente	Alta	Baixa
190	Estreita	Totalmente	Pouco	Sem Concorrência	Baixa	Baixa
191	Estreita	Totalmente	Pouco	Sem Concorrência	Media	Media
192	Estreita	Totalmente	Pouco	Sem Concorrência	Alta	Media
193	Estreita	Totalmente	Pouco	Com Concorrência	Baixa	Baixa
194	Estreita	Totalmente	Pouco	Com Concorrência	Media	Baixa
195	Estreita	Totalmente	Pouco	Com Concorrência	Alta	Media
196	Estreita	Totalmente	Pouco	Extremamente	Baixa	Baixa
197	Estreita	Totalmente	Pouco	Extremamente	Media	Baixa
198	Estreita	Totalmente	Pouco	Extremamente	Alta	Baixa
199	Estreita	Totalmente	Populada	Sem Concorrência	Baixa	Media
200	Estreita	Totalmente	Populada	Sem Concorrência	Media	Media
201	Estreita	Totalmente	Populada	Sem Concorrência	Alta	Media
202	Estreita	Totalmente	Populada	Com Concorrência	Baixa	Baixa
203	Estreita	Totalmente	Populada	Com Concorrência	Media	Baixa
204	Estreita	Totalmente	Populada	Com Concorrência	Alta	Media
205	Estreita	Totalmente	Populada	Extremamente	Baixa	Baixa
206	Estreita	Totalmente	Populada	Extremamente	Media	Baixa
207	Estreita	Totalmente	Populada	Extremamente	Alta	Media
208	Estreita	Totalmente	Muito	Sem Concorrência	Baixa	Media
209	Estreita	Totalmente	Muito	Sem Concorrência	Media	Media
210	Estreita	Totalmente	Muito	Sem Concorrência	Alta	Boa
211	Estreita	Totalmente	Muito	Com Concorrência	Baixa	Baixa
212	Estreita	Totalmente	Muito	Com Concorrência	Media	Media
213	Estreita	Totalmente	Muito	Com Concorrência	Alta	Media
214	Estreita	Totalmente	Muito	Extremamente	Baixa	Baixa
215	Estreita	Totalmente	Muito	Extremamente	Media	Baixa
216	Estreita	Totalmente	Muito	Extremamente	Alta	Media
217	Estreita	Totalmente	Extremamente	Sem Concorrência	Baixa	Media
218	Estreita	Totalmente	Extremamente	Sem Concorrência	Media	Media
219	Estreita	Totalmente	Extremamente	Sem Concorrência	Alta	Boa
220	Estreita	Totalmente	Extremamente	Com Concorrência	Baixa	Baixa
221	Estreita	Totalmente	Extremamente	Com Concorrência	Media	Media
222	Estreita	Totalmente	Extremamente	Com Concorrência	Alta	Media
223	Estreita	Totalmente	Extremamente	Extremamente	Baixa	Baixa
224	Estreita	Totalmente	Extremamente	Extremamente	Media	Baixa
225	Estreita	Totalmente	Extremamente	Extremamente	Alta	Media
226	Larga	Levemente	Levemente	Sem Concorrência	Baixa	Baixa
227	Larga	Levemente	Levemente	Sem Concorrência	Media	Baixa
228	Larga	Levemente	Levemente	Sem Concorrência	Alta	Media
229	Larga	Levemente	Levemente	Com Concorrência	Baixa	Baixa
230	Larga	Levemente	Levemente	Com Concorrência	Media	Baixa
231	Larga	Levemente	Levemente	Com Concorrência	Alta	Baixa
232	Larga	Levemente	Levemente	Extremamente	Baixa	Baixa
233	Larga	Levemente	Levemente	Extremamente	Media	Baixa
234	Larga	Levemente	Levemente	Extremamente	Alta	Baixa
235	Larga	Levemente	Pouco	Sem Concorrência	Baixa	Baixa
236	Larga	Levemente	Pouco	Sem Concorrência	Media	Media
237	Larga	Levemente	Pouco	Sem Concorrência	Alta	Media
238	Larga	Levemente	Pouco	Com Concorrência	Baixa	Baixa
239	Larga	Levemente	Pouco	Com Concorrência	Media	Baixa
240	Larga	Levemente	Pouco	Com Concorrência	Alta	Media
241	Larga	Levemente	Pouco	Extremamente	Baixa	Baixa
242	Larga	Levemente	Pouco	Extremamente	Media	Baixa
243	Larga	Levemente	Pouco	Extremamente	Alta	Baixa
244	Larga	Levemente	Populada	Sem Concorrência	Baixa	Baixa
245	Larga	Levemente	Populada	Sem Concorrência	Media	Media
246	Larga	Levemente	Populada	Sem Concorrência	Alta	Media
247	Larga	Levemente	Populada	Com Concorrência	Baixa	Baixa
248	Larga	Levemente	Populada	Com Concorrência	Media	Baixa

NRegra	Conexão	Urbanização	População	Concorrência	Renda	Potencialidade
249	Larga	Levemente	Populada	Com Concorrência	Alta	Media
250	Larga	Levemente	Populada	Extremamente	Baixa	Baixa
251	Larga	Levemente	Populada	Extremamente	Media	Baixa
252	Larga	Levemente	Populada	Extremamente	Alta	Baixa
253	Larga	Levemente	Muito	Sem Concorrência	Baixa	Media
254	Larga	Levemente	Muito	Sem Concorrência	Media	Media
255	Larga	Levemente	Muito	Sem Concorrência	Alta	Media
256	Larga	Levemente	Muito	Com Concorrência	Baixa	Baixa
257	Larga	Levemente	Muito	Com Concorrência	Media	Baixa
258	Larga	Levemente	Muito	Com Concorrência	Alta	Media
259	Larga	Levemente	Muito	Extremamente	Baixa	Baixa
260	Larga	Levemente	Muito	Extremamente	Media	Baixa
261	Larga	Levemente	Muito	Extremamente	Alta	Media
262	Larga	Levemente	Extremamente	Sem Concorrência	Baixa	Media
263	Larga	Levemente	Extremamente	Sem Concorrência	Media	Media
264	Larga	Levemente	Extremamente	Sem Concorrência	Alta	Boa
265	Larga	Levemente	Extremamente	Com Concorrência	Baixa	Baixa
266	Larga	Levemente	Extremamente	Com Concorrência	Media	Media
267	Larga	Levemente	Extremamente	Com Concorrência	Alta	Media
268	Larga	Levemente	Extremamente	Extremamente	Baixa	Baixa
269	Larga	Levemente	Extremamente	Extremamente	Media	Baixa
270	Larga	Levemente	Extremamente	Extremamente	Alta	Media
271	Larga	Pouco	Levemente	Sem Concorrência	Baixa	Baixa
272	Larga	Pouco	Levemente	Sem Concorrência	Media	Media
273	Larga	Pouco	Levemente	Sem Concorrência	Alta	Media
274	Larga	Pouco	Levemente	Com Concorrência	Baixa	Baixa
275	Larga	Pouco	Levemente	Com Concorrência	Media	Baixa
276	Larga	Pouco	Levemente	Com Concorrência	Alta	Media
277	Larga	Pouco	Levemente	Extremamente	Baixa	Baixa
278	Larga	Pouco	Levemente	Extremamente	Media	Baixa
279	Larga	Pouco	Levemente	Extremamente	Alta	Baixa
280	Larga	Pouco	Pouco	Sem Concorrência	Baixa	Baixa
281	Larga	Pouco	Pouco	Sem Concorrência	Media	Media
282	Larga	Pouco	Pouco	Sem Concorrência	Alta	Media
283	Larga	Pouco	Pouco	Com Concorrência	Baixa	Baixa
284	Larga	Pouco	Pouco	Com Concorrência	Media	Baixa
285	Larga	Pouco	Pouco	Com Concorrência	Alta	Media
286	Larga	Pouco	Pouco	Extremamente	Baixa	Baixa
287	Larga	Pouco	Pouco	Extremamente	Media	Baixa
288	Larga	Pouco	Pouco	Extremamente	Alta	Baixa
289	Larga	Pouco	Populada	Sem Concorrência	Baixa	Media
290	Larga	Pouco	Populada	Sem Concorrência	Media	Media
291	Larga	Pouco	Populada	Sem Concorrência	Alta	Media
292	Larga	Pouco	Populada	Com Concorrência	Baixa	Baixa
293	Larga	Pouco	Populada	Com Concorrência	Media	Baixa
294	Larga	Pouco	Populada	Com Concorrência	Alta	Media
295	Larga	Pouco	Populada	Extremamente	Baixa	Baixa
296	Larga	Pouco	Populada	Extremamente	Media	Baixa
297	Larga	Pouco	Populada	Extremamente	Alta	Media
298	Larga	Pouco	Muito	Sem Concorrência	Baixa	Media
299	Larga	Pouco	Muito	Sem Concorrência	Media	Media
300	Larga	Pouco	Muito	Sem Concorrência	Alta	Boa
301	Larga	Pouco	Muito	Com Concorrência	Baixa	Baixa
302	Larga	Pouco	Muito	Com Concorrência	Media	Media
303	Larga	Pouco	Muito	Com Concorrência	Alta	Media
304	Larga	Pouco	Muito	Extremamente	Baixa	Baixa
305	Larga	Pouco	Muito	Extremamente	Media	Baixa
306	Larga	Pouco	Muito	Extremamente	Alta	Media
307	Larga	Pouco	Extremamente	Sem Concorrência	Baixa	Media
308	Larga	Pouco	Extremamente	Sem Concorrência	Media	Media
309	Larga	Pouco	Extremamente	Sem Concorrência	Alta	Boa
310	Larga	Pouco	Extremamente	Com Concorrência	Baixa	Baixa
311	Larga	Pouco	Extremamente	Com Concorrência	Media	Media
312	Larga	Pouco	Extremamente	Com Concorrência	Alta	Media

