

**UNIVERSIDADE DE SÃO PAULO
ESCOLA POLITÉCNICA**

**MODELAGEM, ANÁLISE E CONTROLE DE UMA MALHA VIÁRIA CONTROLADA
POR SEMÁFOROS**

**Dan Hukusima Okawa
Fabio Takeshi Hirose
Fagner Lemos Del Ghingaro**

**São Paulo
2005**

DAN HUKUSIMA OKAWA
FABIO TAKESHI HIROSE
FAGNER LEMOS DEL GHINGARO

150

note final

7,5 (sete e meia)



**MODELAGEM, ANÁLISE E CONTROLE DE UMA MALHA VIÁRIA CONTROLADA
POR SEMÁFOROS**

Monografia apresentada à Escola Politécnica da
Universidade de São Paulo para obtenção do
diploma de Engenheiro.

Orientadores:

Prof. Dr. Paulo Eigi Miyagi

Eng. Dra. Cristina Toshie Motohashi Matsusaki

São Paulo

2005

DEDICATÓRIA

Dedicamos este trabalho aos nossos amigos e familiares que sempre nos apoiaram, e ajudaram com seu amor, admiração e carinho, sendo os principais responsáveis por nossas conquistas.

AGRADECIMENTOS

Agradecimentos aos nossos orientadores Prof. Dr. Paulo Eigi Miyagi e Profa. Dra. Cristina Toshie Motohashi Matsusaki pela ajuda e orientação durante toda a elaboração deste projeto.

Agradecemos ao doutorando Fabrício Junqueira pela sua ajuda na elaboração do modelo no ProModel®.

Por último, mas não menos importante, agradecemos aos nossos colegas e familiares por seu apoio e convivência durante todo o curso.

RESUMO

O objetivo deste trabalho é modelar e simular uma malha viária, utilizando uma estratégia de controle de semáforos para a priorização de fluxo de um determinado veículo que necessite de urgência para se movimentar, como por exemplo uma ambulância ou viaturas policiais. O sistema de controle baseado no conceito de sistemas a eventos discretos (SED) pode, através de semáforos inteligentes auxiliados por sensores, alterar as características operacionais de acordo com um evento emergencial, permitindo que os veículos devidamente autorizados percorram sua trajetória em um tempo reduzido. Uma estratégia de controle dos semáforos, desenvolvida em um software de simulação como o ProModel®, é implementada em um protótipo composto por um Controlador Lógico Programável (CLP), um computador e uma boteira para entrada de sinais.

Palavras chave: controle de tráfego urbano, rotas emergenciais, sistemas a eventos discretos

ABSTRACT

The purpose of this study is to model and simulate a network, based on a dynamic sequential preemption method. This local detection-based method, try to clear the intersection ahead the emergency vehicle, by activating the traffic light and changing its state to green. Thus, an emergency vehicle is able to move faster and safer through the congested network. A prototype compound of a logical programmable controller (LPC) and a personal computer was built to implement the method and to show more details than the model.

Keywords. Urban traffic control, emergencial routes, discret event system.

LISTAS DE FIGURAS

Figura 1 - Evolução de uma variável de estado de um SVC em função do tempo.....	14
Figura 2 - Evolução do comportamento de um SED a partir da ocorrência de eventos.....	15
Figura 3 - Sistema em estudo.....	44
Figura 4 - Modelo em simulação	44
Figura 5 - Gráfico de comparação dos tempos de percurso.....	50
Figura 6 - Cruzamentos de uma malha viária.....	51
Figura 7- Layout dos equipamentos.....	54
Figura 8 - Diagrama estrutural do protótipo	57
Figura 9 - CLP e Visualização dos LEDs	58
Figura 10 - CLP e os sensores (Inputs).....	59
Figura 11 - Diagrama SFC.....	60
Figura 12 - Nomeação dos trechos das vias.....	61
Figura 13 - Descrição do supervisório.....	67
Figura 14 - Relação Cliente-Servidor	69
Figura 15 - Estrutura do Módulo de Controle.....	71
Figura 16 - Tela de Monitoração	72
Figura 17 - Exemplo de Diagrama de Caso de uso.....	74

LISTA DE TABELAS

Tabela 1 - Resultados obtidos com a simulação	51
Tabela 2 - Lista dos símbolos e endereços dos timers alocados no CLP.....	63
Tabela 3 - Lista dos símbolos e endereços das saídas alocadas no CLP	64
Tabela 4 - Lista dos símbolos e endereços das entradas alocadas no CLP.....	65

LISTA DE SIGLAS

CLP Controlador Lógico Programável

EV Emergency Vehicle

RUP Rational Unified Process

SED Sistemas à Eventos Discretos

SVC Sistemas a Variáveis Contínuas

SFC Sequential Function Chart

VB Visual Basic

SUMÁRIO

LISTA DE TABELAS	8
LISTA DE SIGLAS	9
1. INTRODUÇÃO	12
1.1. MOTIVAÇÃO E JUSTIFICATIVA	12
1.2. OBJETIVOS	12
1.3. METODOLOGIA DE TRABALHO	13
2. FUNDAMENTOS TEÓRICOS	14
2.1. SISTEMAS A EVENTOS DISCRETOS	14
2.2. MALHA VIÁRIA URBANA	17
3. SISTEMAS DE CONTROLE INTELIGENTE	20
3.1. REDES NEURAIS	22
3.2. ALGORITMOS GENÉTICOS	28
3.3. LÓGICA DIFUSA	30
3.4. SISTEMA ESPECIALISTA	34
3.4.1. PROCESSO DE APRENDIZAGEM DE UM SISTEMA ESPECIALISTA	37
3.5. SISTEMAS DE PRODUÇÃO	39
3.6. AQUISIÇÃO DE CONHECIMENTO	41
4. DESENVOLVIMENTO DO SISTEMA DE CONTROLE DE UMA MALHA VIÁRIA CONTROLADA POR SEMÁFOROS	43
4.1. ESTUDO DE CASO	43
4.2. DESCRIÇÃO DA MALHA VIÁRIA	43
4.3. IDENTIFICAÇÃO DO PROBLEMA	45
4.4. SOLUÇÕES PROPOSTAS/ADOTADA DA ESTRATÉGIA DE CONTROLE	45
5. REGRAS DE CONTROLE	47
5.1. DESENVOLVIMENTO DA SIMULAÇÃO	47
5.1.1. ESCOLHA DO SOFTWARE	47
5.1.2. CONSTRUÇÃO DO MODELO	48
5.1.3. ESTRATÉGIA OBTIDA	49
6. DESENVOLVIMENTO DO PROTÓTIPO	53

6.1. ARQUITETURA DE CONTROLE	53
6.2. ESTUDO DA PLATAFORMA DE HARDWARE E SOFTWARE	55
7. ESPECIFICAÇÃO DO PROTÓTIPO.....	57
7.1. PROGRAMAÇÃO DO CLP	59
7.1.1. ELABORAÇÃO DO SEQUENTIAL FUNCTION CHART (SFC).....	60
7.1.2. PROGRAMAÇÃO EM LADDER	62
7.2. CONSTRUÇÃO DO SUPERVISÓRIO.....	65
7.2.1. DESCRIÇÃO DO SUPERVISORES	65
7.2.2. MÓDULO DE COMUNICAÇÃO	67
7.2.3. MÓDULO DE CONTROLE	70
7.2.4. MÓDULO DE MONITORAÇÃO.....	71
7.2.5. DESENVOLVIMENTO DO SOFTWARE.....	73
7.3. OBSERVAÇÕES FINAIS DESTE CAPÍTULO	74
8. CONCLUSÃO	75
REFERÊNCIAS BIBLIOGRÁFICAS	76
ANEXOS	79
ANEXO A - Programação utilizada no simulador PROMODEL®	79
ANEXO B - Diagrama Ladder do ciclo de luzes de um semáforo.....	114
ANEXO C - Documentação do supervisório.....	117
ANEXO D - Codificação do supervisório	144

1. INTRODUÇÃO

1.1. MOTIVAÇÃO E JUSTIFICATIVA

Atualmente, poucos são os semáforos implantados nas ruas das cidades brasileiras que podem ser controlados e monitorados a distância. Assim, o fluxo de veículos é controlado apenas por semáforos de tempo fixo, não sendo possível à alteração imediata das configurações anteriormente estabelecidas.

Os semáforos dotados de inteligência, ou seja, aqueles em que o tempo de seus ciclos se altera em função da necessidade, apresentam um desempenho razoável em diversas taxas de fluxo, porém não existe um estudo de adaptação em relação à presença de veículos que necessitem de prioridade em meio ao tráfego e a facilitar a locomoção destes veículos.

1.2. OBJETIVOS

Obter uma estratégia de controle de priorização de rota, modelando uma malha viária e simulando o tráfego urbano controlado por semáforos de forma a reduzir o tempo de percurso de uma rota pré-estabelecida, baseado em entradas estatísticas, utilizando o conceito de sistemas a eventos discretos.

Para maior detalhamento do projeto, construir um protótipo que represente a detecção física de veículos emergenciais e que por meio de uma comunicação com um supervisório, faça o controle e monitoramento instantâneo do tráfego urbano baseando-se na modelagem e estratégia obtida na parte de simulação.

1.3. METODOLOGIA DE TRABALHO

Visando uma melhor compreensão dos objetivos e resultados deste projeto, uma metodologia de trabalho será utilizada envolvendo os seguintes tópicos:

- Busca de uma estratégia de controle do tráfego;
- Simulação do modelo a ser adotado;
- Avaliação e comparação de resultados obtidos na simulação;
- Construção de um protótipo para detalhamento do projeto.

2. FUNDAMENTOS TEÓRICOS

2.1. SISTEMAS A EVENTOS DISCRETOS

Sistema a eventos discretos (SED) é um sistema dinâmico, construído pelo homem (*man made system*), e cuja dinâmica é dirigida pela ocorrência de eventos discretos a intervalos em geral irregulares e desconhecidos. Entre a ocorrência de dois eventos consecutivos, o sistema permanece em um determinado estado. São exemplos de SED: os sistemas de manufatura, sistemas de filas, sistemas de tráfego e redes de computadores [Cury].

Uma vez que o comportamento dinâmico é baseado em regras estabelecidas pelo próprio homem [Ho, 1989], não existem leis físicas que possam modelar esses sistemas, ao contrário de Sistemas a Variáveis Contínuas (SVC), em que os sistemas são dirigidos pelo tempo e cuja dinâmica é comandada por variações contínuas (eventos e estados).

Nos sistemas SVC, o comportamento dinâmico pode ser modelado por formulações matemáticas, baseado na teoria de cálculo integral e diferencial [Ogata, 1982].

Na figura seguinte, pode-se observar um exemplo da evolução de uma variável de estado em função do tempo no Sistemas a Variáveis Contínuas.

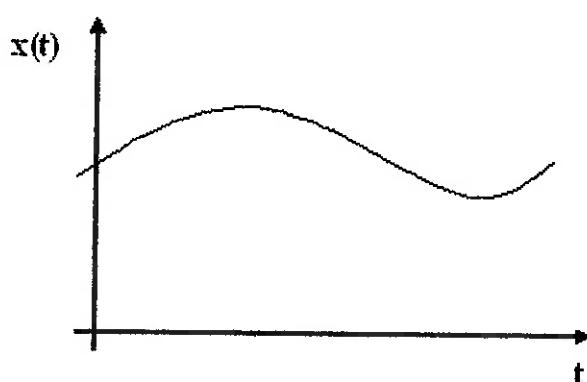


Figura 1 - Evolução de uma variável de estado de um SVC em função do tempo

Em SED, os eventos que ocorrem são geralmente instantâneos, pequenos se comparados à duração do estado, sendo assim classificados como discretos. A seguir encontra-se um gráfico da evolução dos estados de um SED a partir da ocorrência de eventos.

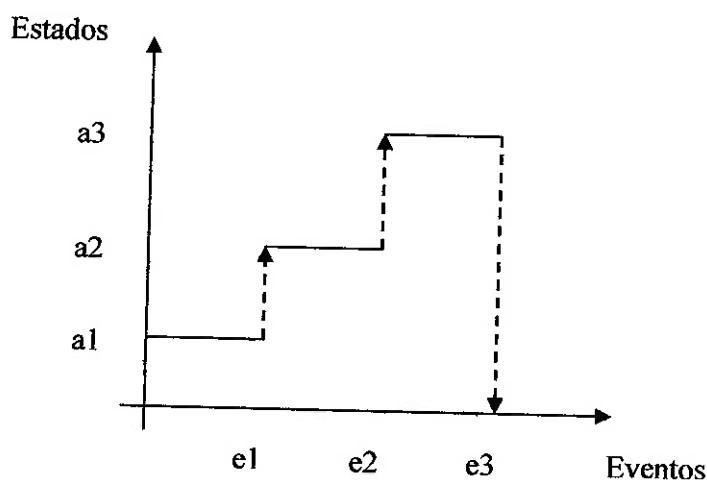


Figura 2 - Evolução do comportamento de um SED a partir da ocorrência de eventos

Comparando esses dois gráficos, os gráficos de SED e SVC, pode-se observar que um SED é limitado a um conjunto de estados discretos, enquanto que em SVC, o espaço de estados é contínuo. Nos SEDs, pode haver a permanência em um mesmo estado durante um intervalo arbitrário de tempo, sendo que a sua trajetória pode ser caracterizada somente por uma seqüência de eventos.

Uma vez que os objetos de estudo de SEDs, são em geral sistemas de alto custo e grande complexidade devido à quantidade e necessidade de integração de diversos componentes (hardware, software, seres humanos), é fundamental o emprego de metodologias de projeto e controle que assegurem o comportamento do sistema de acordo com as especificações lógicas e restrições físico-operacionais [Gustin, 1999].

Existem várias técnicas para a modelagem de SEDs, baseadas nas propriedades que estes sistemas apresentam (reinicialização, sincronização, concorrência, etc.), porém,

nenhuma delas é capaz de modelar todos os sistemas existentes, sendo restritas a apenas alguns. Entre essas técnicas podemos citar por exemplo:

- Cadeias de Markov
- Teoria de Filas
- Processos Semi-Markovianos Generalizados (GSPM)
- Modelos auxiliados por computador
- Redes de Petri

Segundo [Ho, 1989], essas técnicas podem ser classificadas em:

- Modelos temporizados e não temporizados conforme consideram ou não os instantes de tempo em que ocorrem os eventos.
- Modelos lógicos, algébricos, e de análise de desempenho, caso os modelos sejam orientados por lógicas, operações algébricas ou formulação probabilística.

No presente trabalho, o sistema objeto de estudo será tratado utilizando a técnicas de modelo auxiliado por computador, por meio de simulações computacionais.

Considerando os pontos explicitados, é possível criar modelos de cruzamento em SED uma vez que os semáforos podem ser considerados variáveis de estado que mudam em instantes determinados alterados pela ocorrência de algum evento e em conformidade com o estado anterior. Não só os semáforos possuem estados bem definidos (aberto ou fechado), mas também os veículos que podem ter dois estados (em movimento ou parado) e as vias que podem estar ocupadas ou livres [Itano, 1997].

Assim, baseado no conceito de SED, parte do objetivo deste trabalho é estruturar uma estratégia de controle que permita reduzir o tempo de percurso de um veículo emergencial em uma rota pré-determinada.

2.2. MALHA VIÁRIA URBANA

Uma malha viária urbana, composta principalmente de vias e intersecções (cruzamentos), é destinada ao trânsito de veículos e pedestres, sendo projetada de forma a assegurar a cada corrente de tráfego a sua parcela de espaço. Isto, além da organização do trânsito, visa promover a segurança de todos os seus usuários. Existem pontos, porém, onde o traçado da malha viária promove a necessidade de utilização simultânea de uma área por correntes de tráfego conflitantes. Esses pontos, as intersecções, fazem necessária a utilização de formas diferenciadas de gerenciamento da ocupação do espaço viário pelas diferentes correntes, a fim de assegurar a fluidez e segurança do tráfego [Schreiner, 2001].

Para que a circulação em uma malha viária seja ordenada e segura, deve existir um sistema de gerenciamento de tráfego em cada intersecção da malha viária em função da severidade dos conflitos nela existentes. Essa severidade é avaliada geralmente em função de características físicas e do tráfego no local, tais como: o volume de tráfego envolvido (motorizado e não-motorizado), o número de correntes conflitantes e a geometria da interseção, entre outros. Em função destas características deve-se escolher a forma de gerenciamento empregada, que pode ser: a utilização das regras de circulação, sem sinalização específica, previstas no Código de Trânsito Brasileiro – CTB, a utilização de sinalização vertical e/ou horizontal específica, rotatória, ou ainda a utilização de sinais semafóricos. Porém, nem todas podem ser aplicadas em determinados casos. Uma mini-rotatória implica na limitação de fluxo, o que a torna inadequada para aplicações em casos de fluxo intenso. Nesses casos, a utilização de um sinal semafórico é o mais indicado.

O aumento significativo do volume de tráfego nas cidades brasileiras e as exigências da legislação atual tornaram o gerenciamento adequado do tráfego nas intersecções um grande desafio para os órgãos gestores de trânsito. A carença de técnicos capacitados e de ferramentas adequadas para a análise de tais problemas são as principais dificuldades enfrentadas por estes órgãos.

Entre as técnicas citadas de gerenciamento, este trabalho focará nas técnicas de gerenciamento em cruzamentos, mais especificamente em semáforos.

O semáforo é um dispositivo de controle de tráfego que alterna o direito de passagem de veículos e pedestres em cruzamentos, mediante a utilização de indicadores

luminosos. Este sistema de controle organiza de forma cíclica e seqüencial a passagem de veículos e pedestres em uma interseção. Existem dois tipos de semáforos, os veiculares e os de pedestres, diferindo apenas nas indicações luminosas, e portanto nas mensagens que transmitem.

O controle do tráfego com a utilização de semáforos permite a adoção de políticas de controle diferenciadas para um conjunto de interseções. Elas podem ser controladas de forma isolada quando a distância entre elas permite que o funcionamento de um semáforo não interfira na operação dos semáforos adjacentes. Caso contrário, os semáforos são projetados para um funcionamento em conjunto, denominado controle em rede. Neste caso, os semáforos têm seu funcionamento interligado visando otimizar a operação da rede como um todo.

Os tempos semafóricos podem ser fixos ou variáveis, os quais podem, ainda, variar parcial ou completamente em função do tráfego. Neste último caso, são denominados semi-atuados e totalmente atuados, respectivamente. Os semáforos com atuação variável são conhecidos também como semáforos inteligentes [Szasz, 1992]. Na maioria das cidades brasileiras utiliza-se o controle por tempos fixos, tanto na operação isolada quanto em rede.

Os critérios para a seleção de semáforos como forma de controle de uma interseção são definidos pelo Manual de Semáforos do DENATRAN. Esses critérios baseiam-se no volume do tráfego nas aproximações, nas brechas oferecidas pela via principal, no número de aproximações, no volume de pedestres, nos índices de acidentes, na progressão dos veículos através de várias interseções semafORIZADAS, na existência de áreas congestionadas, e em combinações destes critérios.

Segundo estudos realizados por *Services d'Études Techniques des Routes et Autorouts* (SETRA) em Paris, comparando-se os cruzamentos com semáforos convencionais com os semáforos inteligentes, verificou-se que neste último caso houve um ganho no tempo para a travessia do cruzamento e redução no número de paradas nos semáforos, possibilitando assim uma melhoria no fluxo de veículos. Além da melhora do tráfego, pode-se notar uma sensível redução de acidentes de trânsito segundo dados da própria Companhia de Engenharia de Tráfego, supostamente por causa do fato de que o motorista tem um tempo de espera menor, evitando assim cruzar a via em sinal vermelho [Szasz, 1992].

Existem outros benefícios da aplicação de semáforos com atuação variável, como por exemplo, a priorização de algumas rotas, liberando o tráfego para veículos emergenciais como ambulâncias e viaturas policiais, possibilitando assim um atendimento mais eficiente à população.