NRegra	Conexão	Urbanização	População	Concorrência	Renda	Potencialidade
313	Larga	Pouco	Extremamente	Extremamente	Baixa	Baixa
314	Larga	Pouco	Extremamente	Extremamente	Media	Baixa
315	Larga	Pouco	Extremamente	Extremamente	Alta	Media
316	Larga	Urbanizada	Levemente	Sem Concorrença	Baixa	Baixa
317	Larga	Urbanizada	Levemente	Sem Concorrença	Media	Media
318	Larga	Urbanizada	Levemente	Sem Concorrença	Alta	Media
319	Larga	Urbanizada	Levemente	Com Concorrença	Baixa	Baixa
320	Larga	Urbanizada	Levemente	Com Concorrença	Media	Baixa
321	Larga	Urbanizada	Levemente	Com Concorrença	Alta	Media
322	Larga	Urbanizada	Levemente	Extremamente	Baixa	Baixa
323	Larga	Urbanizada	Levemente	Extremamente	Media	Baixa
324	Larga	Urbanizada	Levemente	Extremamente	Alta	Baixa
325	Larga	Urbanizada	Pouco	Sem Concorrença	Baixa	Media
326	Larga	Urbanizada	Pouco	Sem Concorrença	Media	Media
327	Larga	Urbanizada	Pouco	Sem Concorrença	Alta	Media
328	Larga	Urbanizada	Pouco	Com Concorrença	Baixa	Baixa
329	Larga	Urbanizada	Pouco	Com Concorrença	Media	Baixa
330	Larga	Urbanizada	Pouco	Com Concorrença	Alta	Media
331	Larga	Urbanizada	Pouco	Extremamente	Baixa	Baixa
332	Larga	Urbanizada	Pouco	Extremamente	Media	Baixa
333	Larga	Urbanizada	Pouco	Extremamente	Alta	Media
334	Larga	Urbanizada	Populada	Sem Concorrença	Baixa	Media
335	Larga	Urbanizada	Populada	Sem Concorrença	Media	Media
336	Larga	Urbanizada	Populada	Sem Concorrença	Alta	Boa
337	Larga	Urbanizada	Populada	Com Concorrença	Baixa	Baixa
338	Larga	Urbanizada	Populada	Com Concorrença	Media	Media
339	Larga	Urbanizada	Populada	Com Concorrença	Alta	Media
340	Larga	Urbanizada	Populada	Extremamente	Baixa	Baixa
341	Larga	Urbanizada	Populada	Extremamente	Media	Baixa
342	Larga	Urbanizada	Populada	Extremamente	Alta	Media
343	Larga	Urbanizada	Muito	Sem Concorrença	Baixa	Media
344	Larga	Urbanizada	Muito	Sem Concorrença	Media	Media
345	Larga	Urbanizada	Muito	Sem Concorrença	Alta	Boa
346	Larga	Urbanizada	Muito	Com Concorrença	Baixa	Baixa
347	Larga	Urbanizada	Muito	Com Concorrença	Media	Media
348	Larga	Urbanizada	Muito	Com Concorrença	Alta	Media
349	Larga	Urbanizada	Muito	Extremamente	Baixa	Baixa
350	Larga	Urbanizada	Muito	Extremamente	Media	Baixa
351	Larga	Urbanizada	Muito	Extremamente	Alta	Media
352	Larga	Urbanizada	Extremamente	Sem Concorrença	Baixa	Media
353	Larga	Urbanizada	Extremamente	Sem Concorrença	Media	Media
354	Larga	Urbanizada	Extremamente	Sem Concorrença	Alta	Boa
355	Larga	Urbanizada	Extremamente	Com Concorrença	Baixa	Media
356	Larga	Urbanizada	Extremamente	Com Concorrença	Media	Media
357	Larga	Urbanizada	Extremamente	Com Concorrença	Alta	Media
358	Larga	Urbanizada	Extremamente	Extremamente	Baixa	Baixa
359	Larga	Urbanizada	Extremamente	Extremamente	Media	Media
360	Larga	Urbanizada	Extremamente	Extremamente	Alta	Media
361	Larga	Muito	Levemente	Sem Concorrença	Baixa	Media
362	Larga	Muito	Levemente	Sem Concorrença	Media	Media
363	Larga	Muito	Levemente	Sem Concorrença	Alta	Media
364	Larga	Muito	Levemente	Com Concorrença	Baixa	Baixa
365	Larga	Muito	Levemente	Com Concorrença	Media	Baixa
366	Larga	Muito	Levemente	Com Concorrença	Alta	Media
367	Larga	Muito	Levemente	Extremamente	Baixa	Baixa
368	Larga	Muito	Levemente	Extremamente	Media	Baixa
369	Larga	Muito	Levemente	Extremamente	Alta	Media
370	Larga	Muito	Pouco	Sem Concorrença	Baixa	Media
371	Larga	Muito	Pouco	Sem Concorrença	Media	Media
372	Larga	Muito	Pouco	Sem Concorrença	Alta	Boa
373	Larga	Muito	Pouco	Com Concorrença	Baixa	Baixa
374	Larga	Muito	Pouco	Com Concorrença	Media	Media
375	Larga	Muito	Pouco	Com Concorrença	Alta	Media
376	Larga	Muito	Pouco	Extremamente	Baixa	Baixa

NRegra	Conexão	Urbanização	População	Concorrência	Renda	Potencialidade
377	Larga	Muito	Pouco	Extremamente	Media	Baixa
378	Larga	Muito	Pouco	Extremamente	Alta	Media
379	Larga	Muito	Populada	Sem Concorrença	Baixa	Media
380	Larga	Muito	Populada	Sem Concorrença	Media	Media
381	Larga	Muito	Populada	Sem Concorrença	Alta	Boa
382	Larga	Muito	Populada	Com Concorrença	Baixa	Baixa
383	Larga	Muito	Populada	Com Concorrença	Media	Media
384	Larga	Muito	Populada	Com Concorrença	Alta	Media
385	Larga	Muito	Populada	Extremamente	Baixa	Baixa
386	Larga	Muito	Populada	Extremamente	Media	Baixa
387	Larga	Muito	Populada	Extremamente	Alta	Media
388	Larga	Muito	Muito	Sem Concorrença	Baixa	Media
389	Larga	Muito	Muito	Sem Concorrença	Media	Media
390	Larga	Muito	Muito	Sem Concorrença	Alta	Boa
391	Larga	Muito	Muito	Com Concorrença	Baixa	Media
392	Larga	Muito	Muito	Com Concorrença	Media	Media
393	Larga	Muito	Muito	Com Concorrença	Alta	Media
394	Larga	Muito	Muito	Extremamente	Baixa	Baixa
395	Larga	Muito	Muito	Extremamente	Media	Media
396	Larga	Muito	Muito	Extremamente	Alta	Media
397	Larga	Muito	Extremamente	Sem Concorrença	Baixa	Media
398	Larga	Muito	Extremamente	Sem Concorrença	Media	Boa
399	Larga	Muito	Extremamente	Sem Concorrença	Alta	Otima
400	Larga	Muito	Extremamente	Com Concorrença	Baixa	Media
401	Larga	Muito	Extremamente	Com Concorrença	Media	Media
402	Larga	Muito	Extremamente	Com Concorrença	Alta	Boa
403	Larga	Muito	Extremamente	Extremamente	Baixa	Baixa
404	Larga	Muito	Extremamente	Extremamente	Media	Media
405	Larga	Muito	Extremamente	Extremamente	Alta	Media
406	Larga	Totalmente	Levemente	Sem Concorrença	Baixa	Media
407	Larga	Totalmente	Levemente	Sem Concorrença	Media	Media
408	Larga	Totalmente	Levemente	Sem Concorrença	Alta	Boa
409	Larga	Totalmente	Levemente	Com Concorrença	Baixa	Baixa
410	Larga	Totalmente	Levemente	Com Concorrença	Media	Media
411	Larga	Totalmente	Levemente	Com Concorrença	Alta	Media
412	Larga	Totalmente	Levemente	Extremamente	Baixa	Baixa
413	Larga	Totalmente	Levemente	Extremamente	Media	Baixa
414	Larga	Totalmente	Levemente	Extremamente	Alta	Media
415	Larga	Totalmente	Pouco	Sem Concorrença	Baixa	Media
416	Larga	Totalmente	Pouco	Sem Concorrença	Media	Media
417	Larga	Totalmente	Pouco	Sem Concorrença	Alta	Boa
418	Larga	Totalmente	Pouco	Com Concorrença	Baixa	Baixa
419	Larga	Totalmente	Pouco	Com Concorrença	Media	Media
420	Larga	Totalmente	Pouco	Com Concorrença	Alta	Media
421	Larga	Totalmente	Pouco	Extremamente	Baixa	Baixa
422	Larga	Totalmente	Pouco	Extremamente	Media	Baixa
423	Larga	Totalmente	Pouco	Extremamente	Alta	Media
424	Larga	Totalmente	Populada	Sem Concorrença	Baixa	Media
425	Larga	Totalmente	Populada	Sem Concorrença	Media	Media
426	Larga	Totalmente	Populada	Sem Concorrença	Alta	Boa
427	Larga	Totalmente	Populada	Com Concorrença	Baixa	Media
428	Larga	Totalmente	Populada	Com Concorrença	Media	Media
429	Larga	Totalmente	Populada	Com Concorrença	Alta	Media
430	Larga	Totalmente	Populada	Extremamente	Baixa	Baixa
431	Larga	Totalmente	Populada	Extremamente	Media	Media
432	Larga	Totalmente	Populada	Extremamente	Alta	Media
433	Larga	Totalmente	Muito	Sem Concorrença	Baixa	Media
434	Larga	Totalmente	Muito	Sem Concorrença	Media	Boa
435	Larga	Totalmente	Muito	Sem Concorrença	Alta	Otima
436	Larga	Totalmente	Muito	Com Concorrença	Baixa	Media
437	Larga	Totalmente	Muito	Com Concorrença	Media	Media
438	Larga	Totalmente	Muito	Com Concorrença	Alta	Boa
439	Larga	Totalmente	Muito	Extremamente	Baixa	Baixa
440	Larga	Totalmente	Muito	Extremamente	Media	Media

NRegra	Conexão	Urbanização	População	Concorrência	Renda	Potencialidade
441	Larga	Totalmente	Muito	Extremamente	Alta	Media
442	Larga	Totalmente	Extremamente	Sem Concorrença	Baixa	Media
443	Larga	Totalmente	Extremamente	Sem Concorrença	Media	Boa
444	Larga	Totalmente	Extremamente	Sem Concorrença	Alta	Otima
445	Larga	Totalmente	Extremamente	Com Concorrença	Baixa	Media
446	Larga	Totalmente	Extremamente	Com Concorrença	Media	Media
447	Larga	Totalmente	Extremamente	Com Concorrença	Alta	Boa
448	Larga	Totalmente	Extremamente	Extremamente	Baixa	Media
449	Larga	Totalmente	Extremamente	Extremamente	Media	Media
450	Larga	Totalmente	Extremamente	Extremamente	Alta	Media

Apêndice B1 – Codificação: Classe ClsAcesso

```
VERSION 1.0 CLASS
BEGIN
    MultiUse = -1 'True
    Persistable = 0 'NotPersistable
    DataBindingBehavior = 0 'vbNone
    DataSourceBehavior = 0 'vbNone
    MTSTransactionMode = 0 'NotAnMTSObject
END
Attribute VB_Name = "clsAcesso"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = False
Attribute VB_Exposed = True
Attribute VB_Ext_KEY = "SavedWithClassBuilder6" , "Yes"
Attribute VB_Ext_KEY = "Top_Level" , "Yes"
Const ERRO_INATIVO = -2
Const ERRO_INVALIDO = -1
Const ERRO_SEMACCESSO = 0
Public Function ValidaLogin(ByVal Login As String, ByVal Senha As String, ByVal
StringConexao As String) As Long
    Dim cn As ADODB.Connection
    Dim rs As ADODB.Recordset
    Set cn = New ADODB.Connection
    Set rs = New ADODB.Recordset
    cn.Open StringConexao
    Set rs.ActiveConnection = cn
    rs.CursorLocation = adUseClient
    rs.Open "Select * from usuarios where Login = '" & Login & "' and Senha = '" &
Senha & "'"
    If rs.RecordCount > 0 Then
        rs.MoveFirst
        If rs!indativo = 0 Then
            ValidaLogin = ERRO_INATIVO
        Else
            ValidaLogin = rs!Acesso
        End If
    Else
        ValidaLogin = ERRO_INVALIDO
    End If
End Function
```

Apêndice B2 – Codificação: Classe ClsFuzzy

```
VERSION 1.0 CLASS
BEGIN
    MultiUse = -1 'True
    Persistable = 0 'NotPersistable
    DataBindingBehavior = 0 'vbNone
    DataSourceBehavior = 0 'vbNone
    MTSTransactionMode = 1 'NoTransaction
END
Attribute VB_Name = "clsFuzzy"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = False
Attribute VB_Exposed = True
Attribute VB_Ext_KEY = "SavedWithClassBuilder6" , "Yes"
Attribute VB_Ext_KEY = "Top_Level" , "Yes"
Type AntecedentesStruc
    Descricao As String
    IntervaloInic As Long
```

```

        IntervaloFinal As Long
End Type
Type FuncaoAuxiliar
    Descricao As String
    CoefAng As Double
    CoefLin As Double
    IntervInic As Double
    IntervFim As Double
    Continuo As Boolean
End Type
Type Regra
    'Antecedente1 As String
    Antecedente2 As String
    Antecedente3 As String
    Antecedente4 As String
    Antecedente5 As String
    Antecedente6 As String
    Consequente As String
End Type
Public Disponibilidade As Double
Public Ociosidade As Double
Public Conexao As Double
Public Urbanizacao As Double
Public Populacao As Double
Public Renda As Double
Public Alpha As Double
Public Regiao As String
Dim Antecedentes(5) As AntecedentesStruc
Dim Consequentes(0) As AntecedentesStruc
Dim Antecedente2() As FuncaoAuxiliar
Dim Antecedente3() As FuncaoAuxiliar
Dim Antecedente4() As FuncaoAuxiliar
Dim Antecedente5() As FuncaoAuxiliar
Dim Antecedente6() As FuncaoAuxiliar
Dim Consequente() As FuncaoAuxiliar
Dim fAux1() As FuncaoAuxiliar
Dim fAux2() As FuncaoAuxiliar
Dim fAux3() As FuncaoAuxiliar
Dim fAux4() As FuncaoAuxiliar
Dim fAux5() As FuncaoAuxiliar
Dim fAux6() As FuncaoAuxiliar
Dim FuncaoAlpha() As FuncaoAuxiliar
Dim Entrada2() As FuncaoAuxiliar
Dim Entrada3() As FuncaoAuxiliar
Dim Entrada4() As FuncaoAuxiliar
Dim Entrada5() As FuncaoAuxiliar
Dim Entrada6() As FuncaoAuxiliar
Dim Resposta() As FuncaoAuxiliar
Dim Regras() As Regra
Dim ConsequentesResp() As String
Dim PercentuaisResp() As Double
Dim RegiaoBase As String
Const ConsequenteStr = "Potencialidade"
Public NRegras As Integer
Public Resp As Integer 'Se evento realizado com sucesso.
Dim Ent2 As Double
Dim Ent3 As Double
Dim Ent4 As Double
Dim Ent5 As Double
Dim Ent6 As Double
Dim Usuario As String

Public cn As ADODB.Connection
Public rs As ADODB.Recordset
Public rs2 As ADODB.Recordset