3. SISTEMAS DE CONTROLE INTELIGENTE

Há várias propriedades essenciais presentes em diferentes níveis de sistemas de controle inteligente. Abaixo são apresentadas três características que parecem estar mais bem fundamentadas em sistemas de controle inteligente. [GSI,2006]

A capacidade de se adaptar ou mudar de condições é necessária em um sistema inteligente. Embora adaptação não requer necessariamente capacidade para aprender, para sistemas, ser capaz de se adaptar a uma variedade vasta de aprendizagem de mudanças inesperadas é essencial. Assim a capacidade para aprender é uma característica importante de sistemas de controle inteligente.

Autonomia é uma característica importante em um sistema de controle. Quando um sistema tem a capacidade para agir apropriadamente em um ambiente incerto por extensos períodos de tempo, sem intervenção externa, ele é considerado altamente autônomo. Existem níveis de autonomia; um nível adaptativo de sistemas de controle pode ser considerado como um sistema de mais alta autonomia que um sistema de controle com controladores fixos. Embora para autonomia baixa nenhuma inteligência (ou pouca inteligência) é necessária, para altos graus de autonomia, inteligência no sistema (ou altos níveis de inteligência) é essencial.

Para evitar uma complexidade desnecessária, um sistema inteligente deve ter uma arquitetura funcional apropriada ou uma estrutura para análise eficiente e evolutiva de estratégias de controle. Esta estrutura deveria fornecer um mecanismo para construir níveis de abstração ou ao menos algum formulário parcial com a finalidade de reduzir a complexidade. Uma abordagem para estudar máquinas inteligentes envolvendo entropia enfatiza tais eficientes estruturas computacionais. Para lidar com a mudança de circunstâncias, a capacidade para aprender é essencial; assim essas estruturas devem ter significativa capacidade de adaptação.

Note que a estrutura matemática pode não ser abrangente o suficiente em certos casos. De fato já é sabido que existem problemas de controles que não podem ser descritos adequadamente através de equações diferenciais. Exemplos incluem representações de

eventos discretos e sistemas de comunicação, cujo estudo levou ao uso de autômatos e teorias de filas no controle de sistemas.

Em algumas vertentes, controle inteligente vem a significar uma forma de controle usando fuzzy e/ou métodos de redes neurais [GSI,2006]. Entretanto Controle não se restringe somente a essas metodologias. Na verdade, de acordo com algumas definições de controle, nem todos os controlares neural/fuzzy poderiam ser considerados inteligentes. A verdade é que existem problemas de controle que não podem ser formulados e estudados na estrutura de equações matemáticas diferenciais. Para dispor estes problemas de modo sistemático, vários métodos foram desenvolvidos e são comumente conhecidos como métodos de controle inteligente.

Existem diferenças significativas entre controle convencional e inteligente. Vale lembrar que controle inteligente usa métodos de controle convencional para resolver problemas de controle de menor nível e este controle convencional está incluído na área de controle inteligente. O Controle Inteligente tenta constituir e acentuar as metodologias de controle convencional para resolver novos desafios de problemas de controle.

A palavra controle em controle inteligente tem um significado mais geral que em controle convencional. De fato, isto está próximo do modo como o termo controle é usado na linguagem do dia a dia. Porque controle inteligente traz mais problemas de controle geral; ele também inclui os problemas trazidos por controle convencional, é menos difícil trazer exemplos significativos. Controle inteligente pode listar problemas de controle que não podem ser formulados na linguagem de controle convencional. Para ilustrar, em um moinho com roda de aço, por exemplo, enquanto controladores convencionais podem incluir reguladores de velocidade (rpm) das rodas de aço, na estrutura de controle inteligente uma poderia incluir, por exemplo, diagnósticos de falha e sistemas de alarme, e talvez o problema de decidir entre os vários estados dos reguladores, baseado na seqüência de ordens processadas, seleções baseadas em decisões econômicas, manutenção de agenda, disponibilidade de máquinas, etc. Todos estes fatores têm de ser considerados pois eles fazem diferença no controle de todo o processo de produção que é o ponto mais importante.

Outra diferença entre os controles inteligente e convencional está na separação entre o controlador e o sistema a ser controlado. No controle convencional, o sistema a ser controlado, chamado de planta, geralmente é separado e diferente do controlador. O controlador é desenhado pelo *control designer*, enquanto a planta é em geral dada, e não pode ser alterada. Note que tentativas recentes de coordenar o design do sistema e controle foram relatadas em áreas como estruturas espaciais e processos químicos, como a algum tempo certas mudanças do design levaram a sistemas que são mais simples que o controle.

Áreas de pesquisa relativas ao controle inteligente, além de controle convencional incluem áreas como planejamento, aprendizado, algoritmos de busca, sistemas híbridos, diagnósticos de falha e reconfiguração, autômatos, redes de Petri, redes neurais e lógica fuzzy. Além disso, para controlar sistemas complexos, o controlador tem que lidar efetivamente com a questão da complexidade computacional, isto tem estado na periferia dos interesses dos pesquisadores em controle convencional, mas agora está claro que a complexidade computacional é uma questão central, sempre uma tentativa de controlar sistemas complexos.

A implementação de sistemas de controle inteligente exige técnicas avançadas de programação, que simulam aspectos do raciocínio humano e/ou são adaptações de processos da natureza para o mundo da informática.

As principais técnicas utilizadas para esse controle são Redes Neurais, Algoritmos Genéticos e Lógica Difusa.

3.1. REDES NEURAIS

A partir do momento em que as máquinas começaram a evoluir, um grande desejo do homem tem sido a criação de uma máquina que possa operar independentemente do controle humano. Uma máquina cuja independência seja desenvolvida de acordo com seu próprio aprendizado e que tenha a capacidade de interagir com ambientes incertos e desconhecidos por ela, uma máquina que possa ser chamada de autônoma, inteligente ou cognitiva.

O sucesso de uma máquina autônoma dependeria única e exclusivamente de sua capacidade de lidar com uma variedade de eventos inesperados no ambiente em que opera. Estas máquinas teriam maior capacidade de aprender tarefas de alto nível cognitivo que não são facilmente manipuladas por máquinas atuais, e continuariam a se adaptar e realizar tais tarefas gradativamente com maior eficiência, mesmo que em condições de ambiente imprevisíveis. Então, seriam muito úteis onde a iteração humana é perigosa, tediosa ou impossível; como em reatores nucleares, combate ao fogo, operações militares, exploração do espaço a distâncias em que um a nave espacial estaria fora do alcance do controle na terra porém enviando informações.

Organismos humanos são uma fonte de motivação para o desenvolvimento destas máquinas, e proporcionam diversas dicas para o desenvolvimento de algoritmos de aprendizado e adaptação. Assim, espera-se que algumas das características de organismos biológicos de aprendizado e adaptação estejam presentes nas mesmas.

Enquanto computadores funcionam de modo seqüencial, proporcionando maior eficiência na resolução de tarefas nas quais devem ser seguidas etapas. O cérebro humano funciona de modo paralelo, e sendo extremamente conectado é mais eficiente na resolução de tarefas que exigem várias variáveis.

Redes Neurais Artificiais baseiam-se na visão da Inteligência Artificial Conexionista, onde acredita-se que construindo um sistema que simule a estrutura cerebral humana, este apresentará inteligência e será capaz de aprender, assimilar, errar e, com esta experiência, adquirir novos conhecimentos.

Desta forma, as Redes Neurais tendem a representar uma estrutura interconectada similar a estrutura do cérebro humano e com isto resolver situações não facilmente resolvidas na computação convencional, onde a informática é transferida linear e sincronizadamente.

Assim as Redes Neurais tornam-se de fundamental importância no desenvolvimento dos sistemas de Inteligência Artificial onde há necessidade de resolução de problemas não

lineares, bem como no reconhecimento de padrões na otimização e na previsão de sistemas complexos.

De acordo com diversas estruturas neurais e algoritmos de aprendizagem propostos por vários pesquisadores, redes neurais possuem certas características exclusivas de sistemas biológicos. Tais características entram em conflito com os tradicionais métodos computacionais. Sistema de computação baseado em redes neurais tem a capacidade de receber ao mesmo tempo várias entradas e distribuí-las de maneira organizada. Geralmente, as informações armazenadas por uma rede neural são compartilhadas por todas as suas unidades de processamento, característica que contrasta com os atuais esquemas de memória, onde a informação fica confinada em um determinado endereço.

Em um sistema de rede neural, a informação pode parecer ter representação redundante, porém, o fato de que ela se encontre distribuída por todos os elementos da rede significa que mesmo que parte da rede seja destruída, a informação contida nesta parte ainda estará presente na rede, e poderá ser recuperada. Portanto, a redundância na representação de informações em uma rede neural, diferente de outros sistemas, transforma-se em uma vantagem, que torna o sistema tolerante a falhas. Os atributos de uma rede neural, tais como aprender através de exemplos, generalizações redundantes, e tolerância a falhas, proporcionam fortes incentivos para a escolha de redes neurais como uma escolha apropriada para aproximação para a modelagem de sistemas biológicos. Todo o potencial de uma rede neural pode ser enumerado nos parágrafos seguintes.

O modelo de rede neural tem muitos neurônios conectados por pesos com capacidade de adaptação que podem ser arranjados em uma estrutura paralela. Por causa deste paralelismo, a falha de alguns neurônios não causa efeitos significantes para a performance de todo o sistema, o que é chamado de tolerância à falhas.

A principal força na estrutura de redes neurais reside em suas habilidades de adaptação e aprendizagem. A habilidade de adaptação e aprendizagem pelo ambiente significa que modelos de redes neurais podem lidar com dados imprecisos e situações não totalmente definidas. Uma rede treinada de maneira razoável tem a habilidade de

generalizar quando é apresentada a entradas que não estão presentes em dados já conhecidos por ela.

A característica mais significante de redes neurais está em sua habilidade de aproximar qualquer função continua não linear de um grau de correção desejado. Esta habilidade das redes neurais as têm tornado útil para modelar sistemas não lineares na combinação de controladores não lineares.

Redes Neurais podem ter várias entradas e várias saídas, eles são facilmente aplicáveis a sistemas com muitas variáveis.

Com o avanço em tecnologias de hardware, existem componentes com funções voltadas a sistemas com implementações voltadas para redes neurais, o que traz uma velocidade adicional à computação neural.

As primeiras informações mencionadas sobre a neuro computação datam de 1943, em artigos de McCulloch e Pitts, em que sugeriam a construção de uma máquina baseada ou inspirada no cérebro humano. Muitos outros artigos e livros surgiram desde então, porém, por um longo período de tempo, pouco resultado foi obtido. Até que em 1949 Donald Hebb escreveu um livro intitulado "The Organization of Behavior" (A Organização do Comportamento) que perseguia a idéia de que o condicionamento psicológico clássico está presente em qualquer parte dos animais pelo fato de que esta é uma propriedade de neurônios individuais. Suas idéias não eram completamente novas, mas Hebb foi o primeiro a propor uma lei de aprendizagem específica para as sinapses dos neurônios. Este primeiro e corajoso passo serviram de inspiração para que muitos outros pesquisadores pergessem a mesma idéia. Embora muito tenha sido estudado e publicado nos anos que seguiram (1940-1950), estes serviram mais como base para desenvolvimento posterior que para o próprio desenvolvimento.

Nos anos 80, muitos dos pesquisadores foram bastante corajosos e passaram a publicar diversas propostas para a exploração de desenvolvimento de redes neurais bem como suas aplicações. Porém talvez o fato mais importante deste período tenha ocorrido quando Ira Skurnick, um administrador de programas da DARPA (Defense Advanced

Research Projects Agency), decidiu ouvir os argumentos da neuro computação e seus projetistas, e divergindo dos caminhos tradicionais dos conhecimentos convencionais, fundou, em 1983, pesquisas em neuro computação. Este ato não só abriu as portas para a neuro computação, como também deu à DARPA o status de uma das líderes mundiais em se tratando de "moda" tecnológica.

Apesar de um terço dos pesquisadores da área terem aderido à mesma pela influência de Hopfield, foi em 1986 que este campo de pesquisa "explodiu" com a publicação do livro "Parallel Distributed Processing" (Processamento Distribuído Paralelo) editado por David Rumelhart e James McClelland.

Em 1987 ocorreu em São Francisco à primeira conferência de redes neurais em tempos modernos, a IEEE International Conference on Neural Networks, e também foi formada a International Neural Networks Society (INNS). A partir destes acontecimentos decorreram a fundação do INNS journal em 1989, seguido do Neural Computation e do IEEE Transactions on Neural Networks em 1990.

Aplicações interessantes de redes neurais são encontradas nas áreas de finanças, recursos humanos ou marketing. Esta nova tecnologia de informação se baseia no funcionamento do cérebro humano. Ela vem propor um uso diferenciado do enorme volume de dados estocados nos computadores de empresas que muitas vezes são pouco ou mal utilizados, transformando-os em informação útil à tomada de decisão. Mas isto não é feito através da proposição de telas sofisticadas de consulta como os EIS e nem através dos sistemas passivos de análise de cenários como os Sistemas de Apoio à Decisão tradicionais. Sistemas baseados em redes neurais propõem uma ajuda ativa à tomada de decisão.

As redes neurais tem sido utilizadas na área de administração para criação de bases de conhecimento. Uma das aplicações importantes é a avaliação de risco de crédito. As redes são muitas vezes comparadas neste tipo de aplicação com métodos estatísticos, em análise de dados. As redes neurais tem duas grandes vantagens em relação às técnicas estatísticas tradicionais:

Em primeiro lugar, elas tratam não só dados numéricos, mas também dados qualitativos. Por exemplo, na avaliação de uma empresa, a rede pode levar em consideração até mesmo informações como o estilo de gestão efetuado pela empresa.

Em segundo lugar, quanto maior o volume de dados melhor será seu aprendizado, e melhor será a qualidade das previsões. Ela consegue encontrar informações e relações entre os dados que não são possíveis de serem encontradas com técnicas estatísticas convencionais.

Aplicações de redes neurais são inúmeras. Muitos recebem sua primeira introdução lendo a respeito das técnicas no prognóstico de mercados financeiros. Grupos de investimento conhecidos utilizam redes neurais para analisar pelo menos uma parte do mercado financeiro e fazerem suas seleções.

O reconhecimento ótico de caracteres (OCR) é outro tipo de aplicação que já existe e está crescendo, e em breve estaremos em constante contato com esse tipo de aplicação. Outras aplicações bem sucedidas das técnicas de redes neurais artificiais são: análise de pesquisa de mercado, como acima citado, controle de processos industriais, aplicações climáticas, e identificação de fraude de cartão de crédito. Um banco americano chamado Mellon Bank instalou um sistema de detecção de fraudes de cartão de crédito implementado com técnicas de redes neurais e os prejuízos evitados pelo novo sistema conseguiram cobrir os gastos de instalação em seis meses. Vários outros bancos começam a utilizar sistemas baseados em redes neurais para controlar fraudes de cartão de crédito. Estes sistemas têm a capacidade de reconhecer uso fraudulento com base nos padrões criados no passado com uma precisão melhor que em outros sistemas.

Outro exemplo da utilização de redes neurais para melhoria na tomada de decisões é no diagnóstico médico. Em seu aprendizado, são submetidos uma série de diagnósticos de pacientes, de várias características, com vários sintomas e os resultados de seus testes. Também serão fornecidos os diagnósticos médicos para cada doença. Então quando forem apresentados os dados de um novo paciente, com seus sintomas, a rede fornecerá um diagnóstico para os novos casos. Isto essencialmente criará um sistema com o conhecimento de vários médicos, e fornecerá um diagnóstico inicial em tempo real à um

médico. É importante mencionar que com isso o que se pretende é implementar uma ferramenta de auxílio ao médico, e não um programa que o substitua.

Outras aplicações são a análise e processamento de sinais, controle de processos, robótica, classificação de dados, reconhecimento de padrões em linhas de montagem, filtros contra ruídos eletrônicos, análise de imagens, análise de voz, avaliação de crédito, análise de aroma e odor- um projeto que está em desenvolvimento, buscando a análise de odor via nariz eletrônico, análise e diagnóstico de descargas parciais pelo reconhecimento do padrão acústico- trata-se de uma tese de mestrado cujo objetivo é criar um sistema com capacidades de classificar o padrão acústico de uma descarga parcial ;

3.2. ALGORITMOS GENÉTICOS

Muitos problemas podem ser resolvidos de uma forma computacional determinística, contudo, outros problemas não têm um método de resolução exato, ou os métodos para a obtenção de tal solução são complexos em demasia em sua implementação e utilização. Para a resolução deste, utiliza-se métodos heurísticos e até metaheurísticos, como os Algoritmos Genéticos.

Os Algoritmos Genéticos fazem parte da área de IA de Sistemas Inspirados na Natureza; sistemas estes que simulam os processos naturais e aplica-os à solução de problemas reais. Algoritmos Genéticos são métodos generalizados de busca e otimização que simulam os processos naturais da evolução, aplicando a idéia darwiniana de seleção natural das espécies. De acordo com a aptidão e utilizando-se da combinação com outros operadores genéticos, são produzidos métodos de grande robustez e aplicabilidade na resolução de problemas.

Estes algoritmos estão baseados nos processos genéticos dos organismos biológicos, codificando uma possível solução a um problema de "cromossomo" composto por cadeia de bits e caracteres. Estes cromossomos representam indivíduos que são levados ao longo de várias gerações, na forma similar aos problemas naturais, evoluindo de acordo com os princípios de seleção natural e sobrevivência dos mais aptos, descritos pela primeira vez

por Charles Darwin em seu livro "A Origem das Espécies". Emulando estes processos, os Algoritmos Genéticos são capazes de evoluir soluções de problemas do mundo real.

Na natureza os indivíduos competem entre si por recursos como comida, água e refúgio. Adicionalmente, entre os animais de uma mesma espécie, aqueles que não obtêm êxito tendem a ter um número reduzido de descendentes, tendo portanto menor probabilidade de seus genes serem propagados ao longo de sucessivas gerações. A combinação entre os genes dos indivíduos que perduram na espécie, pode produzir um novo indivíduo muito melhor adaptado às características de seu meio ambiente.

Os Algoritmos Genéticos utilizam uma analogia direta deste fenômeno de evolução na natureza, onde cada indivíduo representa uma possível solução para um problema dado. A cada indivíduo se atribui uma pontuação de adaptação, dependendo da resposta dada ao problema por este indivíduo. Aos mais adaptados é dada a oportunidade de reproduzir-se mediante cruzamentos com outros indivíduos da população, produzindo descendentes com características de ambas as partes. Se um Algoritmo Genético for desenvolvido corretamente, a população, ou seja, conjunto de possíveis respostas, convergirá a uma solução boa para o problema proposto, talvez até ótima. Os processos que mais contribuem para a evolução são o crossover e a adaptação baseada na seleção/reprodução. A mutação também tem um papel significativo, no entanto, seu grau de importância continua sendo um assunto em que os pesquisadores ainda não entraram em consenso.

Um Algoritmo Genético pode convergir em uma busca de azar, porém sua utilização assegura que nenhum ponto do espaço de busca tem probabilidade zero de ser examinado. Toda tarefa de busca e otimização possui vários componentes, entre eles: O espaço de busca onde são consideradas todas as possibilidades de solução de um determinado problema e a função de avaliação (ou função de custo), uma maneira de avaliar os membros do espaço de busca.

As técnicas de busca e otimização tradicionais iniciam-se com um único candidato que, iterativamente, é manipulado utilizando algumas heurísticas diretamente associadas ao problema a ser solucionado. Por outro lado, as técnicas de computação evolucionária operam sobre uma população de candidatos em paralelo. Assim, elas podem fazer a busca

em diferentes áreas do espaço de solução, alocando um número de membros apropriado para a busca em várias regiões. Os Algoritmos Genéticos (AGs) diferem dos métodos tradicionais de busca e otimização, principalmente em quatro aspectos: AGs trabalham com uma codificação do conjunto de parâmetros e não com os próprios parâmetros, AGs trabalham com uma população e não com um único ponto, AGs utilizam informações de custo ou recompensa e não derivadas ou outro conhecimento auxiliar, AGs utilizam regras de transição probabilísticas e não determinística.