```

```

Public Function Inicio(IdMunicipio As String, StringConexao As String, User As
String) As String
    Set cn = New ADODB.Connection
    Set rs = New ADODB.Recordset
    Set rs2 = New ADODB.Recordset
    cn.Open StringConexao
    Set rs.ActiveConnection = cn
    Usuario = User
    Regiao = GeraRegiao(IdMunicipio)
    Inicio = FazerInferencia()
End Function

Public Function MontaRegra(RegraNumero As Integer) As Integer
    Dim Contador As Integer
    rs.Close
    Set rs.ActiveConnection = cn
    rs.CursorLocation = adUseServer
    rs.Open "Select * from Funcoes where RTRIM(Funcoes.Antecedente) = '" +
        Antecedentes(1).Descricao + "'" AND RTRIM(Funcoes.TermoLing) = '" +
        Regras(RegraNumero).Antecedente2 + "'"
    If rs.RecordCount <> 0 Then
        NFuncoes = rs.RecordCount 'Numero de Funcoes
        rs.MoveFirst
        Contador = 0
        While Not rs.EOF
            ReDim Preserve Antecedente2(Contador)
            Antecedente2(Contador).Descricao = Trim(rs!antecedente) + " - " +
                Trim(rs!Termoling)
            Antecedente2(Contador).CoefAng = rs!CoefAng
            Antecedente2(Contador).CoefLin = rs!CoefLin
            Antecedente2(Contador).IntervInic = rs!IntervInic
            Antecedente2(Contador).IntervFim = rs!IntervFim
            Contador = Contador + 1
            rs.MoveNext
        Wend
    End If

    'Antecedente 3
    rs.Close
    Set rs.ActiveConnection = cn
    rs.CursorLocation = adUseServer
    rs.Open "Select * from Funcoes where RTRIM(Funcoes.Antecedente) = '" +
        Antecedentes(2).Descricao + "'" AND RTRIM(Funcoes.TermoLing) = '" +
        Regras(RegraNumero).Antecedente3 + "'"
    If rs.RecordCount <> 0 Then
        NFuncoes = rs.RecordCount 'Numero de Funcoes
        rs.MoveFirst
        Contador = 0
        While Not rs.EOF
            ReDim Preserve Antecedente3(Contador)
            Antecedente3(Contador).Descricao = Trim(rs!antecedente) + " - " +
                Trim(rs!Termoling)
            Antecedente3(Contador).CoefAng = rs!CoefAng
            Antecedente3(Contador).CoefLin = rs!CoefLin
            Antecedente3(Contador).IntervInic = rs!IntervInic
            Antecedente3(Contador).IntervFim = rs!IntervFim
            Contador = Contador + 1
            rs.MoveNext
        Wend
    End If

    'Antecedente 4
    rs.Close
    Set rs.ActiveConnection = cn
    rs.CursorLocation = adUseServer
    rs.Open "Select * from Funcoes where RTRIM(Funcoes.Antecedente) = '" +

```

```

Antecedentes(3).Descricao + "' AND RTRIM(Funcoes.TermoLing) = '" +
Regras(RegraNumero).Antecedente4 + ""
If rs.RecordCount <> 0 Then
    NFuncoes = rs.RecordCount 'Numero de Funcoes
    rs.MoveFirst
    Contador = 0
    While Not rs.EOF
        ReDim Preserve Antecedente4(Contador)
        Antecedente4(Contador).Descricao = Trim(rs!antecedente) + " - " +
            Trim(rs!Termoling)
        Antecedente4(Contador).CoefAng = rs!CoefAng
        Antecedente4(Contador).CoefLin = rs!CoefLin
        Antecedente4(Contador).IntervInic = rs!IntervInic
        Antecedente4(Contador).IntervFim = rs!IntervFim
        Contador = Contador + 1
        rs.MoveNext
    Wend
End If

'Antecedente 5
rs.Close
Set rs.ActiveConnection = cn
rs.CursorLocation = adUseServer
rs.Open "Select * from Funcoes where RTRIM(Funcoes.Antecedente) = '" +
    Antecedentes(4).Descricao + "' AND RTRIM(Funcoes.TermoLing) = '" +
    Regras(RegraNumero).Antecedente5 + ""
If rs.RecordCount <> 0 Then
    NFuncoes = rs.RecordCount 'Numero de Funcoes
    rs.MoveFirst
    Contador = 0
    While Not rs.EOF
        ReDim Preserve Antecedente5(Contador)
        Antecedente5(Contador).Descricao = Trim(rs!antecedente) + " - " +
            Trim(rs!Termoling)
        Antecedente5(Contador).CoefAng = rs!CoefAng
        Antecedente5(Contador).CoefLin = rs!CoefLin
        Antecedente5(Contador).IntervInic = rs!IntervInic
        Antecedente5(Contador).IntervFim = rs!IntervFim
        Contador = Contador + 1
        rs.MoveNext
    Wend
End If

'Antecedente 6
rs.Close
Set rs.ActiveConnection = cn
rs.CursorLocation = adUseServer
rs.Open "Select * from Funcoes where RTRIM(Funcoes.Antecedente) = '" +
    Antecedentes(5).Descricao + "' AND RTRIM(Funcoes.TermoLing) = '" +
    Regras(RegraNumero).Antecedente6 + ""
If rs.RecordCount <> 0 Then
    NFuncoes = rs.RecordCount 'Numero de Funcoes
    rs.MoveFirst
    Contador = 0
    While Not rs.EOF
        ReDim Preserve Antecedente6(Contador)
        Antecedente6(Contador).Descricao = Trim(rs!antecedente) + " - " +
            Trim(rs!Termoling)
        Antecedente6(Contador).CoefAng = rs!CoefAng
        Antecedente6(Contador).CoefLin = rs!CoefLin
        Antecedente6(Contador).IntervInic = rs!IntervInic
        Antecedente6(Contador).IntervFim = rs!IntervFim
        Contador = Contador + 1
        rs.MoveNext
    Wend
End If

```

```

        'Consequente
rs.Close
Set rs.ActiveConnection = cn
rs.CursorLocation = adUseServer
rs.Open "Select * from Funcoes where RTRIM(Funcoes.Antecedente) = '" +
        ConsequenteStr + "'" AND RTRIM(Funcoes.TermoLing) = '" +
        Regras(RegraNumero).Consequente + "'"
If rs.RecordCount <> 0 Then
    NFuncoes = rs.RecordCount 'Numero de Funcoes
    rs.MoveFirst
    Contador = 0
    While Not rs.EOF
        ReDim Preserve Consequente(Contador)
        Consequente(Contador).Descricao = Trim(rs!antecedente) + " - " +
            Trim(rs!Termoling)
        Consequente(Contador).CoefAng = rs!CoefAng
        Consequente(Contador).CoefLin = rs!CoefLin
        Consequente(Contador).IntervInic = rs!IntervInic
        Consequente(Contador).IntervFim = rs!IntervFim
        Contador = Contador + 1
        rs.MoveNext
    Wend
End If
MontaRegra = 1
Exit Function
End Function

Public Function BuscaDadosMunicipio(IdMunicipio As Long, StringConexao As String) As String
    Dim cn As ADODB.Connection
    Dim rs As ADODB.Recordset
    Set cn = New ADODB.Connection
    Set rs = New ADODB.Recordset
    cn.Open StringConexao
    Set rs.ActiveConnection = cn
    rs.CursorLocation = adUseServer
    rs.Open "Select * from Customers"
    If rs.RecordCount <> 0 Then
        Dim StrResult As String
        On Error GoTo ErrHandler
        rs.MoveFirst
        While Not rs.EOF
            StrResult = StrResult & rs!ContactName & "<br>"
            rs.MoveNext
        Wend
        BuscaDadosMunicipio = StrResult
    Else
        BuscaDadosMunicipio = "Não encontrado!"
    End If
    Dim myMail
    Exit Function
ErrHandler:
    If Err.Number = 3021 Then
        BuscaDadosMunicipio = "Nao Encontrado"
    Else
        Err.Raise Err.Number, Err.Source, Err.Description
    End If
End Function

Public Function FazerInferencia() As String
Dim RespDesFuzzi As Double
Dim x As Integer
    Resp = CarregarAntecedentes()
    Resp = FuzzificarEntradas(Ent2, Ent3, Ent4, Ent5, Ent6)
    Resp = CarregarRegras()
    Resp = MontaRegra(0)

```

```

If Resp = 1 Then
    Resp = FuncaoMinima(Antecedente2, Entrada2, 2)
    Resp = FuncaoMinima(Antecedente3, Entrada3, 3)
    Resp = FuncaoMinima(Antecedente4, Entrada4, 4)
    Resp = FuncaoMinima(Antecedente5, Entrada5, 5)
    Resp = FuncaoMinima(Antecedente6, Entrada6, 6)
    Alpha = Minimo(Minimo(Supremo(fAux2), Minimo(Supremo(fAux3),
        Supremo(fAux4))), Minimo(Supremo(fAux5), Supremo(fAux6)))
    ReDim fAux2(0)
    ReDim fAux3(0)
    ReDim fAux4(0)
    ReDim fAux5(0)
    ReDim fAux6(0)
    ReDim FuncaoAlpha(0)
    FuncaoAlpha(0).IntervInic = Consequente(0).IntervInic
    FuncaoAlpha(0).IntervFim = Consequente(UBound(Consequente)).IntervFim
    FuncaoAlpha(0).CoefAng = 0
    FuncaoAlpha(0).CoefLin = Alpha
    Resp = FuncaoMinima(FuncaoAlpha, Consequente, 7)
    For x = 1 To NRegras - 1
        Resp = MontaRegra(x)
        If Resp = 1 Then
            Resp = FuncaoMinima(Antecedente2, Entrada2, 2)
            Resp = FuncaoMinima(Antecedente3, Entrada3, 3)
            Resp = FuncaoMinima(Antecedente4, Entrada4, 4)
            Resp = FuncaoMinima(Antecedente5, Entrada5, 5)
            Resp = FuncaoMinima(Antecedente6, Entrada6, 6)
            Alpha = Minimo(Minimo(Supremo(fAux2), Minimo(Supremo(fAux3),
                Supremo(fAux4))), Minimo(Supremo(fAux5),
                Supremo(fAux6)))
            ReDim fAux1(0)
            ReDim fAux2(0)
            ReDim fAux3(0)
            ReDim fAux4(0)
            ReDim fAux5(0)
            ReDim fAux6(0)
            FuncaoAlpha(0).CoefAng = 0
            FuncaoAlpha(0).CoefLin = Alpha
            ReDim fAux2(UBound(Resposta))
            fAux2 = Resposta
            Resp = FuncaoMinima(FuncaoAlpha, Consequente, 7)
            ReDim fAux1(UBound(Resposta))
            fAux1 = Resposta
            Resp = FuncaoMaxima(fAux1, fAux2)
            ReDim fAux1(0)
            ReDim fAux2(0)
        End If
    Next
End If
RespDesFuzzi = DesFuzzificar(Resposta)
FazerInferencia = MontarResposta(RespDesFuzzi)
End Function

Public Function DesFuzzificar(f() As FuncaoAuxiliar) As Double
    Dim IntNum As Double
    Dim IntDen As Double
    IntNum = 0
    IntDen = 0
    For x = 0 To UBound(f)
        If f(x).IntervInic = 0 Then f(x).IntervInic = 0.1
        If f(x).IntervFim = 0 Then f(x).IntervFim = 100
        IntNum = IntNum + f(x).CoefAng * Exp(Log(f(x).IntervFim) * 3) / 3 +
            f(x).CoefLin * Exp(Log(f(x).IntervFim) * 2) / 2 - f(x).CoefAng *
            Exp(Log(f(x).IntervInic) * 3) / 3 - f(x).CoefLin *
            Exp(Log(f(x).IntervInic) * 2) / 2
    Next

```

```

        IntDen = IntDen + f(x).CoefAng * Exp(Log(f(x).IntervFim) * 2) / 2 +
                f(x).CoefLin * f(x).IntervFim - f(x).CoefAng *
                Exp(Log(f(x).IntervInic) * 2) / 2 - f(x).CoefLin * f(x).IntervInic
    Next
    DesFuzzificar = IntNum / IntDen
End Function

Public Function MontarResposta(d As Double) As String
    Dim StrResult As String
    Dim Temp1 As Double
    Dim Temp2 As String
    'Consequente
    rs.Close
    Set rs.ActiveConnection = cn
    rs.CursorLocation = adUseServer
    rs.Open "select distinct potencialidade from regras"
    If rs.RecordCount <> 0 Then
        NFuncoes = rs.RecordCount 'Numero de Consequentes
        rs.MoveFirst
        Contador = 0
        While Not rs.EOF
            ReDim Preserve ConsequentesResp(Contador)
            ReDim PercentuaisResp(Contador)
            ConsequentesResp(Contador) = Trim(rs!Potencialidade)
            Contador = Contador + 1
            rs.MoveNext
        Wend
    End If
    For x = 0 To UBound(ConsequentesResp)
        rs.Close
        Set rs.ActiveConnection = cn
        rs.CursorLocation = adUseServer
        rs.Open "Select * from Funcoes where RTRIM(Funcoes.Antecedente) = '" +
                ConsequenteStr + "' AND RTRIM(Funcoes.TermoLing) = '" +
                ConsequentesResp(x) + "'"
        If rs.RecordCount <> 0 Then
            NFuncoes = rs.RecordCount 'Numero de Funcoes
            rs.MoveFirst
            Contador = 0
            ReDim Consequente(0)
            While Not rs.EOF
                ReDim Preserve Consequente(Contador)
                Consequente(Contador).Descricao = Trim(rs!antecedente) + " - " +
                    Trim(rs!Termoling)
                Consequente(Contador).CoefAng = rs!CoefAng
                Consequente(Contador).CoefLin = rs!CoefLin
                Consequente(Contador).IntervInic = rs!IntervInic
                Consequente(Contador).IntervFim = rs!IntervFim
                Contador = Contador + 1
                rs.MoveNext
            Wend
            For y = 0 To UBound(Consequente)
                If Consequente(y).IntervInic <= d And Consequente(y).IntervFim >= d
                    Then
                        PercentuaisResp(x) = (Consequente(y).CoefAng * d +
                            Consequente(y).CoefLin) * 100
                    End If
                Next
            End If
        End If
    Next
    For x = 0 To UBound(PercentuaisResp) - 1
        For y = x + 1 To UBound(PercentuaisResp)
            If PercentuaisResp(x) < PercentuaisResp(y) Then
                Temp1 = PercentuaisResp(x)
                PercentuaisResp(x) = PercentuaisResp(y)
                PercentuaisResp(y) = Temp1
            End If
        Next
    Next
End Function

```

```

        Temp2 = ConsequentesResp(x)
        ConsequentesResp(x) = ConsequentesResp(y)
        ConsequentesResp(y) = Temp2
    End If
Next
Next
StrResult = StrResult + "<p>" + "Prezado(a) Sr(a). " + Usuario + "</p>"
For x = 0 To UBound(ConsequentesResp)
    If (PercentuaisResp(x) / 100 > 0.01) Then
        StrResult = StrResult + "<p>" + "A Região escolhida é de <b>" +
            ConsequentesResp(x) + "</b> Potencialidade com <b>" +
            CStr(Format(PercentuaisResp(x) / 100, "0.00%")) +
            "</b> pontos percentuais" + Chr(13)
    End If
Next
StrResult = StrResult + "<p>A Região é formada pelos Municípios: </p> <b>" +
    RegiaoBase + "</b></body></html>"
rs.Close
rs.Open "Select * from usuarios where Login = '" & Usuario & "'"

If rs.RecordCount <> 0 Then
    'Resp = EnviaMail(StrResult, rs!Mail)
End If
rs.Close
MontarResposta = StrResult
End Function

Public Function EnviaMail(StrMail As String, MailTo As String) As Integer
    Dim myMail
    CorpMail = StrMail
    Set myMail = CreateObject("CDONTS.NewMail")
    myMail.From = "sicsu@globo.com"
    myMail.To = MailTo
    myMail.CC = "orlando@eng.uerj.br"
    myMail.Subject = "Resultado gerado pelo Sistema SIGAI-API."
    myMail.Body = StrMail
    myMail.BodyFormat = 0
    myMail.MailFormat = 0
    myMail.Importance = 2
    myMail.Value("Reply-To") = "Sicsú<sicsu@globo.com>"
    myMail.Send
    Set myMail = Nothing
    EnviaMail = 1
End Function

Public Function CarregarRegras() As Integer
    Dim Contador As Integer
    Set rs.ActiveConnection = cn
    rs.CursorLocation = adUseServer
    rs.Open "Select * from Regras"
    If rs.RecordCount <> 0 Then
        rs.MoveFirst
        Contador = 0
        While Not rs.EOF
            ReDim Preserve Regras(Contador)
            Regras(Contador).Antecedente2 = Trim(rs!Conexao)
            Regras(Contador).Antecedente3 = Trim(rs!Urbanizacao)
            Regras(Contador).Antecedente4 = Trim(rs!Populacao)
            Regras(Contador).Antecedente5 = Trim(rs!Concorrenciia)
            Regras(Contador).Antecedente6 = Trim(rs!Renda)
            Regras(Contador).Consequente = Trim(rs!Potencialidade)
            Contador = Contador + 1
            rs.MoveNext
        Wend
        NRegras = Contador 'Número de Regras
        CarregaRegras = 1
    End If
End Function

```

```

End If
End Function

```

```

Public Function FuzzificarEntradas(e2 As Double, e3 As Double, e4 As Double, e5 As
Double, e6 As Double) As Integer
    ReDim Entrada2(0)
    ReDim Entrada3(0)
    ReDim Entrada4(0)
    ReDim Entrada5(0)
    ReDim Entrada6(0)
    If e2 > Antecedentes(1).IntervaloFinal Then e2 = Antecedentes(1).IntervaloFinal
    If e2 < 0 Then e2 = 0
    Entrada2(0).IntervInic = e2
    Entrada2(0).IntervFim = e2
    Entrada2(0).CoefLin = 1
    Entrada2(0).CoefAng = 0
    If e3 > Antecedentes(2).IntervaloFinal Then e3 = Antecedentes(2).IntervaloFinal
    If e3 < 0 Then e3 = 0
    Entrada3(0).IntervInic = e3
    Entrada3(0).IntervFim = e3
    Entrada3(0).CoefLin = 1
    Entrada3(0).CoefAng = 0
    If e4 > Antecedentes(3).IntervaloFinal Then e4 = Antecedentes(3).IntervaloFinal
    If e4 < 0 Then e4 = 0
    Entrada4(0).IntervInic = e4
    Entrada4(0).IntervFim = e4
    Entrada4(0).CoefLin = 1
    Entrada4(0).CoefAng = 0
    If e5 > Antecedentes(4).IntervaloFinal Then e5 = Antecedentes(4).IntervaloFinal
    If e5 < 0 Then e5 = 0
    Entrada5(0).IntervInic = e5
    Entrada5(0).IntervFim = e5
    Entrada5(0).CoefLin = 1
    Entrada5(0).CoefAng = 0
    If e6 > Antecedentes(5).IntervaloFinal Then e6 = Antecedentes(5).IntervaloFinal
    If e6 < 0 Then e6 = 0
    Entrada6(0).IntervInic = e6
    Entrada6(0).IntervFim = e6
    Entrada6(0).CoefLin = 1
    Entrada6(0).CoefAng = 0
    FuzzificarEntradas = 1