Algoritmos Genéticos são muito eficientes para busca de soluções ótimas, ou soluções boas, em uma grande variedade de problemas, pois não impõem muitas das limitações encontradas nos métodos de busca tradicionais. Os pesquisadores referem-se a "algoritmos genéticos" ou a "um algoritmo genético" e não "ao algoritmo genético", pois AGs são uma classe de procedimentos com muitos passos separados, e cada um destes passos possui muitas variações possíveis.

Sistemas adaptativos tentam resolver problemas acumulando conhecimento sobre o problema em questão e utilizando estas informações para gerar soluções aceitáveis. Estes problemas, tipicamente, se encontram nas áreas de configuração de sistemas complexos, alocação de tarefas, seleção de rotas, e outros problemas de otimização e aprendizado de máquina.

Seguem-se alguns exemplos de sistemas adaptativos: Controle de Sistemas Dinâmicos, Indução e Otimização de Bases de Regras, Encontrar Novas Topologias Conexionistas: Engenharia de Sistemas Neurais Artificiais, Modelagem de Estruturas Neurais Biológicas, Simulação de Modelos Biológicos, Evolução Interativa de Imagens, Composição Musical.

3.3. LÓGICA DIFUSA

Os conjuntos Fuzzy constituem uma "ponte" no caminho de aproximar o raciocínio humano ao da lógica executada pela máquina.

Aristóteles, filósofo grego (384 - 322 a.C.), foi o fundador da ciência da lógica, e estabeleceu um conjunto de regras rígidas para que conclusões pudessem ser aceitas logicamente válidas. O emprego da lógica de Aristóteles levava a uma linha de raciocínio lógico baseado em premissas e conclusões. Como por exemplo: "Todo ser vivo é mortal" (premissa 1), a seguir é constatado "João é um ser vivo" (premissa 2), como conclusão temos que "João é mortal". Desde então, a lógica Ocidental, assim chamada, tem sido binária, isto é, uma declaração é falsa ou verdadeira, não podendo ser ao mesmo tempo parcialmente verdadeira e parcialmente falsa. Esta suposição e a lei da não contradição, que coloca que "U e não U" cobrem todas as possibilidades, formam a base do pensamento lógico.

A Lógica Difusa (Fuzzy Logic) viola estas suposições. O conceito de dualidade, estabelecendo que algo pode e deve coexistir com o seu oposto, faz a lógica difusa parecer natural, até mesmo inevitável.

A Lógica Difusa, com base na teoria dos Conjuntos Fuzzy, tem se mostrado mais adequada para tratar imperfeições da informação do que a teoria das probabilidades. Fuzzy, em inglês, significa incerto, duvidoso. De forma mais objetiva e preliminar, podemos definir Lógica Difusa como sendo uma ferramenta capaz de capturar informações vagas, em geral descritas em uma linguagem natural e convertê-las para um formato numérico, de fácil manipulação pelos computadores de hoje em dia.

A Lógica Difusa tem encontrado grandes aplicações nas seguintes áreas: Sistemas Especialistas, Computação com Palavras, Raciocínio Aproximado, Linguagem Natural, Controle de Processos, Robótica, Modelamento de Sistemas Parcialmente Abertos, Reconhecimento de Padrões, Processos de Tomada de Decisão (decision making)

A Lógica Difusa ou Lógica Nebulosa, também pode ser definida, como a lógica que suporta os modos de raciocínio que são aproximados, ao invés de exatos, como estamos naturalmente acostumados a trabalhar. Ela está baseada na teoria dos conjuntos fuzzy e difere dos sistemas lógicos tradicionais em suas características e detalhes.

Nesta lógica, o raciocínio exato corresponde a um caso limite do raciocínio aproximado, sendo interpretado como um processo de composição difusa.

A lógica em questão foi desenvolvida por Lofti A. Zadeh da Universidade da Califórnia em Berkeley na década de 60 e combina lógica multivalorada, teoria probabilística, Inteligência Artificial e redes neurais para que possa representar o pensamento humano, ou seja, ligar a lingüística e a inteligência humana, pois muitos conceitos são melhores definidos por palavras do que pela matemática.

As Vantagens apresentadas pela lógica difusa são: Requer poucas regras, valores e decisões, Mais variáveis observáveis podem ser valoradas, O uso de variáveis lingüísticas nos deixa mais perto do pensamento humano. Simplifica a solução de problemas, Proporciona um rápido protótipo dos sistemas, Simplifica a aquisição da base do conhecimento,

Inicialmente, sistemas fuzzy foram amplamente ignorados nos Estados Unidos porque foram associados com Inteligência Artificial, um campo que periodicamente se obscurecia, resultando numa falta de credibilidade por parte da indústria.

Os japoneses não pensaram assim.O interesse em sistemas fuzzy foi demonstrado por Seiji Yasunobu e Soji Miyamoto da Hitachi, que em 1985 apresentou simulações que demonstraram a superioridade de sistemas de controle fuzzy para a estrada de ferro de Sendai.Suas idéias foram adotadas e sistemas fuzzy foram usados para controle de aceleração, frenagem, e parada quando a linha foi inaugurada em 1987.

Outro evento em 1987 ajudou a promover o interesse em sistemas fuzzy: durante um encontro internacional de pesquisadores de fuzzy em Tókio naquele ano, Takeshi Yamakawa demonstrou o uso de controle fuzzy (através de um conjuntos de simples chips fuzzy dedicados) em um experimento de um pêndulo invertido - um problema clássico de controle em que um veículo tenta manter um poste montado no seu topo por uma dobradiça vertical com movimentos de ida e volta.

O entusiasmo dos japoneses por lógica fuzzy é refletido na ampla faixa de outras aplicações que eles tem investigado ou implementado: reconhecimento de caracteres, sistemas fuzzy óticos, robôs, helicópteros comandados por voz através de um robô, sistemas de elevadores, entre outras.

Pesquisa e desenvolvimento estão em andamento em aplicações fuzzy em projeto de software, incluindo sistemas fuzzy expert e integração de lógica fuzzy com redes neurais os denominados algoritmos genéticos adaptativos, com o objetivo de construção de um sistema fuzzy capaz de aprender. Controle Fuzzy

A Lógica Difusa pode ser utilizada para a implementação de controladores fuzzy, aplicados nos mais variados tipos de processos. A utilização de regras fuzzy e variáveis lingüísticas confere ao sistema de controle várias vantagens, incluindo: simplificação do modelo do processo, melhor tratamento das imprecisões inerentes aos sensores utilizados, facilidade na especificação das regras de controle, em linguagem próxima à natural, satisfação de múltiplos objetivos de controle, facilidade de incorporação do conhecimento de especialistas humanos.

Controladores fuzzy são muito simples conceitualmente. Eles consistem de um estágio de entrada, um estágio de processamento, e um estágio de saída.

O estágio de entrada mapeia sensores ou outros tipos de entrada, tais como chaves, interruptores e assim por diante, de maneira apropriada as funções consecutivas e valores verdadeiros. O estágio de processamento invoca cada regra adequada e gera um resultado para cada uma delas e então combinam os resultados dessas regras; e finalmente o estágio de saída converte o resultado combinado no estágio anterior para dentro do controle.

O estágio de processamento é baseado em uma coleção de regras lógicas na forma de declarações IF-THEN, onde o IF é chamado de antecedente e o THEN é chamado de consequência. Sistemas de controle fuzzy típicos têm dúzias de regras. Considere uma regra para um termostato:

Regras podem ser resolvidas em hardware paralelo ou em software seqüencial. A resultada de todas as regras é determinada para um valor conciso por um dos vários métodos existentes na teoria, cada qual com suas vantagens e desvantagens.

O projeto de sistemas de controle fuzzy é baseado em métodos empíricos, basicamente uma aproximação metódica por tentativa – e – erro. Existem poucas regras pré-definidas no presente.

3.4. SISTEMA ESPECIALISTA

Um Sistema Especialista é aquele projetado e desenvolvido para atender a uma aplicação determinada e limitada do conhecimento humano. É capaz de emitir uma decisão apoiada em conhecimento justificado, a partir de uma base de informações, da mesma forma que um especialista de determinada área do conhecimento humano.

Para tomar uma decisão sobre um determinado assunto, baseado nos fatos que encontra, um sistema especialista formula suas hipóteses. Durante o processo de raciocínio, ele verifica qual a importância dos fatos que encontra, comparando-os com as informações já adquiridas sobre esses fatos e hipóteses. Neste processo, vai formulando novas hipóteses e verificando novos fatos; e esses novos fatos vão influenciar no processo de raciocínio. Este raciocínio é sempre baseado no conhecimento prévio acumulado.

Um sistema especialista com esse processo de raciocínio pode não chegar a uma decisão se os fatos de que dispõe para aplicar o seu conhecimento prévio não forem suficientes. Pode, inclusive, chegar a uma conclusão errada. Porém, este erro é justificado em função dos fatos que encontrou e do seu conhecimento acumulado previamente.

Um Sistema Especialista deve, além de inferir conclusões, ter capacidade de aprender e, desse modo, melhorar o seu desempenho de raciocínio e a qualidade de suas decisões.

De um modo geral, sempre que um problema não pode ser algoritmizado, ou sua solução conduz a um processamento muito demorado, os Sistemas Especialistas podem ser uma saída, pois possuem o seu mecanismo apoiado em processos heurísticos. Além disso, pode ser uma boa maneira de preservar e transmitir o conhecimento de um especialista humano em uma determinada área.

Vale ainda lembrar que um Sistema Especialista não é influenciado por elementos externos a ele, como ocorre com o especialista humano. Dessa forma, para as mesmas condições, deverá fornecer sempre o mesmo conjunto de decisões.

Nos anos 50, os pesquisadores já haviam estabelecido os fundamentos da Inteligência Artificial, incluindo lógica matemática e Teoria das Funções Recursivas, guiando a formulação de processamento de listas e da própria linguagem LISP, que fornece um interpretador para desenvolver expressões simbólicas recursivas. Tais capacidades suportaram o surgimento de sistemas práticos de computação simbólica. Ao mesmo tempo, emergiram computadores interativos tornando possíveis ambientes computacionais para desenvolvimento e depuração de programas incrementais. Aproximadamente nesta mesma ocasião, psicólogos cognitivos - estudantes da forma de pensar humana - criaram caminhos padrão do processo de investigação do raciocínio, modelando o aparente processo de tomada de decisão em termos de regras de produção condicionais.

Nos anos 60, os pesquisadores de Inteligência Artificial tentaram simular o complexo processo do pensamento, procurando métodos gerais para resolver uma ampla classe de problemas. No entanto, a despeito de alguns progressos interessantes, as dificuldades eram enormes e não frutificaram. Então, durante a década de 70, concentraram esforços em técnicas como a Representação , isto é, um modo de formular o problema de maneira a tornar sua solução mais fácil, como controlá-la inteligentemente dentro da capacidade de memória do computador. Esta estratégia produziu algum sucesso mas ainda não foi decisiva. Somente no final da década fizeram a descoberta mais importante: o poder do programa em resolver problemas depende mais do conhecimento que possui do que do formalismo ou esquema de inferência empregado. Esta realização levou ao desenvolvimento de programas de computador de propósito particular, sistemas que são peritos em alguma área limitada. Estes programas são chamados Sistemas Especialistas e um novo campo se iniciou.

No campo de IA aplicada, após uma década de trabalho, emergiram três sub-campos principais: Sistemas Especialistas, Linguagem Natural e Robótica - incluindo visão, fala e locomoção. Os Sistemas Especialistas começaram a surgir comercialmente entre 1980 e 1981.

Assim os Sistemas Especialistas, após terem se desenvolvido por 15 anos como mera curiosidade de IA aplicada em laboratórios de pesquisa, tornaram-se alvo de significativos esforços de desenvolvimento, tanto técnicos como comerciais. Estes sistemas empregam o computador de maneira mais diversa que o processamento de dados convencional, abrindo novas e importantes oportunidades. Recentemente, muitas organizações têm explorado esta tecnologia, ampliando suas pesquisas e começando a adaptar suas atividades para tanto.

Embora Sistemas Especialistas e peritos reais possam em alguns casos desempenhar tarefas idênticas, as características de ambos são criticamente diversas. Mesmo havendo algumas vantagens evidentes dos Sistemas Especialistas eles não poderão substituir os peritos em todas as situações, devido a algumas limitações inerentes.

Um Sistema Especialista pode ser comparado com um programa gravado em videocassete, enquanto um especialista se compararia com um programa ao vivo. Isto é, a utilização e reprodução, que compensa de sobra seu alto custo de implantação, pode levar vários anos. Os especialistas humanos, especialmente os melhores, tem altos salários, enquanto que Sistemas Especialistas tem o custo nominal do computador rodar o programa.

Por outro lado, para demonstrar que os especialistas não podem ser simplesmente substituídos, o conhecimento artificial tem atualmente algumas limitações. Uma delas é falta de criatividade. Um perito pode reorganizar informações e usá-las para sintetizar novos conhecimentos; pode manusear eventos inesperados usando imaginação ou novas abordagens, inclusive raciocínio por analogia de um outro domínio completamente diferente. Sistemas Especialistas trabalham de maneira sem inspiração, rotineiramente.

Para entender quais são as características comuns aos Sistemas Especialistas, basta examinar o que estes fazem: resolvem problemas muito complexos tão bem quanto ou melhor que especialistas humanos, raciocinam heuristicamente, usando o que os peritos consideram efetivamente regras práticas, interagem com usuários humanos utilizando inclusive linguagem natural, manipulam e raciocinam sobre descrições simbólicas, funcionam com dados errados e regras incertas de julgamento, contemplam hipóteses

múltiplas simultaneamente, explicam porque estão fazendo determinada pergunta, justificam suas conclusões.

A maior capacidade de um Sistema Especialista consiste em sua especialidade de alto nível que auxilia na solução de problemas. Este conhecimento especializado pode representar a experiência dos melhores peritos no campo. Sua especialização de alto nível, juntamente com a habilidade de aplicação, torna seu custo competitivo e apto a ganhar espaço no mercado comercial. A flexibilidade do sistema também auxilia aqui: ele pode crescer incrementalmente segundo as necessidades do negócio ou organização. Isto significa que pode se iniciar com um investimento relativamente modesto expandindo-o de acordo com as necessidades.

O corpo de conhecimento do sistema que define a proficiência de um Sistema Especialista pode também oferecer uma capacidade adicional: a memória institucional. Se a base de conhecimento foi desenvolvida através de interação de pessoas chave da organização, isto representa a política corrente do grupo. Esta compilação de conhecimento vem a ser o consenso de opiniões de alto nível de um registro permanente das melhores estratégias utilizadas. Quando pessoas chaves desligam-se da organização, suas experiências permanecerão.

Os Benefícios apresentados pelo sistemas especialistas são: Velocidade na determinação dos problemas, A decisão está fundamentada em uma base de conhecimento, Segurança, Exige pequeno número de pessoas para interagir com o sistema, Estabilidade, Dependência decrescente de pessoal específico, Flexibilidade, Integração de ferramentas, Evita interpretação humana de regras operacionais.

3.4.1. PROCESSO DE APRENDIZAGEM DE UM SISTEMA ESPECIALISTA

A aprendizagem de um Sistema Especialista comum se dá de diversas formas: Análise estatística de dados, Tentativa e erro, Leituras, palestras, Troca de experiências com outras pessoas, etc.

Fundamentalmente, verifica-se que o aprendizado vem do processo de experiência, e de seus resultados experimentais.

A capacidade de aprender, no ser humano, é o resultado de um conjunto de habilidades: capacidade de generalizar, de induzir, de fazer analogias e de receber instrução.

Os Sistemas Especialistas devem ser capazes de aprender e fazer crescer o seu conhecimento básico sobre o assunto. Esta capacidade de aprender recebe o nome técnico de protopeiria. O usual é existir um engenheiro de conhecimento que prepara o conhecimento para ser armazenado de forma apropriada, fornecendo as explicações necessárias dos conceitos utilizados, mas o ideal é que o conhecimento possa ser adquirido diretamente pelo Sistema Especialista.

Uma das formas de aprendizagem dos Sistemas Especialistas é através de textos. Um programa captura palavras chaves em um parágrafo do texto, podendo formatá-lo para um formato especial de armazenamento, para representação desse conhecimento (implementação mais eficiente e confiável com Processador de Linguagem natural).

O aprendizado também pode ser feito a partir de conclusões sobre a massa de informações mantidas pelo Sistema Especialista. Ele mantém um banco de casos resolvidos, isto é, a cada conclusão guarda os fatos que pesaram sobre a decisão e a própria decisão, após ter esta sido criticada por um especialista da área. O aprendizado é feito por comparação de dados por um módulo do Sistema Especialista que coloca a nova regra na Base de conhecimento, à medida que a massa de dados cresce, obedecendo ao formato adequado.

Outra forma de aprendizado se dá pela interação direta com o especialista. Como em uma relação professor-aluno, o computador absorve o conhecimento através de uma interface adequada, por exemplo um editor inteligente.

Fundamentalmente, verifica-se que o aprendizado vem do processo de experiência, e de seus resultados experimentais.

3.5. SISTEMAS DE PRODUÇÃO

Os sistemas de produção são uma boa maneira de descrever as operações que podem ser realizadas quando buscamos uma solução para um problema. Este é o sistema que será utilizado em nosso estudo.

Uma vez que a busca forma o núcleo de muitos processos inteligentes, é útil estruturar os programas de uma forma que facilite a descrição e execução dos processos de busca. Os Sistemas de Produção (SP) proporcionam tais estruturas.

Um Sistema de Produção consiste em: [GSI,2006]

- Um conjunto de regras, cada uma delas consistindo em um lado que determina a aplicabilidade de uma regra e um outro lado que descreve a operação a ser efetuada se a regra for aplicada.
- Uma ou mais bases de conhecimentos ou bases de dados que contenham quaisquer informações apropriadas a uma determinada tarefa. Certas partes da base de dados podem ser permanentes, enquanto que outras, podem pertencer apenas à solução do problema corrente.
- Uma estratégia de controle que especifique a ordem em que as regras serão comparadas com a base de dados e uma maneira de solucionar conflitos que surgirem quando várias regras puderem ser aplicadas ao mesmo tempo.
- Um aplicador de regras.

Podemos classificar os SP como:

- SP monotônico: a aplicação de uma regra NUNCA impede a aplicação posterior de uma outra regra que também pudesse ter sido aplicada quando a primeira regra foi escolhida;

- SP não-monotônico: a aplicação de uma regra PODE impedir a aplicação posterior de uma regra que também, pudesse ter sido aplicada quando a primeira regra foi escolhida;
- SP parcialmente comutativo: se a aplicação de uma determinada sequência de regras transforma um estado x em um estado y , então qualquer permutação permitida dessas regras também transforma o estado x em estado y ;
- SP comutativo: é monotônico e parcialmente comutativo simultaneamente.

Pode-se dizer que para qualquer problema solucionável, existe um número infinito de SP que descrevem maneiras de encontrar soluções. Alguns serão mais naturais ou mais eficientes que outros.

Qualquer problema que possa ser solucionado por qualquer SP, pode ser solucionado por um SP comutativo, mas o SP comutativo pode ter o manejo tão complicado que se torna praticamente inútil.

Os SP parcialmente comutativos monotônicos são importantes, do ponto de vista da implementação, porque podem ser implementados sem que haja necessidade de retrocesso a estágios anteriores, quando se descobre que um caminho errado foi seguido. Embora normalmente seja útil implementar tais sistemas com o recurso do retrocesso, a fim de garantir uma busca sistemática, a base de dados, que na verdade representa o estado do problema, não precisa ser restaurado, o que se traduz em eficiência.

Os SP comutativos são bons para problemas nos quais os estados não mudam e estados novos são criados. Os SP parcialmente comutativos não-monotônicos são úteis para problemas nos quais aconteçam mudanças que possam ser revertidas e nos quais a ordem das operações não seja crucial.

Os SP não-parcialmente comutativos têm menos chances de produzir o mesmo nó muitas vezes durante o processo de busca.

3.6. AQUISIÇÃO DE CONHECIMENTO

A aquisição do conhecimento é a etapa do desenvolvimento que busca obter o conhecimento necessário à construção do sistema, e não acontece de uma só vez, mas distribuída por todo o processo.