```

```

End Function

```

```

Public Function FuncaoMinima(Func1() As FuncaoAuxiliar, Func2() As FuncaoAuxiliar, f
As Integer) As Integer
    Dim fAux() As FuncaoAuxiliar
    Dim C1 As Integer 'Contador de Partes da Func1
    Dim C2 As Integer 'Contador de Partes da Func2
    Dim C3 As Integer 'Contador de Partes da Resposta
    Dim C4 As Integer 'Contador de Partes da Ultima
    Dim C5 As Integer 'Contador de Partes da fAux
    Dim BordaAux As Double
    Dim f1e As Double
    Dim f1d As Double
    Dim f2e As Double
    Dim f2d As Double
    Dim Terminou As Boolean
    Dim QuebrouIntervalo As Boolean
    C1 = 0
    C2 = 0
    C3 = 0
    C4 = 0
    C5 = 0
    ReDim fAux(0)
    Terminou = False

```

```

If (Func1(0).IntervInic <> Func1(0).IntervFim) And (Func2(0).IntervInic <>
Func2(0).IntervFim) Then
  QuebrouIntervalo = False
  While Terminou = False
    If QuebrouIntervalo Then
      QuebrouIntervalo = False
      fAux(C5).IntervInic = BordaAux
    ElseIf Func1(C1).IntervInic >= Func2(C2).IntervInic Then
      fAux(C5).IntervInic = Func1(C1).IntervInic
    Else
      fAux(C5).IntervInic = Func2(C2).IntervInic
    End If
    If Func1(C1).IntervFim <= Func2(C2).IntervFim Then
      BordaAux = Func1(C1).IntervFim
    Else
      BordaAux = Func2(C2).IntervFim
    End If
    f1e = Func1(C1).CoefAng * fAux(C5).IntervInic + Func1(C1).CoefLin
    f2e = Func2(C2).CoefAng * fAux(C5).IntervInic + Func2(C2).CoefLin
    f1d = Func1(C1).CoefAng * BordaAux + Func1(C1).CoefLin
    f2d = Func2(C2).CoefAng * BordaAux + Func2(C2).CoefLin
    If f1e < 0.0000000001 Then f1e = 0
    If f1d < 0.0000000001 Then f1d = 0
    If f2e < 0.0000000001 Then f2e = 0
    If f2d < 0.0000000001 Then f2d = 0
    If ((Abs(f1e - f2e) < 0.000000001) And (f1d >= f2d)) Or ((f1e > f2e)
      And(f1d >= f2d)) Then
      fAux(C5).CoefAng = Func2(C2).CoefAng
      fAux(C5).CoefLin = Func2(C2).CoefLin
      fAux(C5).IntervFim = BordaAux
    ElseIf ((Abs(f1e - f2e) < 0.000000001) And (f1d < f2d)) Or ((f2e > f1e)
      And (f2d >= f1d)) Then
      fAux(C5).CoefAng = Func1(C1).CoefAng
      fAux(C5).CoefLin = Func1(C1).CoefLin
      fAux(C5).IntervFim = BordaAux
    ElseIf ((f1e > f2e) And (f1d < f2d)) Or ((f2e > f1e) And (f2d < f1d))
      Then
      BordaAux = (Func2(C2).CoefLin - Func1(C1).CoefLin) /
        (Func1(C1).CoefAng - Func2(C2).CoefAng)
      fAux(C5).IntervFim = BordaAux
      QuebrouIntervalo = True
      If f1e > f2e Then
        fAux(C5).CoefAng = Func2(C2).CoefAng
        fAux(C5).CoefLin = Func2(C2).CoefLin
      Else
        fAux(C5).CoefAng = Func1(C1).CoefAng
        fAux(C5).CoefLin = Func1(C1).CoefLin
      End If
    End If
    If (UBound(fAux) > 0) Then
      If (Abs(fAux(C5 - 1).CoefAng - fAux(C5).CoefAng) < 0.000000001) And
        (Abs(fAux(C5 - 1).CoefLin - fAux(C5).CoefLin) < 0.000000001)
        Then
        fAux(C5 - 1).IntervFim = fAux(C5).IntervFim
        ReDim Preserve fAux(C5 - 1)
        C5 = C5 - 1
      End If
    End If
    If (C1 = UBound(Func1) And C2 = UBound(Func2) And Not QuebrouIntervalo)
      Then
      Terminou = True
    Else
      C5 = C5 + 1
      ReDim Preserve fAux(C5)
    End If
  End While
End If

```

```

        If Not QuebrouIntervalo Then
            If Abs(Func1(C1).IntervFim - BordaAux) < 0.000000001 Then C1 = C1 + 1
            If Abs(Func2(C2).IntervFim - BordaAux) < 0.000000001 Then C2 = C2 + 1
        End If
    Wend
Else
    If UBound(Func1) > 0 Then 'Func1 = Funcao e Func2 = Ponto
        fAux(0).IntervInic = Func1(0).IntervInic
        fAux(0).IntervFim = Func1(0).IntervFim
        fAux(0).CoefAng = Func2(0).CoefAng
        fAux(0).CoefLin = Minimo(AvaliaFuncao(Func1, Func2(0).IntervInic),
                                Func2(0).CoefLin)
    Else
        fAux(0).IntervInic = Func2(0).IntervInic
        fAux(0).IntervFim = Func2(0).IntervFim
        fAux(0).CoefAng = Func1(0).CoefAng
        fAux(0).CoefLin = Minimo(AvaliaFuncao(Func2, Func1(0).IntervInic),
                                Func1(0).CoefLin)
    End If
End If
If (f = 1) Then
    ReDim fAux1(UBound(fAux))
    fAux1 = fAux
End If
If (f = 2) Then
    ReDim fAux2(UBound(fAux))
    fAux2 = fAux
End If
If (f = 3) Then
    ReDim fAux3(UBound(fAux))
    fAux3 = fAux
End If
If (f = 4) Then
    ReDim fAux4(UBound(fAux))
    fAux4 = fAux
End If
If (f = 5) Then
    ReDim fAux5(UBound(fAux))
    fAux5 = fAux
End If
If (f = 6) Then
    ReDim fAux6(UBound(fAux))
    fAux6 = fAux
End If
If (f = 7) Then
    ReDim Resposta(UBound(fAux))
    Resposta = fAux
End If
FuncaoMinima = 1
End Function

Public Function FuncaoMaxima(Func1() As FuncaoAuxiliar, Func2() As FuncaoAuxiliar)
As Integer
    Dim fAux() As FuncaoAuxiliar
    Dim C1 As Integer 'Contador de Partes da Func1
    Dim C2 As Integer 'Contador de Partes da Func2
    Dim C3 As Integer 'Contador de Partes da Resposta
    Dim C4 As Integer 'Contador de Partes da Ultima
    Dim C5 As Integer 'Contador de Partes da fAux
    Dim BordaAux As Double
    Dim f1e As Double
    Dim f1d As Double
    Dim f2e As Double
    Dim f2d As Double
    Dim Terminou As Boolean
    Dim QuebrouIntervalo As Boolean