As características do especialista terão grande impacto na aquisição do conhecimento. A experiência deste influencia a organização, o armazenamento e a recuperação do conhecimento. Quanto maior for à organização e mais refinado o processo de raciocínio, mais difícil decodificar o conhecimento e expressá-lo de forma básica. O conhecimento do especialista não se encontra organizado de forma que possa ser extraído diretamente para a base de conhecimento. Muitos pesquisadores e engenheiros do conhecimento têm tentado trabalhar na construção de mecanismos que facilitem este processo.

Outros fatores contribuem para que a aquisição do conhecimento seja a parte mais crítica no processo de desenvolvimento de um sistema baseado em conhecimento. São eles: Falta de gerência e organização, Treinamento incompleto de engenheiros do conhecimento, Dificuldade da tradução do conhecimento para uma linguagem compreensiva pelo computador, Necessidade da flexibilidade da base de conhecimento, Tratamento de conflitos e informações incertas.

Vários pontos devem ser considerados no processo de aquisição do conhecimento, independente do método de desenvolvimento utilizado. A primeira delas, e talvez a mais complexa, é a identificação dos especialistas do domínio da organização que irão participar do desenvolvimento. O segundo ponto é a motivação do especialista do domínio, que deve ser reforçada pelo engenheiro do conhecimento. Também devemos considerar a disponibilidade do especialista, que na maioria das vezes é um profissional muito ocupado. Por último, devemos nos preocupar com o que se refere à seleção e ao sequenciamento das técnicas de elicitação do conhecimento.

O engenheiro do conhecimento desempenha um papel fundamental no processo de aquisição do conhecimento, fornecendo meios ao especialista para que ele consiga transmitir a sua maneira de solucionar os problemas.

4. DESENVOLVIMENTO DO SISTEMA DE CONTROLE DE UMA MALHA VIÁRIA CONTROLADA POR SEMÁFOROS

4.1. ESTUDO DE CASO

Para o desenvolvimento de uma estratégia de controle a ser desenvolvida por meio de simulações computacionais, para ser posteriormente implantada em um protótipo, a determinação de uma caso de estudo se mostra a mais indicada.

O objeto do estudo de caso deve ser um sistema real, já existente, que forneça dados consistentes e confiáveis para serem introduzidos na representação do sistema, ou seja, no modelo. Uma vez que os dados utilizados representam dados reais, o funcionamento do modelo deve então fornecer resultados semelhantes aos fornecidos pelo sistema real. Nesse caso, pode-se dizer que o modelo é consistente, sendo uma representação fiel, do sistema, utilizando apenas as características que sejam realmente necessárias.

Para tanto, uma região da avenida Paulista, localizada na cidade de São Paulo, foi escolhida como o estudo de caso para este projeto.

Essa região foi escolhida por não possuir um controle dos tempos dos semáforos em tempo real e por apresentar vários hospitais nas imediações, necessitando da implantação de um sistema que possibilite a priorização de rota de veículos emergenciais.

É necessário observar que essa região corresponde a uma malha viária, uma vez que ela é composta de cinco vias e seis cruzamentos, todos controlados por semáforos. As vias que compõem essa região são as ruas Bela Cintra, Haddock Lobo e Augusta, a alameda Santos e a avenida Paulista.

4.2. DESCRIÇÃO DA MALHA VIÁRIA

Seguindo as definições de malha viária e utilizando a abordagem por meio de SED, discutidas nos tópicos anteriores, pode-se definir um sistema a ser analisado.

O sistema em estudo é composto por seis cruzamentos todos controlados por semáforos independentes, isto é, todos os semáforos funcionam de forma independente e fixa, ou seja, sem qualquer alteração em seus tempos. Existe uma avenida principal, de grande circulação, com veículos trafegando em ambos os sentidos, paralela a uma rua de sentido único, cruzando com outras três vias, duas de sentido único e uma de duplo sentido. Pode-se observar melhor o sistema em estudo na figura abaixo.

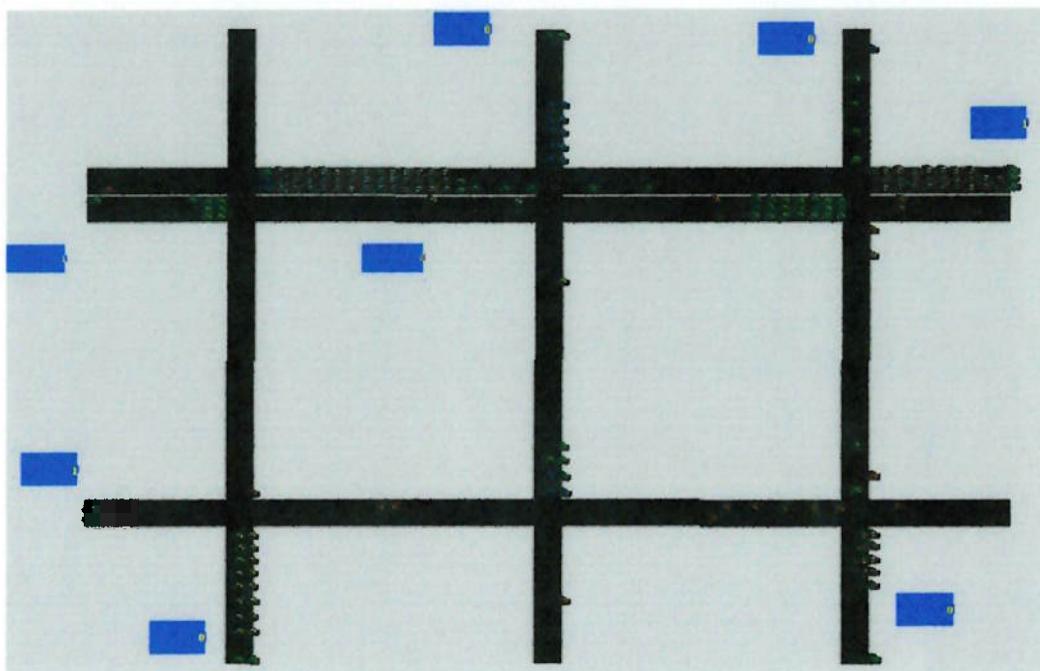


Figura 3 - Sistema em estudo

Como dito no tópico anterior, esse sistema foi construído baseado em uma região da Avenida Paulista na cidade de São Paulo, adotando suas principais características como as conversões possíveis, o tempo dos semáforos e a extensão das vias. Essa região é composta pelas seguintes vias: Avenida Paulista, Rua Bela Cintra, Rua Haddock Lobo, Rua Augusta e Alameda Santos.

4.3. IDENTIFICAÇÃO DO PROBLEMA

Em meio ao tráfego do período diurno, pode-se notar que durante o deslocamento de um veículo emergencial (EV), observa-se a dificuldade de locomoção do mesmo em meio a outros veículos que ocupam a via.

Durante o sinal vermelho, em que os veículos ficam parados, observa-se que o veículo com prioridade se move muito lentamente, apesar de os outros motoristas abrirem caminho entre eles. Assim, surge a necessidade de agilizar a movimentação do veículo para oferecer à população um atendimento mais eficiente.

De acordo com observações realizadas durante a elaboração deste projeto, pôde-se constatar que praticamente todo o tempo perdido pelo EV ocorre quando existem outros veículos a sua frente. Exetuando-se as condições extremas de tráfego intenso, veículos parados ocorrem quando o sinal do semáforo encontra-se vermelho. Assim, a estratégia de controle se focará na redução do tempo que esses veículos permanecem parados, atrapalhando a movimentação do EV.

4.4. SOLUÇÕES PROPOSTAS/ADOTADA DA ESTRATÉGIA DE CONTROLE

Para reduzir o problema de veículos parados a frente do EV durante o sinal vermelho, diversas soluções foram propostas. A solução escolhida no entanto, baseia-se nos sistemas de priorização de movimentação de EV (*Emergency Vehicle Signal Preemption*) já implantando em várias cidades dos Estados Unidos [Nelson, 2003]. Nesse sistema, sensores instalados nos cruzamentos detectam a aproximação de um veículo emergencial, reduzindo automaticamente o tempo de vermelho e alternando assim mais rapidamente para o sinal verde, ou no caso de o verde já estiver acionado, aumentar o tempo de permanência nesse estado, facilitando a movimentação do EV.

O sistema de detecção funciona com um emissor de sinal (rádio, luz ou som, p. ex.) montando no veículo e um receptor montado nos semáforos. Há sempre o respeito à ordem

correta dos estados (verde, amarelo e vermelho, nessa ordem). Após a passagem do EV por um cruzamento, os semáforos deste voltam a operar normalmente, com os tempo anteriormente adotados.

De acordo com análises realizadas por uma empresa fornecedora de um sistema de priorização [Nelson, 2003], é possível obter uma redução de aproximadamente 20 % no tempo de resposta a uma situação de emergência, isso devido apenas à atuação do sistema de priorização.

Assim, optou-se por implementar um sistema equivalente neste projeto, com atuação isolada para cada cruzamento, ou seja, procurando intervir em um semáforo de cada, minimizando os impactos às outras vias.

Para a obtenção dos tempos e a forma de atuação do sistema de controle, uma simulação foi desenvolvida, a ser melhor explicada no próximo item.

5. REGRAS DE CONTROLE

Para a obtenção das regras de controle, uma simulação computacional se mostra muito eficiente, por fornecer dados confiáveis e em pouco tempo, podendo-se observar a atuação do sistema de controle. A seguir, mais detalhes da simulação.

5.1. DESENVOLVIMENTO DA SIMULAÇÃO

5.1.1. ESCOLHA DO SOFTWARE

Para o desenvolvimento do modelo para a realização da simulação, será utilizado o ProModel®, um software comercial voltado para a modelagem e simulação de sistemas de manufatura. Apesar de ser voltado para aplicações industriais, ele atende muito bem às necessidades deste projeto, mesmo não possuindo elementos de modelagem específicos para essa finalidade, como por exemplo, sinais semafóricos.

O ProModel® é um software de simulação discreta baseado no sistema operacional Windows®, não necessitando de uma programação muito extensa, pois sua programação é orientada a objeto, além do que, sua interface gráfica possibilita fácil compreensão e visualização do modelo.

Devido às suas inúmeras funções como por exemplo, a facilidade de alteração das entradas de dados, modificações na estrutura do modelo, e nas regras de controle, esse software permite realizar uma simulação sob diferentes condições de operação. Pode-se nesse caso alterar o tempo dos ciclos dos semáforos, procurando reduzir o tempo de percurso

Para a análise dos resultados da simulação, o ProModel® possibilita a visualização dos resultados mediante a apresentação de relatórios fornecidos pelo próprio software, podendo-se obter dados como a ocupação média das vias, facilitando a busca por uma solução.

5.1.2. CONSTRUÇÃO DO MODELO

O processo de análise de sistemas reais através de simulações baseia-se na metodologia da simulação, sendo a modelagem o seu cerne. Para modelar parte do mundo real utiliza-se uma visão parcial, ou até mesmo simplificada, de parte desse mundo. O resultado desse processo de abstração do sistema real é geralmente uma representação estruturada chamada de modelo.

A simulação que se deseja realizar tem como objetivo a busca por uma configuração dos tempos dos semáforos para reduzir o tempo de percurso em uma rota determinada, sob a condição de necessidade de prioridade de movimentação. O modelo deve refletir a dinâmica do sistema nessas condições, dado que o tempo de percurso em uma determinada rota fica sujeito à duração dos ciclos dos semáforos e da taxa de entrada de veículos nesta rota; consequentemente, é diretamente influenciado por um congestionamento.

No desenvolvimento do modelo será admitido que a entrada de veículos na área modelada é constante durante um intervalo de tempo, ou seja, as modificações no fluxo de veículos ocorrem em tempos discretos. Além disso, os semáforos de pedestres existentes no mundo real podem ser desconsiderados para a análise, pois adicionam um valor fixo aos tempos dos ciclos semafóricos de cada intersecção.

Para uma melhor aproximação do mundo real, o modelo considerará que os veículos podem efetuar conversões das vias onde estão trafegando para as vias que a cruzam, mas será assumido que a proporção entre estes veículos e os que mantêm o seu curso será constante.

As vias externas ao modelo serão admitidas como capazes de absorver o fluxo de saída das vias modeladas, isto pode ser explicado, pois as vias que possuem características para serem modeladas são as principais vias de movimentação, concentrando o maior número de veículos e distribuindo estes para as áreas vizinhas.

Este modelo então, seria um modelo simplificado, mas complexo o suficiente para responder as questões levantadas no objetivo, possuindo as características de Sistemas a Eventos Discretos como já havia sido especificado em tópico anterior.

Para a simulação deste sistema será utilizado a Metodologia de Simulação de SED, viabilizando a avaliação mais detalhada de um grande número de configurações dos tempos semafóricos e a possibilidade de estabelecer critérios mais objetivos para comparação destas configurações.

Para a verificação e validação do modelo, será avaliado se o nível de acúmulo de veículos nas vias analisadas é coerente, ou seja, se o nível de congestionamento apresentado é razoável, não ultrapassando um limite crítico.

Após a validação do modelo, os cenários são definidos, sendo então realizada a simulação e a análise das informações obtidas.

5.1.3. ESTRATÉGIA OBTIDA

Como dito anteriormente, a estratégia obtida é semelhante aos sistemas de priorização de rota implantado em cidades dos EUA.

Basicamente a estratégia age da seguinte forma: caso o semáforo esteja na fase verde no momento da detecção do veículo que necessite de prioridade, existe uma extensão dessa fase. Caso esteja na fase vermelha, existe uma redução do tempo total da fase, evitando que o EV, assim como os outros veículos a frente, fiquem parado por mais tempo. Após a passagem do veículo, os tempos alterados retornam a sua configuração normal.

Por meio de diversas simulações, utilizando sempre as mesmas entradas, pode-se obter os tempos necessários para garantir 20 % de redução em média no tempo de percurso do EV.



Figura 4 - Modelo em simulação

A tabela 1 mostra os dados e a figura 5 mostra o gráfico contendo uma comparação entre os tempos de percurso para semáforos do tipo fixo e do tipo variável, para diferentes instantes de entrada para o EV.

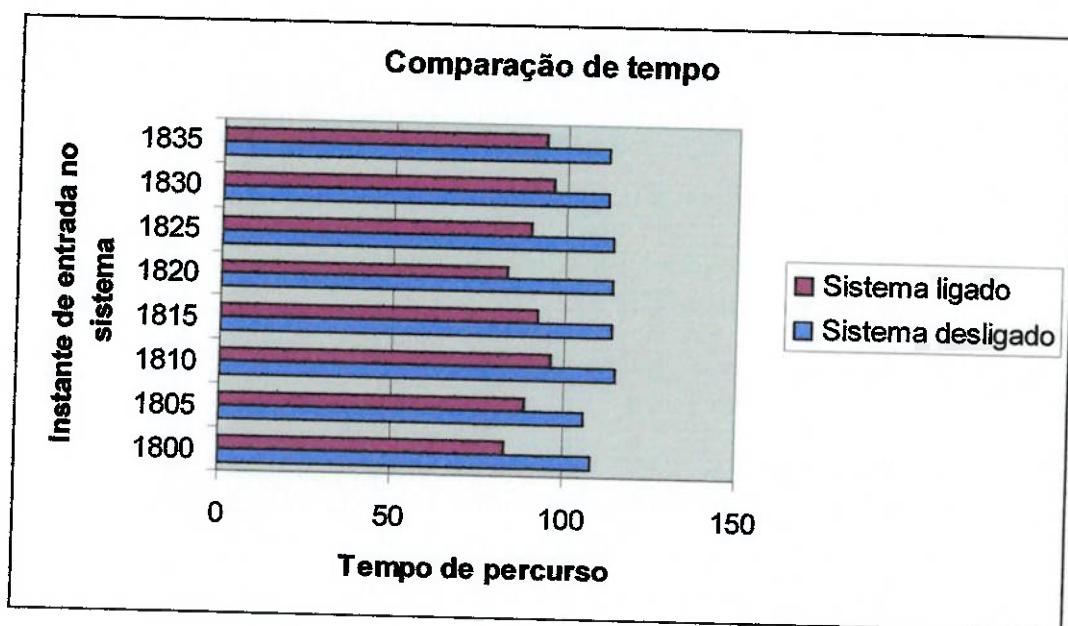


Figura 5 - Gráfico de comparação dos tempos de percurso

Tabela 1 - Resultados obtidos com a simulação

Instante de entrada (s)	sistema desligado (s)	sistema ligado (s)	%
1800	108	83	23,15%
1805	106	89	16,04%
1810	115	96	16,52%
1815	114	92	19,30%
1820	114	83	27,19%
1825	114	90	21,05%
1830	112	96	14,29%
1835	112	94	16,07%
Média			21,94%

A estratégia adotada funciona da seguinte forma: uma ambulância ao ingressar no sistema é detectada, por meio de um sensor. O semáforo adiante recebe o sinal do sensor, executando um ciclo de tempos alternativo. Esse ciclo alternativo, durante o sinal vermelho, reduz o tempo de espera com um tempo pré-determinado (definido nas simulações), ou no caso de o sinal ser verde, existe uma extensão da fase, adicionando um tempo fixo. A atuação da estratégia permanece assim até que um sensor posicionado após o cruzamento detecte a passagem da ambulância. Após a passagem, o ciclo do semáforo volta a operar normalmente. Essa estratégia se repete em todos os semáforos do percurso utilizado pelo EV.

6. DESENVOLVIMENTO DO PROTÓTIPO

O protótipo tem como intuito o maior detalhamento do sistema, utilizando como base o modelo construído para a simulação. O protótipo ainda deve possibilitar uma maior aproximação, compreensão e visualização da implementação real da estratégia adotada, adicionando a este modelo outras variáveis que não foram ou não poderiam ser consideradas na simulação unicamente computacional.

6.1. ARQUITETURA DE CONTROLE

No contexto do controle de malha viária, uma arquitetura de controle deveria fornecer uma boa representação do sistema descrevendo de forma completa e não-ambígua a sua estrutura e os relacionamentos existentes entre as entradas (inputs) e as saídas (outputs) do sistema. Esta representação é importante para o projeto e construção do protótipo. A partir da arquitetura, que descreve os componentes de controle em termos de tarefas e interações, é possível estabelecer a funcionalidade do sistema antes da sua implementação. Dos diversos diferentes tipos de arquitetura: centralizada, heterárquica e hierárquica, é nesta última na qual o protótipo deste projeto se baseia.

Sistemas com controle hierárquico são baseados e construídos utilizando o conceito de níveis de controle, sendo arranjados em uma estrutura piramidal. Este tipo de controle estabelece relacionamentos do tipo *mestre-escravo*. Cada nível de controle é responsável pela decomposição do comando recebido, do seu nível superior, em sub-comandos mais detalhados e sua posterior passagem para o nível inferior. Neste tipo de controle é usada a filosofia de distribuição de tomada de decisão, onde cada nível toma suas decisões de acordo com comandos do nível superior e resultados do nível inferior [Fried, 1996].

Seguindo esta idéia de arquitetura e adaptando para a elaboração do protótipo, cada cruzamento da malha viária possui um controlador que responde diretamente a um supervisório. O supervisório tem a função de receber e enviar dados a cada um dos controladores de acordo com a lógica de controle, além de monitorar o comportamento do

sistema a cada instante. Os controladores são responsáveis receber e enviar dados do supervisório fazendo o contato com o mundo exterior por meio de dispositivos de detecção, comando e monitoramento.