```

```

C1 = 0
C2 = 0
C3 = 0
C4 = 0
C5 = 0
ReDim fAux(0)
Terminou = False
If (Func1(0).IntervInic <> Func1(0).IntervFim) And (Func2(0).IntervInic <>
    Func2(0).IntervFim) Then
    QuebrouIntervalo = False
    While Terminou = False
        If QuebrouIntervalo Then
            QuebrouIntervalo = False
            fAux(C5).IntervInic = BordaAux
        ElseIf Func1(C1).IntervInic >= Func2(C2).IntervInic Then
            fAux(C5).IntervInic = Func1(C1).IntervInic
        Else
            fAux(C5).IntervInic = Func2(C2).IntervInic
        End If
        If Func1(C1).IntervFim <= Func2(C2).IntervFim Then
            BordaAux = Func1(C1).IntervFim
        Else
            BordaAux = Func2(C2).IntervFim
        End If
        f1e = Func1(C1).CoefAng * fAux(C5).IntervInic + Func1(C1).CoefLin
        f2e = Func2(C2).CoefAng * fAux(C5).IntervInic + Func2(C2).CoefLin
        f1d = Func1(C1).CoefAng * BordaAux + Func1(C1).CoefLin
        f2d = Func2(C2).CoefAng * BordaAux + Func2(C2).CoefLin
        If f1e < 0.0000000001 Then f1e = 0
        If f1d < 0.0000000001 Then f1d = 0
        If f2e < 0.0000000001 Then f2e = 0
        If f2d < 0.0000000001 Then f2d = 0
        If ((Abs(f1e - f2e) < 0.0000000001) And (f1d >= f2d)) Or ((f1e > f2e)
            And (f1d >= f2d)) Then
            fAux(C5).CoefAng = Func1(C1).CoefAng
            fAux(C5).CoefLin = Func1(C1).CoefLin
            fAux(C5).IntervFim = BordaAux
        ElseIf ((Abs(f1e - f2e) < 0.0000000001) And (f1d < f2d)) Or ((f2e >
            f1e)
            And (f2d >= f1d)) Then
            fAux(C5).CoefAng = Func2(C2).CoefAng
            fAux(C5).CoefLin = Func2(C2).CoefLin
            fAux(C5).IntervFim = BordaAux
        ElseIf ((f1e > f2e) And (f1d < f2d)) Or ((f2e > f1e) And (f2d < f1d))
            Then
            BordaAux = (Func2(C2).CoefLin - Func1(C1).CoefLin) /
                (Func1(C1).CoefAng - Func2(C2).CoefAng)
            fAux(C5).IntervFim = BordaAux
            QuebrouIntervalo = True
            If f1e > f2e Then
                fAux(C5).CoefAng = Func1(C1).CoefAng
                fAux(C5).CoefLin = Func1(C1).CoefLin
            Else
                fAux(C5).CoefAng = Func2(C2).CoefAng
                fAux(C5).CoefLin = Func2(C2).CoefLin
            End If
        End If
        If (UBound(fAux) > 0) Then
            If (Abs(fAux(C5 - 1).CoefAng - fAux(C5).CoefAng) < 0.0000000001) And
                (Abs(fAux(C5 - 1).CoefLin - fAux(C5).CoefLin) < 0.0000000001)
                Then
                fAux(C5 - 1).IntervFim = fAux(C5).IntervFim
                ReDim Preserve fAux(C5 - 1)
                C5 = C5 - 1
            End If
        End If
    End If

```

```

        If (C1 = UBound(Func1) And C2 = UBound(Func2) And Not QuebrouIntervalo)
            Then
                Terminou = True
            Else
                C5 = C5 + 1
                ReDim Preserve fAux(C5)
            End If
            If Not QuebrouIntervalo Then
                If Abs(Func1(C1).IntervFim - BordaAux) < 0.000000001 Then C1 = C1 + 1
                If Abs(Func2(C2).IntervFim - BordaAux) < 0.000000001 Then C2 = C2 + 1
            End If
        Wend
    Else
        If UBound(Func1) > 0 Then 'Func1 = Funcao e Func2 = Ponto
            fAux(0).IntervInic = Func1(0).IntervInic
            fAux(0).IntervFim = Func1(0).IntervFim
            fAux(0).CoefAng = Func2(0).CoefAng
            fAux(0).CoefLin = Maximo(AvaliaFuncao(Func1, Func2(0).IntervInic),
                                    Func2(0).CoefLin)
            'Debug.Print fAux(0).CoefAng
            'Debug.Print fAux(0).CoefLin

        Else 'Func1 = Ponto e Func2 = Funcao
            fAux(0).IntervInic = Func2(0).IntervInic
            fAux(0).IntervFim = Func2(0).IntervFim
            fAux(0).CoefAng = Func1(0).CoefAng
            fAux(0).CoefLin = Maximo(AvaliaFuncao(Func2, Func1(0).IntervInic),
                                    Func1(0).CoefLin)
        End If
    End If
    ReDim Resposta(UBound(fAux))
    Resposta = fAux
    FuncaoMaxima = 1
End Function

Public Function Minimo(n1 As Double, n2 As Double) As Double
    If (n1 < n2) Then
        Minimo = n1
    Else
        Minimo = n2
    End If
End Function

Public Function Maximo(n1 As Double, n2 As Double) As Double
    If (n1 > n2) Then
        Maximo = n1
    Else
        Maximo = n2
    End If
End Function

Public Function Supremo(F1() As FuncaoAuxiliar) As Double
    Dim NTermos As Integer 'Numero de Termos
    Dim Contador As Integer
    NTermos = UBound(F1)
    If NTermos > 0 Then
        For Contador = 0 To NTermos
            Supremo = Maximo(Supremo, Maximo(F1(Contador).CoefAng *
            F1(Contador).IntervInic + F1(Contador).CoefLin, F1(Contador).CoefAng *
            F1(Contador).IntervFim + F1(Contador).CoefLin))
        Next
    Else
        Supremo = F1(0).CoefLin
    End If
End Function

```

```

Public Function GeraRegiao(Municipio As String) As String
    Dim NMunicipio As Integer
    Dim GRegiao As String
    Dim LatMun As Double
    Dim LongMun As Double
    Dim TConexao As Double
    Dim TUrbanizacao As Double
    Dim TPopulacao As Double
    Dim TPopulacaoRenda As Double
    Dim TConcorrencencia As Double
    Dim TRenda As Double
    Dim ContMunic As Double
    Dim Urb1 As Double
    Dim Urb2 As Double
    Dim Urb3 As Double
    Dim Urb4 As Double
    ContMunic = 0
    TPopulacao = 0
    TPopulacaoRenda = 0
    TRenda = 0
    TConcorrencencia = 0
    TUrbanizacao = 0
    TConexao = 0
    Urb1 = 0
    Urb2 = 0
    Urb3 = 0
    Urb4 = 0
    GRegiao = Municipio + " "
    Set rs.ActiveConnection = cn
    Set rs2.ActiveConnection = cn
    rs.CursorLocation = adUseServer
    rs.Open "Select * from Sintese where CODIGO = '" + Trim(Municipio) + "'"
    rs2.CursorLocation = adUseServer
    rs2.Open "Select * from Municipios where CODIGO = '" + Trim(Municipio) + "'"
    If rs.RecordCount <> 0 Then
        LatMun = rs!Latitude
        LongMun = rs!Longitude
        RegiaoBase = "<p> Município Base:" + rs!N101 + "<br>Muncípios a serem  
atendidos na Região:</p>"
        'Populacao
        TPopulacao = TPopulacao + rs!n1
        'Renda
        If (rs!n88 > 0 And rs!n89 > 0) Then
            TRenda = TRenda + ((rs!n88 + rs!n89) * 1000)
            TPopulacaoRenda = TPopulacaoRenda + rs!n1
        End If
        'Concorrencencia
        TConcorrencencia = TConcorrencencia + rs2!provedores
        'Conexao
        If (rs2!Banda = 0) Then TConexao = TConexao + (64 * rs!n1)
        If (rs2!Banda = 1) Then TConexao = TConexao + (256 * rs!n1)
        If (rs2!Banda = 2) Then TConexao = TConexao + (512 * rs!n1)
        'Urbanizacao
        Urb1 = ((rs!n87) / (rs!n1)) * 25 / 0.0001615768
        If (Urb1 > 25) Then Urb1 = 25
        Urb2 = ((rs!n39) / (rs!n1)) * 25 / 0.00511
        If (Urb2 > 25) Then Urb2 = 25
        Urb3 = ((rs!n54 - rs!n77) / (rs!n1)) * 25 / 0.024086
        If (Urb3 > 25) Then Urb3 = 25
        Urb4 = ((rs!n35 + rs!n36 + rs!n37) / (rs!n1)) * 25 / 0.0008102259
        If (Urb4 > 25) Then Urb4 = 25
        TUrbanizacao = TUrbanizacao + (Urb1 + Urb2 + Urb3 + Urb4)
        ContMunic = ContMunic + 1
    rs2.Close