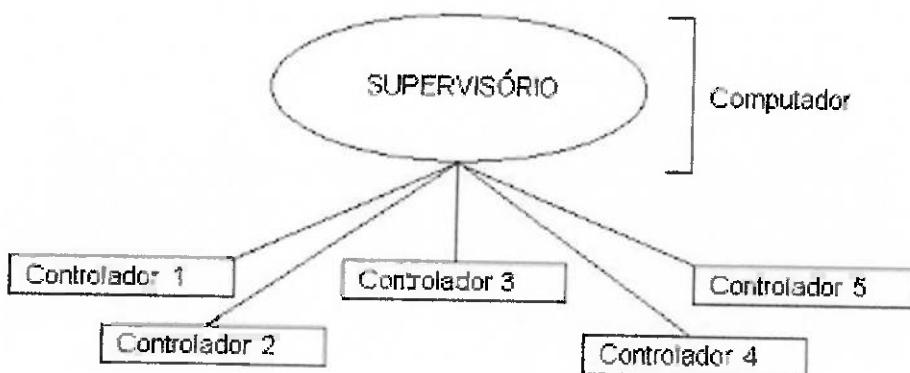


Figura 6 - Cruzamentos de uma malha viária

Uma adaptação do protótipo é a de considerar apenas uma dessas ligações (controlador-supervisório) em que um computador fará as funções do supervisório e um Controlador Lógico Programável (CLP) alocará, não apenas um cruzamento da malha viária, mas todos os cruzamentos do modelo adotado.

Resumindo, o protótipo será então baseado em uma comunicação serial entre um CLP e um computador, na qual o primeiro será uma representação do contato físico dos equipamentos com o mundo (sensores e semáforos) e o segundo será um supervisório do tráfego de veículos, tendo o controle dos ciclos dos semáforos e monitoração do tráfego a partir da manipulação e acompanhamento dos estados das entradas e saídas do CLP.

A figura 7 ilustra a montagem dos equipamentos que formam o protótipo.



Figura 7- Layout dos equipamentos

6.2. ESTUDO DA PLATAFORMA DE HARDWARE E SOFTWARE

O software TRILOGI 5.3 (Education) Aplication é uma ferramenta muito útil, pois pode simular o funcionamento de um programa Ladder de controle real a ser ou já implementado, permitindo treinamento prático sem a necessidade do hardware do controlador estar fisicamente conectado ao microcomputador. As saídas, entradas, timers e variáveis externas booleanas podem ser monitoradas durante a execução do programa.

O RSLOGIX 500 é o software de programação do CLP Micrologix que ajuda a maximizar a performance, minimizar o tempo gasto no desenvolvimento do projeto, é de fácil manuseio e instalação. É desenvolvido para operar em sistema operacional comercial em conjunto com um CLP Allen-Bradley de família de processadores Micrologix. É responsável por traduzir o programa em Ladder testado e enviá-lo ao CLP.

O controlador programável Micrologix 1500 contém uma fonte de energia, circuitos de entradas e saídas e um processador. O controlador possui um processador 1764-LSP, 12

entradas de 24V, 12 saídas de 24V, 2 saídas tipo relê e uma fonte de alimentação de 120/240V. Possui interface de comunicação RS232 para programação ou IHM (HMI).

O modelo Micrologix 1500 1761-L32BBB do fabricante Allen Bradley tem a capacidade de processar 500 instruções em 1.56ms e conectividade com interfaces opcionais DH485 e DeviceNET.

Para acessar as variáveis internas do Micrologix 1500, é necessário a utilização do RSLINX, que permite a comunicação com uma enorme variedade de aplicativos. Permite ainda que dados e parâmetros individuais sejam lidos e escritos em tempo real para os mais diversos fins. O RSLINX suporta as mais variadas formas de comunicação como RS-232, DH-485 ou PCMCIA.

O supervisório deverá ter boa interface visual, ser de fácil manipulação pelos usuários, possibilitar a comunicação através de portas seriais e ser compatível com o desenvolvimento de sistemas orientados a objeto. Sabendo disso, o software Visual Basic (VB) ou Visual Basic for Applications é uma das alternativas para este projeto, tendo larga utilização tanto comercialmente quanto didaticamente, e ainda tem como vantagem a larga fonte de consulta disponível e modelos de interface conhecida.

7. ESPECIFICAÇÃO DO PROTÓTIPO

Como descrito anteriormente, a figura 8 ilustra o diagrama estrutural dos elementos que compõe o protótipo.

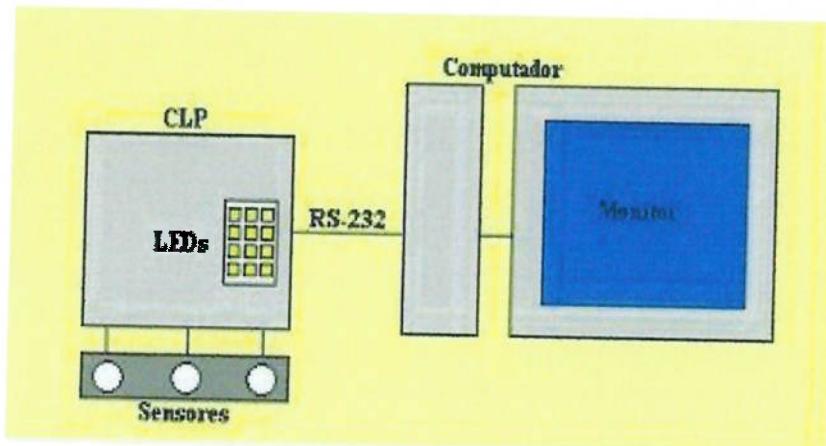


Figura 8 - Diagrama estrutural do protótipo

A partir dos resultados obtidos e considerando o mesmo modelo e hipóteses adotadas na simulação, o protótipo terá as seguintes características:

Serão utilizados os LEDs do CLP para a representação das luzes dos semáforos mais significativos na rota de percurso de um EV. (Figura 9)



Figura 9 - CLP e Visualização dos LEDs

Devido ao baixo número LEDs disponíveis, não será possível representar todos os semáforos contidos no modelo. Para fazer com que essa restrição não seja um impedimento, os outros semáforos serão representados por variáveis internas do CLP, mas seguindo a mesma lógica aplicada para as saídas através dos LEDs.

Os estados dessas variáveis internas serão enviados para o computador, assim como o estado das variáveis responsáveis pelo acendimento dos LEDS, passando pela lógica de controle.

Para a detecção do EV tentou-se reproduzir um sistema de emissor-receptor, onde há um emissor de sinal (rádio, luz ou som, por exemplo) montado no veículo e um receptor montado em local estratégico próximo aos semáforos. Desta maneira, para a representação dos sensores dos semáforos na detecção da proximidade de um EV, serão utilizados os sensores magnéticos próprios do CLP (Figura 10). Quando estes sensores indicarem uma mudança do estado de uma variável do CLP, o supervisório deverá receber esta informação pela comunicação serial e iniciar a alteração nos tempos dos semáforos, estes, determinados através da simulação.

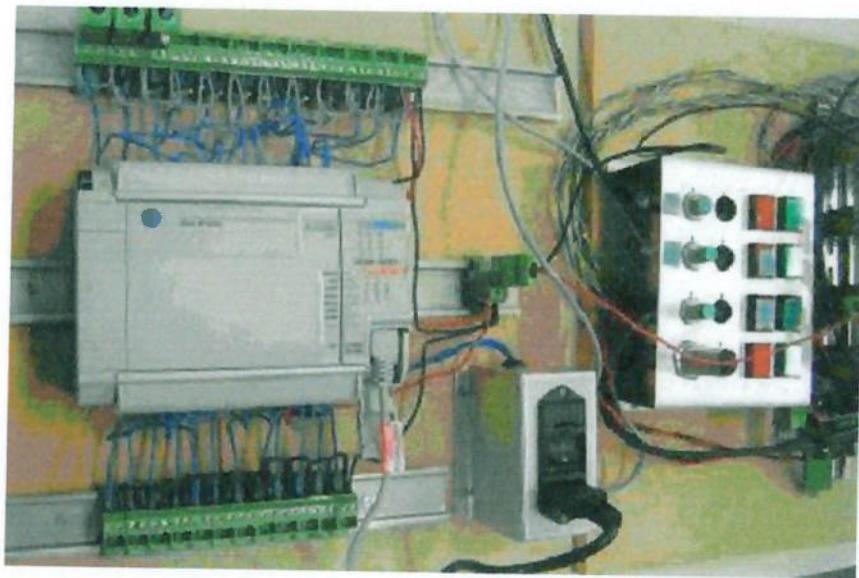


Figura 10 - CLP e os sensores (Inputs)

7.1. PROGRAMAÇÃO DO CLP

Para se chegar à efetiva programação do CLP é feita a seguinte metodologia (principais tópicos):

- Análise das necessidades e especificações;
- Elaboração do Sequential Function Chart (SFC) de um cruzamento;
- Transcrição do SFC para diagrama Ladder sendo este, elaborado e testado no software TRILOGI;
- Execução do programa em Ladder pelo RSLOGIX. . Este é responsável por traduzir o programa em Ladder e enviá-lo ao CLP.

7.1.1. ELABORAÇÃO DO SEQUENTIAL FUNCTION CHART (SFC)

A construção de uma representação gráfica se torna necessária para o melhor entendimento de um controle seqüencial e posteriormente, facilitar a transcrição do modelo para a linguagem de programação em Ladder [Silveira, 1998].

Como todo semáforo obedece à mesma ordem de etapas é necessário somente construir o ciclo do semáforo de apenas um modelo de cruzamento em SFC, pois todos os outros serão semelhantes. A figura 11 abaixo ilustra o diagrama SFC de um dos cruzamentos a ser implementado.

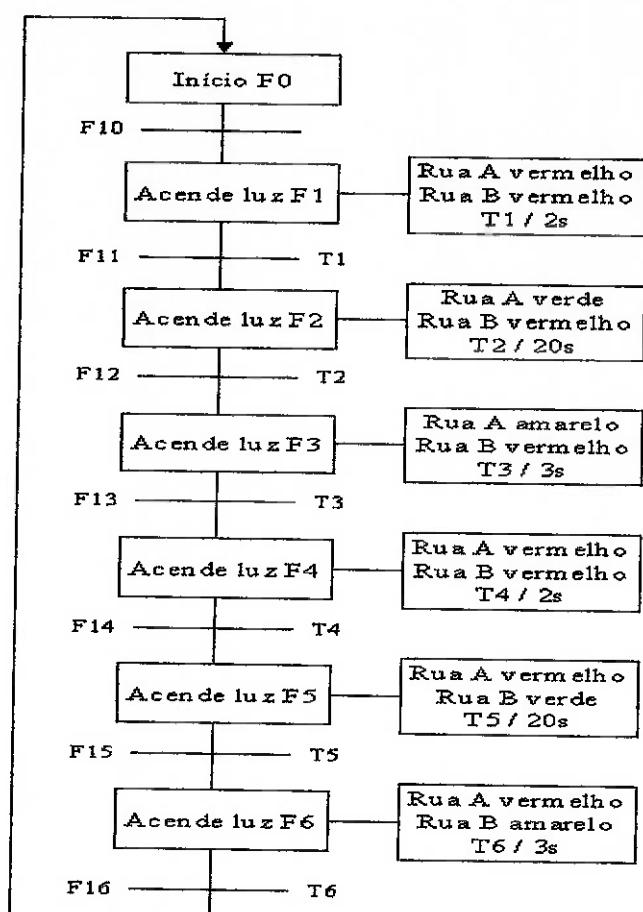


Figura 11 - Diagrama SFC

Neste ciclo pode-se observar, que as saídas são referentes ao primeiro cruzamento, cujo cruzamento é entre as ruas A e B (Figura 12).

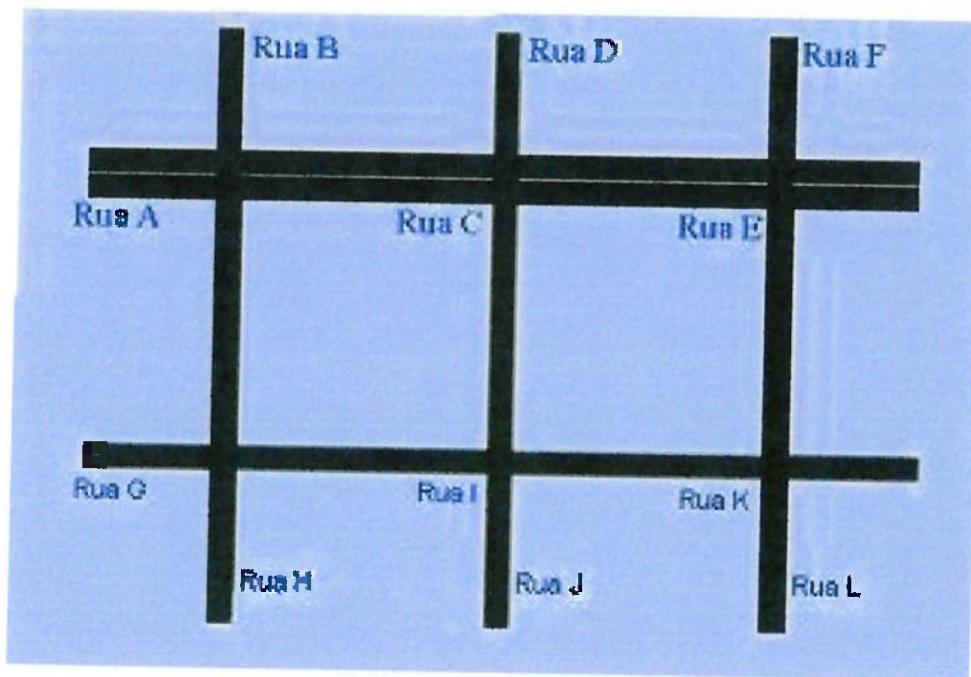


Figura 12 - Nomeação dos trechos das vias

Assim, para este cruzamento têm-se as ações contendo as saídas e os timers. As saídas são denominadas de: “Rua A vermelho”, “Rua A amarelo”, “Rua A verde”, “Rua B vermelho”, “Rua B amarelo” e “Rua B verde”. Os timers denominados de “T1” a “T6”.

As transições do controle seqüencial são denominadas de “F10” a “F16” e as etapas de “F0” a “F6”.

Nota-se que assim como o modelo aplicado na simulação, este ciclo de cruzamento também leva em conta o tempo de entre-verdes dos semáforos de 2 segundos.

7.1.2. PROGRAMAÇÃO EM LADDER

A chamada programação ladder é, na realidade, o desenho direto do controle em forma de diagrama Ladder, um esquema representativo do quadro de relés que seria utilizado caso relés fossem utilizados na implementação do controle.

A programação ladder não é uma linguagem de programação e sim uma forma de codificação. Tanto que os sistemas PLC não armazenam diretamente o diagrama ladder desenhado, e sim o programa decorrente que é gerado a partir do desenho, geralmente em lista de instruções [Santos Filho, 2003].

A programação em Ladder se encontra no anexo B. Esta programação leva em conta a transcrição (SFC para Ladder) dos seis cruzamentos do sistema (Figura 12). A metodologia de transcrição segue a referência de controle discreto [Silveira, 1998].

A modelagem do sistema resume-se da seguinte forma:

- Entradas: sensores de presença de EV's.
- Saídas: Luzes vermelhas, amarelas e verdes dos semáforos dos seis cruzamentos do modelo.

A seguir, seguem as tabelas dos endereços onde os bits dos timers, outputs e inputs foram alocados no CLP:

Tabela 2 - Lista dos símbolos e endereços dos timers alocados no CLP

TIMERS

Cruzamentos entre:

A e B		C e D		E e F		G e H		I e J		K e L	
Simb.	End.	Simb.	End.	Simb.	End.	Simb.	End.	Simb.	End.	Simb.	End.
T1	T4:0	U1	T4:6	V1	T4:12	X1	T4:18	Y1	T4:24	Z1	T4:30
T2	T4:1	U2	T4:7	V2	T4:13	X2	T4:19	Y2	T4:25	Z2	T4:31
T3	T4:2	U3	T4:8	V3	T4:14	X3	T4:20	Y3	T4:26	Z3	T4:32
T4	T4:3	U4	T4:9	V4	T4:15	X4	T4:21	Y4	T4:27	Z4	T4:33
T5	T4:4	U5	T4:10	V5	T4:16	X5	T4:22	Y5	T4:28	Z5	T4:34
T6	T4:5	U6	T4:11	V6	T4:17	X6	T4:23	Y6	T4:29	Z6	T4:35

Tabela 3 - Lista dos símbolos e endereços das saídas alocadas no CLP

OUTPUTS

Cruzamentos entre:

A e B		C e D		E e F	
Simb.	End.	Simb.	End.	Simb.	End.
a_red	O:0/0	c_red	O:0/4	e_red	O:0/8
a_yellow	O:0/1	c_yellow	O:0/5	e_yellow	O:0/9
a_green	O:0/2	c_green	O:0/6	e_green	O:0/10
b_red	O:0/3	d_red	O:0/16	F_red	O:0/19
b_yellow	O:0/7	d_yellow	O:0/17	F_yello w	O:0/20
b_green	O:0/11	d_green	O:0/18	F_green	O:0/21
G e H		I e J		K e L	
Simb.	End.	Simb.	End.	Simb.	End.
g_red	O:0/32	i_red	O:0/36	k_red	O:0/40
g_yellow	O:0/33	i_yellow	O:0/37	k_yellow	O:0/41
g_green	O:0/34	i_green	O:0/38	k_green	O:0/42
h_red	O:0/35	j_red	O:0/48	L_red	O:0/51
h_yellow	O:0/39	j_yellow	O:0/49	L_yellow	O:0/52
h_green	O:0/43	j_green	O:0/50	L_green	O:0/53

Tabela 4 - Lista dos símbolos e endereços das entradas alocadas no CLP

INPUTS

Cruzamentos entre:

A e B		C e D		E e F	
Simb.	End.	Simb.	End.	Simb.	End.
sensor_1	B:0/0	sensor_2	B:0/1	sensor_3	B:0/2

Observando-se as tabelas entende-se que para cada cruzamento há diferentes denominações dos timers e outputs.

Nota-se que há inputs somente para os três cruzamentos (AB, CD e EF) que fazem parte do percurso de um EV, já que somente neles estariam implementados receptores de ondas emitidas pelo EV.

7.2. CONSTRUÇÃO DO SUPERVISÓRIO

7.2.1. DESCRIÇÃO DO SUPERVISORES

Nos estudos realizados, utilizamos os diagramas de Reles, SFC e Ladder de maneira significativa. Os avanços adquiridos em programação para estes diagramas são obviamente mensuráveis, porém podem não fornecer ao usuário final a adaptabilidade necessária na utilização do programa.

Assim, pode-se notar que seria complexo monitorar atividades através apenas dos diagramas, devido à baixa capacidade ilustrativa e intuitiva.

Esse tipo de situação pode provocar dificuldades à verificação, monitoramento e reação a possíveis problemas detectados e ações corretivas. Uma maneira de solucionar tal problema, seria a criação de um supervisório entre o usuário e o diagrama responsável pela lógica de controle, ou mais precisamente, entre o usuário e o CLP, obtendo as informações deste em tempo real.

Para que estas informações sejam obtidas um módulo de comunicação deve ser elaborado, sendo o responsável pelo recebimento das variáveis do CLP, como os Timers, Inputs e Outputs, armazenando em variáveis do supervisório, tratando os dados obtidos e exibindo na interface gráfica aqueles que necessitam desta ação, como mostrado na figura 13.

O módulo de comunicação também é o responsável por enviar os dados necessários para o módulo de controle, ou mais precisamente, a detecção de um EV se aproximando da área controlada. Esses dados então são interpretados pelo módulo de controle que envia instruções para o CLP de forma que o semáforo que está adiante execute o ciclo alternativo que foi determinado através das simulações executadas anteriormente.

Como dito anteriormente, o usuário interagirá com o supervisório através da interface gráfica, podendo enviar também instruções para o módulo de controle e de comunicação, por exemplo alterando manualmente os tempos semafóricos.