```

```

Else
    GeraRegiao = ""
End If
rs.Close
Set rs.ActiveConnection = cn
rs.CursorLocation = adUseServer
rs.Open "Select * from Sintese"
If rs.RecordCount <> 0 Then
    NMunicipio = rs.RecordCount
    rs.MoveFirst
    Contador = 0
    RegiaoBase = RegiaoBase + "<tr>"
    While Not rs.EOF
        If (rs!codigo <> Municipio) Then
            Distancia = CalculaDistancia(rs!Latitude, rs!Longitude, LatMun, LongMun)
            If Distancia <= 50 Then 'km
                rs2.Open "Select * from Municipios where CODIGO = '" + Trim(rs!codigo) + "'"
                GRegiao = GRegiao + rs!codigo + " "
                RegiaoBase = RegiaoBase + "<td><b>" + Trim(rs!N101) + "</b></td>"
                'Populacao
                TPopulacao = TPopulacao + rs!n1
                'Renda
                If (rs!n88 > 0 And rs!n89 > 0) Then
                    TRenda = TRenda + ((rs!n88 + rs!n89) * 1000)
                    TPopulacaoRenda = TPopulacaoRenda + rs!n1
                End If
                'Concorrencencia
                TConcorrencencia = TConcorrencencia + rs2!provedores
                'Conexao
                If (rs2!Banda = 0) Then TConexao = TConexao + (64 * rs!n1)
                If (rs2!Banda = 1) Then TConexao = TConexao + (256 * rs!n1)
                If (rs2!Banda = 2) Then TConexao = TConexao + (512 * rs!n1)
                'Urbanizacao
                Urb1 = ((rs!n87) / (rs!n1)) * 25 / 0.0001615768
                If (Urb1 > 25) Then Urb1 = 25
                If (Urb1 < 0) Then Urb1 = 0
                Urb2 = ((rs!n39) / (rs!n1)) * 25 / 0.00511
                If (Urb2 > 25) Then Urb2 = 25
                If (Urb2 < 0) Then Urb2 = 0
                Urb3 = ((rs!n54 - rs!n77) / (rs!n1)) * 25 / 0.024086
                If (Urb3 > 25) Then Urb3 = 25
                If (Urb3 < 0) Then Urb3 = 0
                Urb4 = ((rs!n35 + rs!n36 + rs!n37) / (rs!n1)) * 25 / 0.0008102259
                If (Urb4 > 25) Then Urb4 = 25
                If (Urb4 < 0) Then Urb4 = 0
                TUrbanizacao = TUrbanizacao + (Urb1 + Urb2 + Urb3 + Urb4)
                ContMunic = ContMunic + 1
                If (ContMunic = 4 Or ContMunic = 7 Or ContMunic = 10 Or ContMunic = 13 Or ContMunic = 16 Or ContMunic = 19 Or ContMunic = 22 Or ContMunic = 25 Or ContMunic = 28) Then
                    RegiaoBase = RegiaoBase + "</tr><tr>"
                End If
                rs2.Close
            End If
            rs.MoveNext
        End If
    Wend
    RegiaoBase = RegiaoBase + "</tr>"
End If
rs.Close
TRenda = TRenda / TPopulacaoRenda
TConcorrencencia = TConcorrencencia / ContMunic
TConexao = TConexao / Tpopulacao

```

```

TURbanizacao = TURbanizacao / ContMunic
Ent2 = TConexao
Ent3 = TURbanizacao
Ent4 = TPopulacao
Ent5 = TConcorrenzia
Ent6 = TRenda
GeraRegiao = GRegiao
End Function

Public Function CalculaDistancia(Lat1 As Double, Long1 As Double, Lat2 As Double,
Long2 As Double) As Double
    Dim seMaior As Double
    Dim df As Double
    Dim f As Double
    Dim seMenor As Double
    Dim e2 As Double
    Dim Pi As Double
    Dim dSigma1 As Double
    Dim dSigma2 As Double
    Dim dSigma As Double
    Dim FiA As Double
    Dim FiB As Double
    Dim LambA As Double
    Dim LambB As Double
    Dim PsiA As Double
    Dim PsiB As Double
    Dim dl As Double
    Dim AlfaAB As Double
    Dim AlfaBA As Double
    Dim Psi0 As Double
    Dim AlfaE As Double
    Dim dLong As Double
    Dim SigmaA As Double
    Dim SigmaM As Double
    Dim a As Double
    Dim b As Double
    Dim C As Double
    Dim e22 As Double
    Dim K2 As Double
    Dim A2 As Double
    Dim B2 As Double
    Dim C2 As Double
    Dim D2 As Double
    Dim s As Double
    Dim CosSigma As Double
    Dim TGSigma As Double
    Dim TSeno As Double
    Dim TCos As Double
    Dim TTan As Double
    Dim TATan As Double
    Pi = 3.14159265358979
    'Parâmetros do elipsoide (SAD-69)
    seMaior = 6378160
    df = 298.25
    'Cálculos Auxiliares
    f = 1 / df
    seMenor = seMaior + (-seMaior * f)
    e2 = 2 * f - (f ^ 2)
    FiA = Lat1      'Lat1
    LambA = Long1   'Long1
    FiB = Lat2      'Lat2
    LambB = Long2   'Long2
    'Transformar Graus para Radianos FiA, FiB, LambA e LambB
    FiA = Pi * FiA / 180
    LambA = Pi * LambA / 180
    FiB = Pi * FiB / 180

```

```

LambB = Pi * LambB / 180
'Passo 1
PsiA = Atn(seMenor / seMaior * Tan(FiA))
PsiB = Atn(seMenor / seMaior * Tan(FiB))
dl = LambB - LambA
Do
    dl2 = dl
    ' Passo 2 - Cálculo de dSigma
    dSigma1 = ((Sin(dl) * Cos(PsiB)) ^ 2)
    dSigma2 = ((Sin(PsiB)*Cos(PsiA)-Sin(PsiA) * Cos(PsiB) * Cos(dl)) ^2)
    dSigma = ((dSigma1 + dSigma2) ^ (0.5))
    CosSigma = ((1 - (dSigma ^ 2)) ^ (0.5))
    TGSigma = dSigma / CosSigma
    dSigma = Atn(TGSigma)
    ' Passo3 - Cálculo de AlfaAB
    AlfaAB = Atn(Sin(dl) * Cos(PsiB) / (Sin(PsiB) * Cos(PsiA) - Cos(dl)
    *
    Sin(PsiA) * Cos(PsiB)))
    dLong = LambB - LambA
    ' Quadrante de AlfaAB
    If (dLong > 0 And Tan(AlfaAB) < 0 Or dLong < 0 And Tan(AlfaAB) > 0)
        Then AlfaAB = AlfaAB + Pi
    ' Passo 4 - Cálculo de Psi0
    TCos = Sin(AlfaAB) * Cos(PsiA)
    TSeno = ((1 - (TCos ^ 2)) ^ (0.5))
    Psi0 = ((1 - (TCos ^ 2)) ^ (0.5)) / TCos
    Psi0 = Atn(Psi0)

    ' Passo 5 - Cálculo de SigmaM
    TSeno = Cos(PsiA) * Sin(AlfaAB)
    AlfaE = Atn(TSeno / ((1 - (TSeno ^ 2)) ^ (0.5)))
    TSeno = Sin(PsiA) / Cos(AlfaE)
    SigmaA = Atn(TSeno / ((1 - (TSeno ^ 2)) ^ (0.5)))
    SigmaM = SigmaA + (dSigma / 2)
    ' Passo 6 - Cálculo de dl
    a = 1 + (e2 / 4) + ((e2 ^ 2) / 8) - ((e2 / 8) * (Sin(Psi0) ^ 2)) -
        ((e2 ^ 2) / 8 * (Sin(Psi0) ^ 2)) + (3 / 64 * (e2 ^ 2) *
        (Sin(Psi0) ^ 4))
    b = (e2 / 8 * (Sin(Psi0) ^ 2)) + ((e2 ^ 2) / 8 * (Sin(Psi0) ^ 2)) -
        ((e2 ^ 2) / 16 * (Sin(Psi0) ^ 4))
    C = (e2 ^ 2) / 64 * (Sin(Psi0) ^ 4)
    dl = dLong + e2 / 2 * Cos(Psi0) * (a * dSigma + b * Cos(2 * SigmaM)
        * Sin(dSigma) + C / 2 * Cos(4 * SigmaM) * Sin(2 * dSigma))
    Loop While (dl = dl2) ' // retorna ao passo 2, se o dl antigo diferente
    ' do novo.

' Convergência
'AlfaAB = aSin(Cos(Psi0) / Cos(PsiA))
TSeno = Cos(Psi0) / Cos(PsiA)
AlfaAB = Atn(TSeno / ((1 - (TSeno ^ 2)) ^ (0.5)))
'AlfaBA = aSin(-Cos(Psi0) / Cos(PsiB))
TSeno = -Cos(Psi0) / Cos(PsiB)
AlfaBA = Atn(TSeno / ((1 - (TSeno ^ 2)) ^ (0.5)))
e22 = ((seMaior ^ 2) - (seMenor ^ 2)) / (seMenor ^ 2)
K2 = e22 * (Sin(Psi0) ^ 2)
A2 = 1 + K2 / 4 - 3 / 64 * (K2 ^ 2) + 5 / 256 * (K2 ^ 3)
B2 = -K2 / 4 + (K2 ^ 2) / 16 - 5 * (K2 ^ 3) / 512
C2 = -(K2 ^ 2) / 64 + 3 * (K2 ^ 3) / 256
D2 = -(K2 ^ 3) / 512
s = seMenor * (A2 * dSigma + B2 * Cos(2 * SigmaM) * Sin(dSigma) + C2 / 2
    * Cos(4 * SigmaM) * Sin(2 * dSigma) + D2 / 3 * Cos(6 * SigmaM) *
    Sin(3 * dSigma))
CalculaDistancia = s / 1000 'de m Para Km
End Function

```