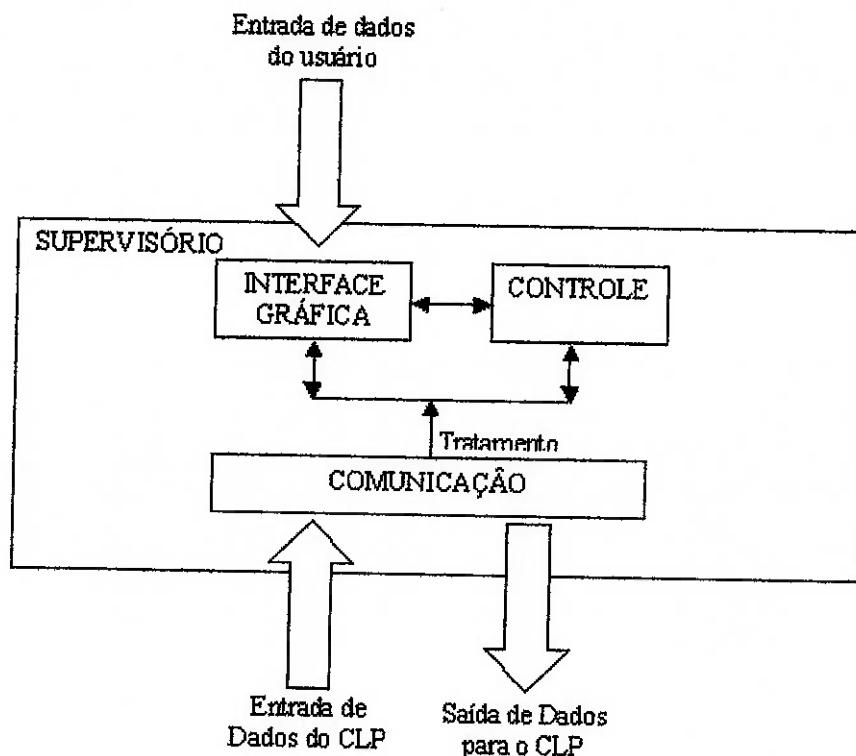


Figura 13 - Descrição do supervisório

7.2.2. MÓDULO DE COMUNICAÇÃO

As principais formas de comunicação existentes entre um computador e um dispositivo externo a ele, que neste projeto será o Controlador Lógico Programável, são a comunicação paralela, a comunicação serial e a comunicação por modems.

A comunicação paralela é caracterizada quando mais de um bit é transmitido ao mesmo tempo entre os dois equipamentos, geralmente o número de bits é igual a 8, já na comunicação serial, apenas um bit é transmitido de cada vez.

A capacidade de a comunicação paralela transmitir, por exemplo, 8 bits ao mesmo tempo faz com que esta seja mais rápida e eficiente, mas não faz com que seja a melhor opção para todos os projetos. A grande desvantagem da comunicação paralela é que para grandes distâncias o custo se eleva muito, o que não acontece com a comunicação serial.

Este fato faz com que a maioria dos periféricos possuam portas de comunicação serial ao invés de paralela e o CLP que estamos utilizando é um destes casos.

Quanto à comunicação por modems, a distância de transmissão dos dados não possui limites, desde que o local para onde se deseja enviá-los possua a estrutura necessária. Os custos são independentes da distância de transmissão, mas são variáveis de acordo com a velocidade de transmissão dos dados necessárias. Para o projeto em questão a velocidade necessária é de apenas alguns bits por segundo, correspondente a transmissão dos estados das variáveis internas do CLP, portanto não pode ser considerado uma restrição de projeto. Quanto a possibilidade de falhas, a transmissão de dados por modems é menos vantajosa em relação ao outros tipos de transmissão, já que a maioria dos protocolos de comunicação, inclusive o TCP/IP, um dos mais utilizados atualmente, não asseguram a chegada das informações no seu destino[GOODWIN].

A transmissão a longa distância é obrigatória para a comunicação do CLP com o usuário, e isto nos faz descartar a comunicação paralela como alternativa viável para este projeto. Nem todos os CLPs existentes no mercado dispõem de alternativas para a comunicação por modems, mas é sabido que facilmente podemos utilizar as saídas seriais desses modems e conectá-los diretamente em modems que posteriormente enviar as informações para o seu destino.

Estas informações permitiram a escolha da comunicação serial como melhor alternativa para o projeto, fazendo com que seja uma solução menos restrita a disponibilidade de hardware e ainda construindo um projeto que pode ser reutilizado para os mais diversos tipos de CLP existentes.

Para estabelecer a comunicação é necessário utilizar o software RSLINX, pois é este que habilita o acesso as variáveis internas do CLP, tanto para leitura quanto para escrita. Para a comunicação com o sistema será utilizado será utilizado o OLE for Process Control (OPC) que é um padrão de comunicação criado para interfaces entre aplicações de controle e automação.

OPC é um padrão de comunicação cliente/servidor estruturada como Cliente OPC, Servidor OPC, Grupo OPC e Item OPC. O Cliente OPC é quem inicia a comunicação pedindo ou enviando algum tipo de dado e o Servidor OPC é o responsável por aceitar ou informar o dado que está sendo requisitado caso esta informação faça algum sentido para ele. Um cliente pode acessar um ou mais Servidores OPC, que por sua vez podem ser acessados por um ou mais Clientes OPC, como ilustrado na figura 14.

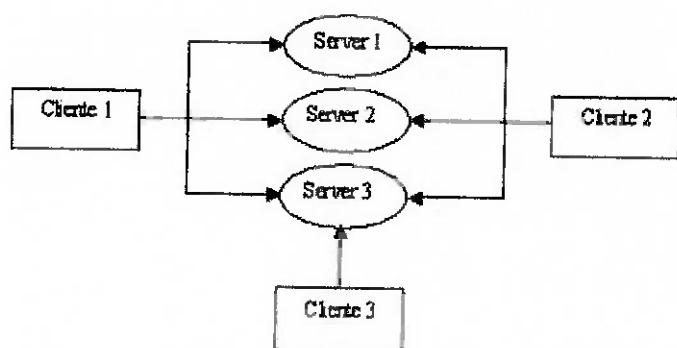


Figura 14 - Relação Cliente-Servidor

O servidor OPC mantém todas as informações armazenadas dentro dos Grupos OPC. Os Grupos OPC permitem que as informações sejam organizadas separadamente, mantendo as informações sobre ele mesmo e dando os mecanismos necessários para que os Itens OPC sejam organizados e endereçados logicamente. Estes grupos podem ainda ser configurados de forma a permitir acesso de qualquer Cliente OPC, neste caso eles são chamados de Grupos Públicos, ou de apenas alguns clientes, neste caso ele é chamado de Grupo Locais ou Privados.

Os Itens OPC representam as conexões entre a origem dos dados com o servidor, mas ele não pode ser acessado diretamente pelo Cliente, sendo necessário sempre um Grupo para fazer este acesso. Por este motivo é dito que o Item OPC pode ser descrito simplesmente como o endereço dos dados e não como a origem física dos dados ao qual é referenciado [SILVEIRA, 1998].

Através desta estrutura podemos determinar que o Cliente OPC é o Supervisório, o servidor é o CLP. As variáveis internas do CLP como saída, entradas e timer são os Itens OPC que devem ser organizados dentro dos Grupos OPC e Servidores OPC da forma que estes sejam representativos as funções exercidas.

7.2.3. MÓDULO DE CONTROLE

O módulo de controle é o responsável pela tomada de decisões quanto a necessidade de aplicar a priorização de rotas para um EV que tenha sido detectado. Como dito anteriormente, esta detecção pode ser feita das mais diversas maneiras através da utilização de sensores, isto porque o supervisório recebe através do módulo de comunicação o estado das variáveis de entrada do CLP.

Ainda existe um outro tipo de ativação da estratégia de priorização que pode ser utilizada. Esta pode ser considerada como uma contingência para casos em que os sensores que estiverem sendo utilizados estejam falhando, seja não detectando EVs que estejam presentes na malha viária que está sendo controlada, seja detectando EVs que não estão presentes na malha. Isto pode ser feito através da interação do operador do supervisório com o módulo de controle através do módulo de monitoração.

Como descrito na estrutura da figura 13, as entradas, sendo elas as variáveis internas de Input do CLP ou uma interação do operador, serão verificadas pelo módulo de controle e identificadas como sendo a presença ou ausência de um EV, e também identificando a qual cruzamento corresponde aquela entrada. Feito isso, as regras de decisões, formuladas como Regras de Produção, serão responsáveis por verificar para aquela entrada, qual o tempo que deverá ser aplicado para aquele cruzamento, tempo este determinado através das simulações feitas anteriormente.

Tendo estes tempos, envia-se para variáveis internas de saída do CLP, através do módulo de comunicação, os valores necessários para realizar a Estratégia de Controle, estas variáveis são em sua grande maioria os valores máximos dos contadores dos Timers.

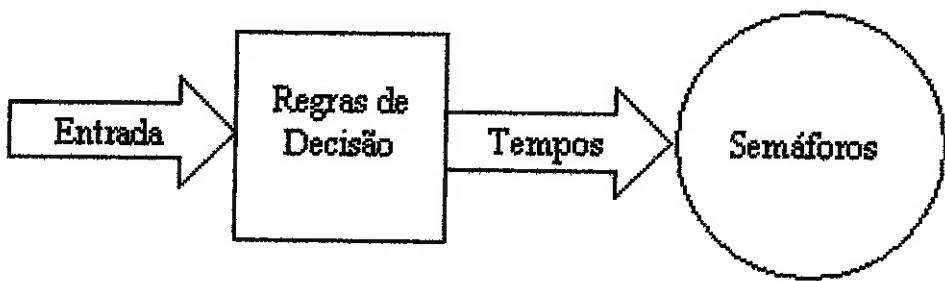


Figura 15 - Estrutura do Módulo de Controle

7.2.4. MÓDULO DE MONITORAÇÃO

O módulo de monitoração é por onde o operador realizará todas as interações possíveis e também por onde serão verificados e monitorados os estados de todos os semáforos contidos na malha viária.

Para que as necessidades sejam atendidas, são criados telas, onde visualmente seja possível essa monitoração, sendo ela diferente para cada uma das malhas viárias que estiverem sendo controladas.

A figura 14 mostra a tela da malha viária que está sendo controlada pelo supevisório. A interface gráfica foi feita de forma que o seu entendimento fosse intuitivo. As setas indicam a direção em que o movimento está ocorrendo e a cor desta seta indica qual o estado do semáforo que controla aquele sentido de movimento.

Também é possível ver, qual é o valor da contagem dos timers que controlam aquele semáforo através de número que são alterados na parte interna dessas setas. Estes números são alterados constantemente através da leitura do envio contínuo das informações necessárias através do módulo de comunicação.

Quando a presença de um EV é detectada, o módulo de controle envia para o módulo de monitoração dados para que o módulo de monitoração indique para o operador que naquela malha viária existe um EV e que a estratégia de priorização de rota está ativada. Para a malha viária em questão, isto acontece através de um alternamento das

cores da faixa central da via correspondente a Av. Paulista entre as cores branca e vermelha.

A interação do usuário pode ser feita através dos botões presentes na parte superior esquerda da tela de monitoração. Onde são disponibilizados as funções de ativação/desativação da estratégia de controle para um cruzamento desejado e alteração manual do tempo de verde de um cruzamento escolhido.

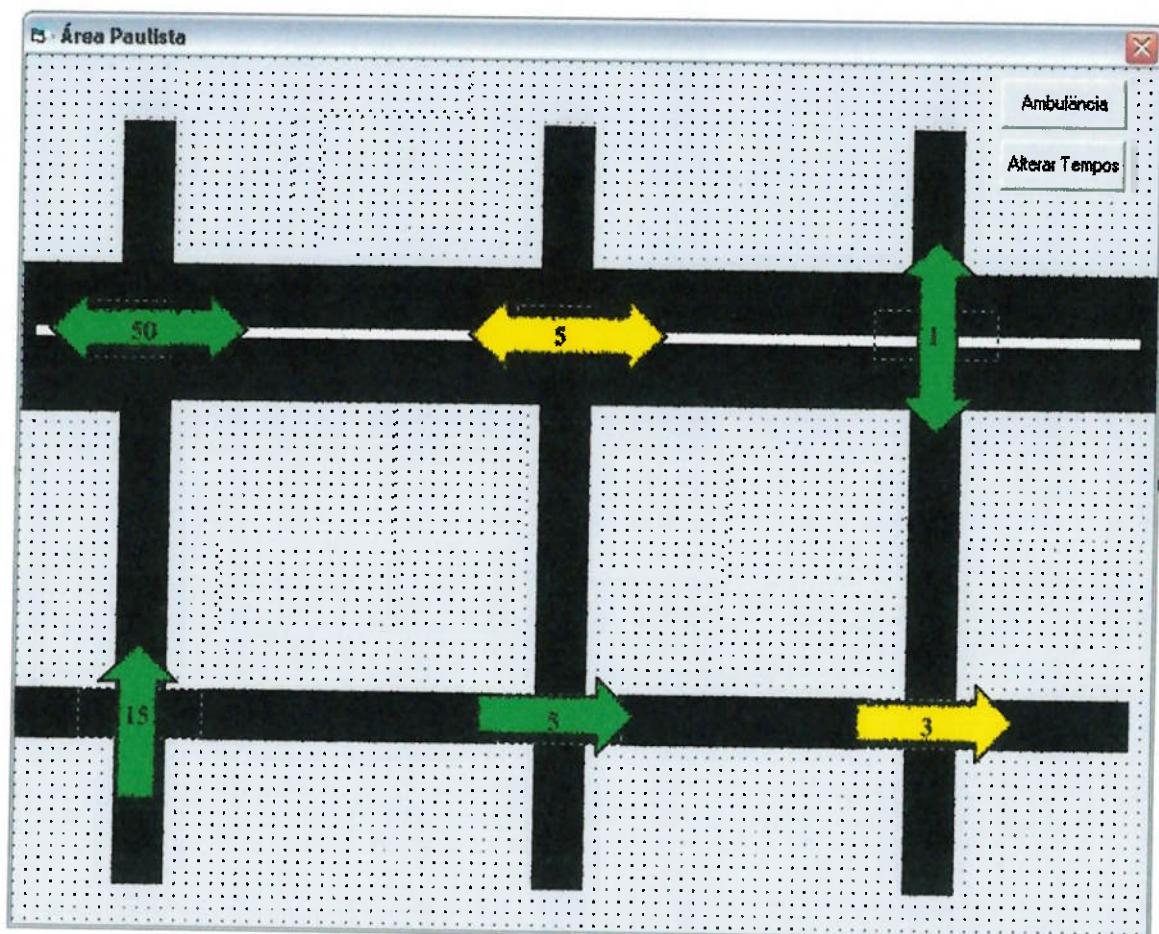


Figura 16 - Tela de Monitoração

7.2.5. DESENVOLVIMENTO DO SOFTWARE

A elaboração de um sistema deve prever reformulações e manutenções futuras e o método que vem se mostrado mais eficaz é o Método de Desenvolvimento de Sistemas Orientados a Objetos. Isto se deve ao fato de que um projeto de sistemas orientado a objeto melhora a extensibilidade, promove a reusabilidade e encapsula as dependências do sistema, permitindo a reutilização, configuração ou seleção de componentes comercialmente avaliados, agregando novas funcionalidades ao sistema sem a necessidade de retrabalhos ou grandes esforços[ROCKWELL, 2002].

O objetivo deste trabalho não é seguir a risca as normas muitas vezes estabelecidas para o desenvolvimento de sistemas orientado a objeto gerando o grande número de documentos sugeridos, mas sim tomar como base as boas práticas indicadas pelo RUP (Rational Unified Process) para a engenharia de software, para tentar possibilitar uma visão geral do escopo do projeto, mesmo antes dele começar a ser transformado em códigos de programação, permitindo avaliações de qualidade e atendimento das necessidades e evitando a reengenharia ou reprogramação.

Um dos principais modelos utilizados para entendimento das funcionalidades de um sistema a ser desenvolvido na orientação a objeto é o diagrama de casos de uso. Este diagrama é um elemento gráfico que ilustra uma parte ou aspecto particular do sistema, demonstrando os atores externos e as funcionalidades que o sistema fornece.

O ator é alguém ou algo fora do sistema que interage com ele de alguma forma, representa um papel humano, hardware ou outro sistema em relação ao sistema em desenvolvimento. O nome do ator deve claramente referenciar o seu papel.

Os casos de uso representam algo que o usuário/cliente quer que o sistema faça. Define uma sequência de ações executadas por um sistema, que trazem um resultado de valor observável para um ator. É uma descrição completa de um processo. Cada caso de uso poderá se tornar uma ou mais funções dentro do sistema, tendo entradas e saídas pré-determinadas.



Figura 17 - Exemplo de Diagrama de Caso de uso

Os atores e os casos de usos mapeados para a construção do protótipo estão descritos nos documentos anexos [ABRIL, 2002].

7.3. OBSERVAÇÕES FINAIS DESTE CAPÍTULO

A construção de um protótipo que permita a visão geral de um projeto e de suas funcionalidades é de extrema importância para um projeto de sistemas. Isto pode ser alcançado através da metodologia de desenvolvimento de sistemas orientados a objeto e as boas práticas do RUP.

Com relação à estratégia de controle, existem atualmente estudos na área com o propósito de aumentar a eficiência do sistema em estudo. Um sistema muito interessante e com grandes chances de sucesso, chamado de *Dynamic Route Clearance*, utiliza em conjunto com o sistema *Emergency Vehicle Preemption*, um sistema de cálculo e geração de uma rota menos congestionada, de acordo com análises em tempo real das condições de tráfego [Nelson, 2003]. No entanto, o desenvolvimento desse sistema complexo foge dos objetivos deste trabalho.

8. CONCLUSÃO

Com base em estratégias já adotadas em cidades dos Estados Unidos e adaptando-as para o estudo de caso deste projeto, pode-se comprovar, de acordo com análises realizadas na simulação, que a adoção desta estratégia de controle reduz o tempo de percurso de veículos emergenciais, podendo-se realizar um atendimento mais eficiente a toda a população. No entanto, vale lembrar que este tipo de estratégia, bem como qualquer outra, não pode ser aplicada em situações de tráfego intenso, onde dificilmente haveria uma solução para essa situação. Considera-se promissor um estudo para a implantação desse sistema, ou similares, em cidades brasileiras, carentes de controle inteligente de semáforos. Pode-se também aplicar este sistema em conjunto com outras estratégias de priorização de rota, como por exemplo o cálculo de uma rota menos congestionada, de forma a tornar o resultado ainda mais eficiente.

Pela análise dos conceitos de malha viária e sistemas a eventos discretos pode-se modelar, simular e programar todos os comportamentos necessários à organização de uma malha viária urbana. E consequentemente desta forma, pode-se ao longo deste trabalho desenvolver habilidades como a programação no ProModel®, no Visual Basic for Applications, assim como o estudo das características do CLP e da implementação de rotinas em sua memória.

O sucesso na construção de um protótipo mostrou-se muito útil, na medida em que um usuário do protótipo passa a compreender e visualizar melhor os elementos que fazem parte da estratégia de controle para priorização de rota de um veículo emergencial.

Seria interessante para um futuro trabalho, acrescentar outros eventos que não foram abordados neste trabalho e que freqüentemente ocorrem no trânsito das cidades grandes, como por exemplo, a ocorrência de colisões de veículos ou acidentes com pedestres. Desta maneira, a estratégia a ser adotada, a modelagem, a simulação do sistema e programação do projeto se tornaria muito mais complexa e próxima da realidade.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] ABRIL. Metodologia de gerenciamento de projetos e metodologia de desenvolvimento de sistemas. Disponível em:
<<http://www.abrilnet.com.br/index98.asp?pai=30&link=http://www1.abrilnet.com.br/servi/tecnologia/>>. Acesso em: 20 out. 2005.
- [2] BORGES DE PAULO, M. E. Desempenho do Sistema Viário de São Paulo. Notas Técnicas-CETSP. São Paulo, 1978.
- [3] CURY, J. E. R. Modelagem e Controle de Sistemas Automatizados. Notas de aula, disponível em: <<http://www.das.ufsc.br/~cury/cursos/transp-mod-cont-introd.pdf>>. Acesso em 09 dez. 2005
- [4] EJZENBERG, S. Reprogramação de semáforos: Métodos baseados em observação de campo. Notas Técnicas-CETSP, 1994.
- [5] FRIED, R. Conceitos e abordagens: uma revisão. Cap. 2. Teses, São Carlos, 1996. Disponível em: <<http://www.eps.ufsc.br/teses96/friedrich/cap2/capitulo2.htm>>. Acesso em 13 dez. 2005
- [6] GOODWIN, M. Serial communications programming in C and C++, 1992.
- [7] GSI. Grupo de Sistemas Inteligentes. Disponível em:<<http://www.din.uem.br/~ia/1024x768/intelige.html>>. Acesso em: 15 Jan de 2006.
- [8] GUSTIN, G. D. B. Aplicação de Redes de Petri Interpretadas na Modelagem de Sistemas de Elevadores em Edifícios Inteligentes. Dissertação de Mestrado, 1999.
- [9] HO, Y.C. Special Issue on Dynamics of Discrete Event Systems. In IEEE. Proc. 1989, v.77, p.3-6
- [10] ITANO, F. H. Desenvolvimento de Modelos e Estratégias de Controle Para um Sistema Viário. Trabalho de Formatura, 1997.

- [11] KWON, E.; Kim, S. Development of dynamic route clearance strategies for emergency vehicle operations, phase 1. Disponível em:
<<http://www.lrrb.gen.mn.us/PDF/200327.pdf>>. Acesso em: 20 ago. 2005.
- [12] MICROSOFT. Fórum de discussões Microsoft Visual Basic. Disponível em:
<<http://www.microsoft.com/brasil/msdn/newsgroups/default.mspx?dg=microsoft.public.br.vb&lang=pt&cr=BR&r=833f278b-a1db-4967-bb5d-0a7793f0b405>>. Acesso em 10 out. 2005.
- [13] MIYAGI, P. E. Controle Programável. Editora Edgard Blücher Ltda, 1997.
- [14] MOREIRA, V. Estudo DATEC-Pesquisas de Tráfego. Notas Técnicas-CETSP. São Paulo, 1979.
- [15] MULTIMEDIA.
<http://multimedia.mmm.com/mws/mediawebserver.dyn?eeeeeeeO48qtezIfefeeemj8Q9KA3TN>. Acesso em 15 nov. 2005.
- [16] NELSON, P. Dynamic route clearance: Saving time, saving lives. Disponível em:
<<http://www.its.umn.edu/sensor/2003/spring/signals.html>>. Acesso em: 20 ago 2005.
- [17] RATIONAL. Rational Unified Process. Disponível em:
<http://en.wikipedia.org/wiki/Rational_Unified_Process>. Acesso em 10 nov. 2005.
- [18] ROCKWELL AUTOMATION. Rockwell Automation, RSLinx Application Guide: ControlLogix Data Collection with RSLinx. U.S.A, 2002.
- [19] ROCKWELL AUTOMATION. MicroLogix 1500 Programmable Controllers User Manual. U.S.A, 1999.
- [20] ROCKWELL AUTOMATION. Fórum de discussões da Rockwell Automation – Softwares. Disponível em:
<<http://www.software.rockwell.com/forum/rslinx/categories.cfm?catid=73>>. Acesso em: 5 set. 2005.

- [21] SANTOS FILHO, D. J. Apostila A03 – Sequential Function Chart. PMR2460-Modelagem e Controle de Sistemas Discretos, 2004.
- [22] SANTOS FILHO, D. J. Apostila A02 - Modelos e Linguagens de Programação. PMR2460-Modelagem e Controle de Sistemas Discretos, 2003.
- [23] SILVEIRA, R. P.; Santos W. E. Automação e Controle Discreto. Editora Érica. São Paulo, 1998.
- [24] SCHREINER JÚNIOR, S.A. Sistema de Apoio à Decisão para o Controle do Tráfego em Interseções Isoladas. Disponível em:
<http://www.ceftru.unb.br/arquivos/sigeti_artigo.pdf>. Acesso em: 20 ago 2005.
- [25] SOLA, S.M. Modelo de simulação de Tráfego Urbano-UTCS-1. Notas técnicas-CETSP. São Paulo, 1979.
- [26] SZASZ, P. Soluções alternativas ao semáforo: aspectos teóricos. Notas Técnicas-CETSP, São Paulo, 1992
- [27] VILANOVA L. M. O controle dos semáforos em tempo real no Brasil. Disponível em: <<http://www.fcc.org.br/pesquisa/detranIdeias2.html>>. Acesso em: 14 jun. 2005.

ANEXOS

ANEXO A - Programação utilizada no simulador PROMODEL®

```
*****
*          *
*          * Formatted Listing of Model:          *
*          D:\promodel antigo\NOVO TF\MODELO_EVP_ATIVO_v5a.MOD      *
*          * ****
*****  
*****
```

Time Units: Seconds
Distance Units: Meters
Initialization Logic:
 ACTIVATE sub_semaforpb()
 ACTIVATE sub_semaforph()
 ACTIVATE sub_semaforpa()
 ACTIVATE sub_semaforosb()
 ACTIVATE sub_semaforosh()
 ACTIVATE sub_semaforosa()

```
*****
*****  
*          Locations          *
*****  
*****
```

Name	Cap	Units	Stats	Rules	Cost
Loc1pb	inf	1	Time Series	Oldest,	,
Loc2pb	3	1	Time Series	Oldest,	,
Loc3pb	3	1	Time Series	Oldest,	,
Loc4pb	3	1	Time Series	Oldest,	,
Loc5pb	3	1	Time Series	Oldest,	,
Loc6pb	3	1	Time Series	Oldest,	,
Loc7pb	3	1	Time Series	Oldest,	,
Loc8pb	3	1	Time Series	Oldest,	,
Loc9pb	3	1	Time Series	Oldest,	,
Loc10pb	3	1	Time Series	Oldest,	,
Loc11pb	3	1	Time Series	Oldest,	,
Loc12pb	3	1	Time Series	Oldest,	,
Loc13pb	3	1	Time Series	Oldest,	,
Loc14pb	3	1	Time Series	Oldest,	,
Loc15pb	3	1	Time Series	Oldest,	,

Loc16pb	3	1	Time Series Oldest, ,
Loc17pb	3	1	Time Series Oldest, ,
Loc18pb	3	1	Time Series Oldest, ,
Loc19pb	3	1	Time Series Oldest, ,
Loc20pb	3	1	Time Series Oldest, ,
Loc21pb	3	1	Time Series Oldest, ,
Loc22pb	3	1	Time Series Oldest, ,
Loc23pb	3	1	Time Series Oldest, ,
Loc24pb	3	1	Time Series Oldest, ,
Loc25pb	3	1	Time Series Oldest, ,
Loc26pb	3	1	Time Series Oldest, ,
Loc27pb	3	1	Time Series Oldest, ,
Loc28pb	3	1	Time Series Oldest, ,
Loc29pb	3	1	Time Series Oldest, ,
Loc30pb	3	1	Time Series Oldest, ,
Loc31pb	3	1	Time Series Oldest, ,
Loc32pb	3	1	Time Series Oldest, ,
Loc33pb	3	1	Time Series Oldest, ,
Loc34pb	3	1	Time Series Oldest, ,
Loc35pb	3	1	Time Series Oldest, ,
Loc36pb	3	1	Time Series Oldest, ,
Loc37pb	3	1	Time Series Oldest, ,
Loc38pb	3	1	Time Series Oldest, ,
Loc39pb	3	1	Time Series Oldest, ,
Loc40pb	3	1	Time Series Oldest, ,
Loc41pb	2	1	Time Series Oldest, ,
Loc42pb	2	1	Time Series Oldest, ,
Loc43pb	2	1	Time Series Oldest, ,
Loc44pb	2	1	Time Series Oldest, ,
Loc45pb	2	1	Time Series Oldest, ,
Loc46pb	2	1	Time Series Oldest, ,
Loc47pb	2	1	Time Series Oldest, ,
Loc48pb	2	1	Time Series Oldest, ,
Loc49pb	2	1	Time Series Oldest, ,
Loc50pb	2	1	Time Series Oldest, ,
Loc51pb	2	1	Time Series Oldest, ,
Loc52pb	2	1	Time Series Oldest, ,
Loc53pb	2	1	Time Series Oldest, ,
Loc54pb	2	1	Time Series Oldest, ,
Loc55pb	2	1	Time Series Oldest, ,
Loc56pb	2	1	Time Series Oldest, ,
Loc57pb	2	1	Time Series Oldest, ,
Loc58pb	2	1	Time Series Oldest, ,
Loc59pb	2	1	Time Series Oldest, ,
Loc60pb	2	1	Time Series Oldest, ,
Loc1ph	3	1	Time Series Oldest, ,
Loc2ph	3	1	Time Series Oldest, ,

Loc3ph	3	1	Time Series Oldest, ,
Loc4ph	3	1	Time Series Oldest, ,
Loc5ph	3	1	Time Series Oldest, ,
Loc6ph	3	1	Time Series Oldest, ,
Loc7ph	3	1	Time Series Oldest, ,
Loc8ph	3	1	Time Series Oldest, ,
Loc9ph	3	1	Time Series Oldest, ,
Loc10ph	3	1	Time Series Oldest, ,
Loc11ph	3	1	Time Series Oldest, ,
Loc12ph	3	1	Time Series Oldest, ,
Loc13ph	3	1	Time Series Oldest, ,
Loc14ph	3	1	Time Series Oldest, ,
Loc15ph	3	1	Time Series Oldest, ,
Loc16ph	3	1	Time Series Oldest, ,
Loc17ph	3	1	Time Series Oldest, ,
Loc18ph	3	1	Time Series Oldest, ,
Loc19ph	3	1	Time Series Oldest, ,
Loc20ph	3	1	Time Series Oldest, ,
Loc21ph	3	1	Time Series Oldest, ,
Loc22ph	3	1	Time Series Oldest, ,
Loc23ph	3	1	Time Series Oldest, ,
Loc24ph	3	1	Time Series Oldest, ,
Loc25ph	3	1	Time Series Oldest, ,
Loc26ph	3	1	Time Series Oldest, ,
Loc27ph	3	1	Time Series Oldest, ,
Loc28ph	3	1	Time Series Oldest, ,
Loc29ph	3	1	Time Series Oldest, ,
Loc30ph	3	1	Time Series Oldest, ,
Loc31ph	3	1	Time Series Oldest, ,
Loc32ph	3	1	Time Series Oldest, ,
Loc33ph	3	1	Time Series Oldest, ,
Loc34ph	3	1	Time Series Oldest, ,
Loc35ph	3	1	Time Series Oldest, ,
Loc36ph	3	1	Time Series Oldest, ,
Loc37ph	3	1	Time Series Oldest, ,
Loc38ph	3	1	Time Series Oldest, ,
Loc39ph	3	1	Time Series Oldest, ,
Loc40ph	3	1	Time Series Oldest, ,
Loc41ph	2	1	Time Series Oldest, ,
Loc42ph	2	1	Time Series Oldest, ,
Loc43ph	2	1	Time Series Oldest, ,
Loc44ph	2	1	Time Series Oldest, ,
Loc45ph	2	1	Time Series Oldest, ,
Loc46ph	2	1	Time Series Oldest, ,
Loc47ph	2	1	Time Series Oldest, ,
Loc48ph	2	1	Time Series Oldest, ,
Loc49ph	2	1	Time Series Oldest, ,

Loc50ph	inf	1	Time Series Oldest, ,
Loc51ph	2	1	Time Series Oldest, ,
Loc52ph	2	1	Time Series Oldest, ,
Loc53ph	2	1	Time Series Oldest, ,
Loc54ph	2	1	Time Series Oldest, ,
Loc55ph	2	1	Time Series Oldest, ,
Loc56ph	2	1	Time Series Oldest, ,
Loc57ph	2	1	Time Series Oldest, ,
Loc58ph	2	1	Time Series Oldest, ,
Loc59ph	2	1	Time Series Oldest, ,
Loc60ph	2	1	Time Series Oldest, ,
Loc1pa	3	1	Time Series Oldest, ,
Loc2pa	3	1	Time Series Oldest, ,
Loc3pa	3	1	Time Series Oldest, ,
Loc4pa	3	1	Time Series Oldest, ,
Loc5pa	3	1	Time Series Oldest, ,
Loc6pa	3	1	Time Series Oldest, ,
Loc7pa	3	1	Time Series Oldest, ,
Loc8pa	3	1	Time Series Oldest, ,
Loc9pa	3	1	Time Series Oldest, ,
Loc10pa	3	1	Time Series Oldest, ,
Loc11pa	3	1	Time Series Oldest, ,
Loc12pa	3	1	Time Series Oldest, ,
Loc13pa	3	1	Time Series Oldest, ,
Loc14pa	3	1	Time Series Oldest, ,
Loc15pa	3	1	Time Series Oldest, ,
Loc16pa	3	1	Time Series Oldest, ,
Loc17pa	3	1	Time Series Oldest, ,
Loc18pa	3	1	Time Series Oldest, ,
Loc19pa	3	1	Time Series Oldest, ,
Loc20pa	3	1	Time Series Oldest, ,
Loc21pa	inf	1	Time Series Oldest, ,
Loc22pa	3	1	Time Series Oldest, ,
Loc23pa	3	1	Time Series Oldest, ,
Loc24pa	3	1	Time Series Oldest, ,
Loc25pa	3	1	Time Series Oldest, ,
Loc26pa	3	1	Time Series Oldest, ,
Loc27pa	3	1	Time Series Oldest, ,
Loc28pa	3	1	Time Series Oldest, ,
Loc29pa	3	1	Time Series Oldest, ,
Loc30pa	3	1	Time Series Oldest, ,
Loc31pa	3	1	Time Series Oldest, ,
Loc32pa	3	1	Time Series Oldest, ,
Loc33pa	3	1	Time Series Oldest, ,
Loc34pa	3	1	Time Series Oldest, ,
Loc35pa	3	1	Time Series Oldest, ,
Loc36pa	3	1	Time Series Oldest, ,

Loc37pa	3	1	Time Series Oldest, ,
Loc38pa	3	1	Time Series Oldest, ,
Loc39pa	3	1	Time Series Oldest, ,
Loc40pa	3	1	Time Series Oldest, ,
Loc41pa	2	1	Time Series Oldest, ,
Loc42pa	2	1	Time Series Oldest, ,
Loc43pa	2	1	Time Series Oldest, ,
Loc44pa	2	1	Time Series Oldest, ,
Loc45pa	2	1	Time Series Oldest, ,
Loc46pa	2	1	Time Series Oldest, ,
Loc47pa	2	1	Time Series Oldest, ,
Loc48pa	2	1	Time Series Oldest, ,
Loc49pa	2	1	Time Series Oldest, ,
Loc50pa	2	1	Time Series Oldest, ,
Loc51pa	2	1	Time Series Oldest, ,
Loc52pa	2	1	Time Series Oldest, ,
Loc53pa	2	1	Time Series Oldest, ,
Loc54pa	2	1	Time Series Oldest, ,
Loc55pa	2	1	Time Series Oldest, ,
Loc56pa	2	1	Time Series Oldest, ,
Loc57pa	2	1	Time Series Oldest, ,
Loc58pa	2	1	Time Series Oldest, ,
Loc59pa	2	1	Time Series Oldest, ,
Loc60pa	2	1	Time Series Oldest, ,
Loc61pa	1	1	Time Series Oldest, ,
Loc62pa	1	1	Time Series Oldest, ,
Loc63pa	1	1	Time Series Oldest, ,
Loc64pa	1	1	Time Series Oldest, ,
Loc65pa	1	1	Time Series Oldest, ,
Loc66pa	1	1	Time Series Oldest, ,
Loc67pa	1	1	Time Series Oldest, ,
Loc68pa	1	1	Time Series Oldest, ,
Loc69pa	1	1	Time Series Oldest, ,
Loc70pa	1	1	Time Series Oldest, ,
Loc71pa	1	1	Time Series Oldest, ,
Loc72pa	1	1	Time Series Oldest, ,
Loc73pa	1	1	Time Series Oldest, ,
Loc74pa	1	1	Time Series Oldest, ,
Loc75pa	1	1	Time Series Oldest, ,
Loc76pa	1	1	Time Series Oldest, ,
Loc77pa	1	1	Time Series Oldest, ,
Loc78pa	1	1	Time Series Oldest, ,
Loc79pa	1	1	Time Series Oldest, ,
Loc80pa	inf	1	Time Series Oldest, ,
Loc1sb	inf	1	Time Series Oldest, ,
Loc2sb	2	1	Time Series Oldest, ,
Loc3sb	2	1	Time Series Oldest, ,

Loc4sb	2	1	Time Series Oldest, ,
Loc5sb	2	1	Time Series Oldest, ,
Loc6sb	2	1	Time Series Oldest, ,
Loc7sb	2	1	Time Series Oldest, ,
Loc8sb	2	1	Time Series Oldest, ,
Loc9sb	2	1	Time Series Oldest, ,
Loc10sb	2	1	Time Series Oldest, ,
Loc11sb	2	1	Time Series Oldest, ,
Loc12sb	2	1	Time Series Oldest, ,
Loc13sb	2	1	Time Series Oldest, ,
Loc14sb	2	1	Time Series Oldest, ,
Loc15sb	2	1	Time Series Oldest, ,
Loc16sb	2	1	Time Series Oldest, ,
Loc17sb	2	1	Time Series Oldest, ,
Loc18sb	2	1	Time Series Oldest, ,
Loc19sb	2	1	Time Series Oldest, ,
Loc20sb	2	1	Time Series Oldest, ,
Loc21sb	inf	1	Time Series Oldest, ,
Loc22sb	2	1	Time Series Oldest, ,
Loc23sb	2	1	Time Series Oldest, ,
Loc24sb	2	1	Time Series Oldest, ,
Loc25sb	2	1	Time Series Oldest, ,
Loc26sb	2	1	Time Series Oldest, ,
Loc27sb	2	1	Time Series Oldest, ,
Loc28sb	2	1	Time Series Oldest, ,
Loc29sb	2	1	Time Series Oldest, ,
Loc30sb	2	1	Time Series Oldest, ,
Loc31sb	2	1	Time Series Oldest, ,
Loc32sb	2	1	Time Series Oldest, ,
Loc33sb	2	1	Time Series Oldest, ,
Loc34sb	2	1	Time Series Oldest, ,
Loc35sb	2	1	Time Series Oldest, ,
Loc36sb	2	1	Time Series Oldest, ,
Loc37sb	2	1	Time Series Oldest, ,
Loc38sb	2	1	Time Series Oldest, ,
Loc39sb	2	1	Time Series Oldest, ,
Loc40sb	2	1	Time Series Oldest, ,
Loc1sh	2	1	Time Series Oldest, ,
Loc2sh	2	1	Time Series Oldest, ,
Loc3sh	2	1	Time Series Oldest, ,
Loc4sh	2	1	Time Series Oldest, ,
Loc5sh	2	1	Time Series Oldest, ,
Loc6sh	2	1	Time Series Oldest, ,
Loc7sh	2	1	Time Series Oldest, ,
Loc8sh	2	1	Time Series Oldest, ,
Loc9sh	2	1	Time Series Oldest, ,
Loc10sh	2	1	Time Series Oldest, ,

Loc11sh 2 1 Time Series Oldest, ,
Loc12sh 2 1 Time Series Oldest, ,
Loc13sh 2 1 Time Series Oldest, ,
Loc14sh 2 1 Time Series Oldest, ,
Loc15sh 2 1 Time Series Oldest, ,
Loc16sh 2 1 Time Series Oldest, ,
Loc17sh 2 1 Time Series Oldest, ,
Loc18sh 2 1 Time Series Oldest, ,
Loc19sh 2 1 Time Series Oldest, ,
Loc20sh 2 1 Time Series Oldest, ,
Loc21sh 2 1 Time Series Oldest, ,
Loc22sh 2 1 Time Series Oldest, ,
Loc23sh 2 1 Time Series Oldest, ,
Loc24sh 2 1 Time Series Oldest, ,
Loc25sh 2 1 Time Series Oldest, ,
Loc26sh 2 1 Time Series Oldest, ,
Loc27sh 2 1 Time Series Oldest, ,
Loc28sh 2 1 Time Series Oldest, ,
Loc29sh 2 1 Time Series Oldest, ,
Loc30sh 2 1 Time Series Oldest, ,
Loc31sh 2 1 Time Series Oldest, ,
Loc32sh 2 1 Time Series Oldest, ,
Loc33sh 2 1 Time Series Oldest, ,
Loc34sh 2 1 Time Series Oldest, ,
Loc35sh 2 1 Time Series Oldest, ,
Loc36sh 2 1 Time Series Oldest, ,
Loc37sh 2 1 Time Series Oldest, ,
Loc38sh 2 1 Time Series Oldest, ,
Loc39sh 2 1 Time Series Oldest, ,
Loc40sh 2 1 Time Series Oldest, ,
Loc1sa 2 1 Time Series Oldest, ,
Loc2sa 2 1 Time Series Oldest, ,
Loc3sa 2 1 Time Series Oldest, ,
Loc4sa 2 1 Time Series Oldest, ,
Loc5sa 2 1 Time Series Oldest, ,
Loc6sa 2 1 Time Series Oldest, ,
Loc7sa 2 1 Time Series Oldest, ,
Loc8sa 2 1 Time Series Oldest, ,
Loc9sa 2 1 Time Series Oldest, ,
Loc10sa 2 1 Time Series Oldest, ,
Loc11sa 2 1 Time Series Oldest, ,
Loc12sa 2 1 Time Series Oldest, ,
Loc13sa 2 1 Time Series Oldest, ,
Loc14sa 2 1 Time Series Oldest, ,
Loc15sa 2 1 Time Series Oldest, ,
Loc16sa 2 1 Time Series Oldest, ,
Loc17sa 2 1 Time Series Oldest, ,

Loc18sa	2	1	Time Series Oldest, ,
Loc19sa	2	1	Time Series Oldest, ,
Loc20sa	2	1	Time Series Oldest, ,
Loc21sa	inf	1	Time Series Oldest, ,
Loc22sa	2	1	Time Series Oldest, ,
Loc23sa	2	1	Time Series Oldest, ,
Loc24sa	2	1	Time Series Oldest, ,
Loc25sa	2	1	Time Series Oldest, ,
Loc26sa	2	1	Time Series Oldest, ,
Loc27sa	2	1	Time Series Oldest, ,
Loc28sa	2	1	Time Series Oldest, ,
Loc29sa	2	1	Time Series Oldest, ,
Loc30sa	2	1	Time Series Oldest, ,
Loc31sa	2	1	Time Series Oldest, ,
Loc32sa	2	1	Time Series Oldest, ,
Loc33sa	2	1	Time Series Oldest, ,
Loc34sa	2	1	Time Series Oldest, ,
Loc35sa	2	1	Time Series Oldest, ,
Loc36sa	2	1	Time Series Oldest, ,
Loc37sa	2	1	Time Series Oldest, ,
Loc38sa	2	1	Time Series Oldest, ,
Loc39sa	2	1	Time Series Oldest, ,
Loc40sa	2	1	Time Series Oldest, ,
Loc41sa	1	1	Time Series Oldest, ,
Loc42sa	1	1	Time Series Oldest, ,
Loc43sa	1	1	Time Series Oldest, ,
Loc44sa	1	1	Time Series Oldest, ,
Loc45sa	1	1	Time Series Oldest, ,
Loc46sa	1	1	Time Series Oldest, ,
Loc47sa	1	1	Time Series Oldest, ,
Loc48sa	1	1	Time Series Oldest, ,
Loc49sa	1	1	Time Series Oldest, ,
Loc50sa	1	1	Time Series Oldest, ,
Loc51sa	1	1	Time Series Oldest, ,
Loc52sa	1	1	Time Series Oldest, ,
Loc53sa	1	1	Time Series Oldest, ,
Loc54sa	1	1	Time Series Oldest, ,
Loc55sa	1	1	Time Series Oldest, ,
Loc56sa	1	1	Time Series Oldest, ,
Loc57sa	1	1	Time Series Oldest, ,
Loc58sa	1	1	Time Series Oldest, ,
Loc59sa	1	1	Time Series Oldest, ,
Loc60sa	inf	1	Time Series Oldest, ,

* Entities *

Name	Speed (mpm)	Stats	Cost
Car	1140	Time Series	
Ev	1140	Time Series	

* Processing *

Process		Routing	
Entity	Location	Operation	
		Blk Output	Destination Rule Move Logic
ALL	Loc1pb	IF Entity()=Ev THEN BEGIN v_presencapb=1 END	1 ALL Loc2pb FIRST 1 sub_accel
Car	Loc2pb		1 Car Loc3pb FIRST 1 sub_accel
Ev	Loc2pb	DISPLAY "Acionado sistema de priorização de rota! Cronometro ativado!" v_tempo_inicial=CLOCK()	1 Ev Loc3pb FIRST 1 sub_accel
ALL	Loc3pb		1 ALL Loc4pb FIRST 1 sub_accel
ALL	Loc4pb		1 ALL Loc5pb FIRST 1 sub_accel
ALL	Loc5pb		1 ALL Loc6pb FIRST 1 sub_accel
ALL	Loc6pb		1 ALL Loc7pb FIRST 1 sub_accel
ALL	Loc7pb		1 ALL Loc8pb FIRST 1 sub_accel
ALL	Loc8pb		1 ALL Loc9pb FIRST 1 sub_accel
ALL	Loc9pb		1 ALL Loc10pb FIRST 1 sub_accel
ALL	Loc10pb	WAIT UNTIL v_semaforo_horizontalpb=1 sub_quadrante(1,2)	1 ALL Loc11pb FIRST 1 a_sentido=1
Car	Loc11pb	1 Car v_presencapb=0 v_presencaph=1	Loc12pb FIRST 1 sub_accel
Ev	Loc11pb		1 Ev Loc12pb FIRST 1 sub_accel
ALL	Loc12pb		1 ALL Loc13pb FIRST 1 sub_accel

Ev	Loc13pb	1	Ev	Loc14pb	FIRST 1 sub_accel			
ALL	Loc13pb	1	ALL	Loc14pb	FIRST 1 sub_accel			
ALL	Loc14pb	1	ALL	Loc15pb	FIRST 1 sub_accel			
ALL	Loc15pb	1	ALL	Loc16pb	FIRST 1 sub_accel			
ALL	Loc16pb	1	ALL	Loc17pb	FIRST 1 sub_accel			
ALL	Loc17pb	1	ALL	Loc18pb	FIRST 1 sub_accel			
ALL	Loc18pb	1	ALL	Loc19pb	FIRST 1 sub_accel			
Ev	Loc19pb	1	Ev	Loc20pb	FIRST 1 sub_accel			
Car	Loc19pb	1	Car	Loc20pb	FIRST 1 sub_accel			
ALL	Loc21pb	1	ALL	Loc22pb	FIRST 1 sub_accel			
ALL	Loc22pb	1	ALL	Loc23pb	FIRST 1 sub_accel			
ALL	Loc23pb	1	ALL	Loc24pb	FIRST 1 sub_accel			
ALL	Loc24pb	1	ALL	Loc25pb	FIRST 1 sub_accel			
ALL	Loc25pb	1	ALL	Loc26pb	FIRST 1 sub_accel			
ALL	Loc26pb	1	ALL	Loc27pb	FIRST 1 sub_accel			
ALL	Loc27pb	1	ALL	Loc28pb	FIRST 1 sub_accel			
ALL	Loc28pb	1	ALL	Loc29pb	FIRST 1 sub_accel			
ALL	Loc29pb	1	ALL	Loc30pb	FIRST 1 sub_accel			
Ev	Loc30pb	WAIT UNTIL v_semaforo_horizontalpb=1						
		1	Ev	Loc31pb	FIRST 1 a_sentido=1			
					sub_accel			
Car	Loc30pb	WAIT UNTIL v_semaforo_horizontalpb=1						
		IF (RAND(1)<0.8) THEN BEGIN						
			sub_quadrante(1,1)					
			ROUTE 1					
			END					
			ELSE BEGIN					
				sub_quadrante(1,1)				
				ROUTE 2				
			END					
			1	Car	Loc31pb	FIRST 1 a_sentido=1		
			2	Car	Loc41pb	FIRST 1 a_sentido=2		sub_accel
								sub_accel
ALL	Loc31pb	1	ALL	Loc32pb	FIRST 1 sub_accel			
ALL	Loc32pb	1	ALL	Loc33pb	FIRST 1 sub_accel			
ALL	Loc33pb	1	ALL	Loc34pb	FIRST 1 sub_accel			
ALL	Loc34pb	1	ALL	Loc35pb	FIRST 1 sub_accel			
ALL	Loc35pb	1	ALL	Loc36pb	FIRST 1 sub_accel			
ALL	Loc36pb	1	ALL	Loc37pb	FIRST 1 sub_accel			
ALL	Loc37pb	1	ALL	Loc38pb	FIRST 1 sub_accel			
ALL	Loc38pb	1	ALL	Loc39pb	FIRST 1 sub_accel			
ALL	Loc39pb	1	ALL	Loc40pb	FIRST 1 sub_accel			
Car	Loc60pb	1	Car	Loc59pb	FIRST 1 sub_accel			

Car	Loc59pb	1	Car	Loc58pb	FIRST 1 sub_accel
Car	Loc58pb	1	Car	Loc57pb	FIRST 1 sub_accel
Car	Loc57pb	1	Car	Loc56pb	FIRST 1 sub_accel
Car	Loc56pb	1	Car	Loc55pb	FIRST 1 sub_accel
Car	Loc55pb	1	Car	Loc54pb	FIRST 1 sub_accel
Car	Loc54pb	1	Car	Loc53pb	FIRST 1 sub_accel
Car	Loc53pb	1	Car	Loc52pb	FIRST 1 sub_accel
Car	Loc52pb	1	Car	Loc51pb	FIRST 1 sub_accel
Car	Loc51pb	WAIT UNTIL v_semaforo_verticalpb=1			

```

real x = RAND(1)
IF (x<0.3) THEN BEGIN
    sub_quadrante(1,1)
    ROUTE 1
END

```

```

ELSE IF (x<0.6) THEN BEGIN
    sub_quadrante(1,2)
    ROUTE 2
END

```

```

ELSE BEGIN
    sub_quadrante(1,1)
    ROUTE 3
END

```

1 Car Loc41pb FIRST 1 a_sentido=2

2 Car Loc11pb FIRST 1 a_sentido=1
sub_accel

3 Car Loc31pb FIRST 1 a_sentido=1
sub_accel

Car	Loc41pb	1	Car	Loc42pb	FIRST 1 sub_accel
Car	Loc42pb	1	Car	Loc43pb	FIRST 1 sub_accel
Car	Loc43pb	1	Car	Loc44pb	FIRST 1 sub_accel
Car	Loc44pb	1	Car	Loc45pb	FIRST 1 sub_accel
Car	Loc45pb	1	Car	Loc46pb	FIRST 1 sub_accel
Car	Loc46pb	1	Car	Loc47pb	FIRST 1 sub_accel
Car	Loc47pb	1	Car	Loc48pb	FIRST 1 sub_accel
Car	Loc48pb	1	Car	Loc49pb	FIRST 1 sub_accel
Car	Loc49pb	1	Car	Loc50pb	FIRST 1 sub_accel
ALL	Loc1ph	1	ALL	Loc2ph	FIRST 1 sub_accel
ALL	Loc2ph	1	ALL	Loc3ph	FIRST 1 sub_accel
ALL	Loc3ph	1	ALL	Loc4ph	FIRST 1 sub_accel
ALL	Loc4ph	1	ALL	Loc5ph	FIRST 1 sub_accel

ALL	Loc5ph	1	ALL	Loc6ph	FIRST 1 sub_accel
ALL	Loc6ph	1	ALL	Loc7ph	FIRST 1 sub_accel
ALL	Loc7ph	1	ALL	Loc8ph	FIRST 1 sub_accel
ALL	Loc8ph	1	ALL	Loc9ph	FIRST 1 sub_accel
ALL	Loc9ph	1	ALL	Loc10ph	FIRST 1 sub_accel
Ev	Loc10ph	WAIT UNTIL v_semaforo_horizontalph=1 sub_quadrante(1,3)			
		1	Ev	Loc11ph	FIRST 1 a_sentido=1
					sub_accel
Car	Loc10ph	WAIT UNTIL v_semaforo_horizontalph=1 IF (RAND(1)<0.8) THEN BEGIN sub_quadrante(1,3) ROUTE 1 END			
					ELSE BEGIN
					sub_quadrante(2,2)
					ROUTE 2
					END
			1	Car	Loc11ph FIRST 1 a_sentido=1
					sub_accel
		2	Car	Loc51ph	FIRST 1 a_sentido=2
					sub_accel
Car	Loc11ph	1	Car	Loc12ph	FIRST 1 sub_accel
Ev	Loc11ph v_presencaph=0 v_presencapa=1	1	Ev	Loc12ph	FIRST 1 sub_accel
ALL	Loc12ph	1	ALL	Loc13ph	FIRST 1 sub_accel
ALL	Loc13ph	1	ALL	Loc14ph	FIRST 1 sub_accel
ALL	Loc14ph	1	ALL	Loc15ph	FIRST 1 sub_accel
ALL	Loc15ph	1	ALL	Loc16ph	FIRST 1 sub_accel
ALL	Loc16ph	1	ALL	Loc17ph	FIRST 1 sub_accel
ALL	Loc17ph	1	ALL	Loc18ph	FIRST 1 sub_accel
ALL	Loc18ph	1	ALL	Loc19ph	FIRST 1 sub_accel
Ev	Loc19ph	1	Ev	Loc20ph	FIRST 1 sub_accel
Car	Loc19ph	1	Car	Loc20ph	FIRST 1 sub_accel
ALL	Loc21ph	1	ALL	Loc22ph	FIRST 1 sub_accel
ALL	Loc22ph	1	ALL	Loc23ph	FIRST 1 sub_accel
ALL	Loc23ph	1	ALL	Loc24ph	FIRST 1 sub_accel
ALL	Loc24ph	1	ALL	Loc25ph	FIRST 1 sub_accel
ALL	Loc25ph	1	ALL	Loc26ph	FIRST 1 sub_accel
ALL	Loc26ph	1	ALL	Loc27ph	FIRST 1 sub_accel
ALL	Loc27ph	1	ALL	Loc28ph	FIRST 1 sub_accel
ALL	Loc28ph	1	ALL	Loc29ph	FIRST 1 sub_accel
ALL	Loc29ph	1	ALL	Loc30ph	FIRST 1 sub_accel

ALL Loc30ph WAIT UNTIL v_semaforo_horizontalph=1

sub_quadrante(1,1)

1 ALL Loc31ph FIRST 1 a_sentido=1

sub_accel

ALL Loc31ph	1 ALL Loc32ph FIRST 1 sub_accel
ALL Loc32ph	1 ALL Loc33ph FIRST 1 sub_accel
ALL Loc33ph	1 ALL Loc34ph FIRST 1 sub_accel
ALL Loc34ph	1 ALL Loc35ph FIRST 1 sub_accel
ALL Loc35ph	1 ALL Loc36ph FIRST 1 sub_accel
ALL Loc36ph	1 ALL Loc37ph FIRST 1 sub_accel
ALL Loc37ph	1 ALL Loc38ph FIRST 1 sub_accel
ALL Loc38ph	1 ALL Loc39ph FIRST 1 sub_accel
ALL Loc39ph	1 ALL Loc40ph FIRST 1 sub_accel
Car Loc50ph	1 Car Loc49ph FIRST 1 sub_accel
Car Loc49ph	1 Car Loc48ph FIRST 1 sub_accel
Car Loc48ph	1 Car Loc47ph FIRST 1 sub_accel
Car Loc47ph	1 Car Loc46ph FIRST 1 sub_accel
Car Loc46ph	1 Car Loc45ph FIRST 1 sub_accel
Car Loc45ph	1 Car Loc44ph FIRST 1 sub_accel
Car Loc44ph	1 Car Loc43ph FIRST 1 sub_accel
Car Loc43ph	1 Car Loc42ph FIRST 1 sub_accel
Car Loc42ph	1 Car Loc41ph FIRST 1 sub_accel
Car Loc41ph	WAIT UNTIL v_semaforo_verticalph=1

real x = RAND(1)

IF (x<0.3) THEN BEGIN
sub_quadrante(1,1)
ROUTE 1

END

ELSE IF (x<0.6) THEN BEGIN

sub_quadrante(2,2)
ROUTE 2
END

ELSE BEGIN

sub_quadrante(1,3)
ROUTE 3

END

1 Car Loc31ph FIRST 1 a_sentido=1

2 Car Loc51ph FIRST 1 a_sentido=2

sub_accel

3 Car Loc11ph FIRST 1 a_sentido=1
 Car Loc51ph 1 Car Loc52ph FIRST 1 sub_accel
 Car Loc52ph 1 Car Loc53ph FIRST 1 sub_accel
 Car Loc53ph 1 Car Loc54ph FIRST 1 sub_accel
 Car Loc54ph 1 Car Loc55ph FIRST 1 sub_accel
 Car Loc55ph 1 Car Loc56ph FIRST 1 sub_accel
 Car Loc56ph 1 Car Loc57ph FIRST 1 sub_accel
 Car Loc57ph 1 Car Loc58ph FIRST 1 sub_accel
 Car Loc58ph 1 Car Loc59ph FIRST 1 sub_accel
 Car Loc59ph 1 Car Loc60ph FIRST 1 sub_accel
 ALL Loc1pa 1 ALL Loc2pa FIRST 1 sub_accel
 ALL Loc2pa 1 ALL Loc3pa FIRST 1 sub_accel
 ALL Loc3pa 1 ALL Loc4pa FIRST 1 sub_accel
 ALL Loc4pa 1 ALL Loc5pa FIRST 1 sub_accel
 ALL Loc5pa 1 ALL Loc6pa FIRST 1 sub_accel
 ALL Loc6pa 1 ALL Loc7pa FIRST 1 sub_accel
 ALL Loc7pa 1 ALL Loc8pa FIRST 1 sub_accel
 ALL Loc8pa 1 ALL Loc9pa FIRST 1 sub_accel
 ALL Loc9pa 1 ALL Loc10pa FIRST 1 sub_accel
 Ev Loc10pa WAIT UNTIL v_semaforo_horizontalpa=1
 1 Ev Loc11pa FIRST 1 a_sentido=1

Car Loc10pa WAIT UNTIL v_semaforo_horizontalpa=1
 IF (RAND(1)<0.8) THEN BEGIN
 sub_quadrante(1,3)
 ROUTE 1
 END
 ELSE BEGIN
 sub_quadrante(2,3)
 ROUTE 2
 END
 1 Car Loc11pa FIRST 1 a_sentido=1
 2 Car Loc70pa FIRST 1 a_sentido=2

Car Loc11pa 1 Car Loc12pa FIRST 1 sub_accel
 Ev Loc11pa v_presencapa=0 1 Ev Loc12pa FIRST 1 sub_accel
 ALL Loc12pa 1 ALL Loc13pa FIRST 1 sub_accel
 ALL Loc13pa 1 ALL Loc14pa FIRST 1 sub_accel
 ALL Loc14pa 1 ALL Loc15pa FIRST 1 sub_accel
 ALL Loc15pa 1 ALL Loc16pa FIRST 1 sub_accel
 ALL Loc16pa 1 ALL Loc17pa FIRST 1 sub_accel
 ALL Loc17pa 1 ALL Loc18pa FIRST 1 sub_accel
 Car Loc18pa 1 Car Loc19pa FIRST 1 sub_accel

```

Ev Loc18pa v_tempo_final=clock()
    v_tempo_percorso=v_tempo_final-v_tempo_inicial
        1 Ev Loc19pa FIRST 1 sub_accel
ALL Loc19pa      1 ALL Loc20pa FIRST 1 sub_accel
ALL Loc21pa      1 ALL Loc22pa FIRST 1 sub_accel
ALL Loc22pa      1 ALL Loc23pa FIRST 1 sub_accel
ALL Loc23pa      1 ALL Loc24pa FIRST 1 sub_accel
ALL Loc24pa      1 ALL Loc25pa FIRST 1 sub_accel
ALL Loc25pa      1 ALL Loc26pa FIRST 1 sub_accel
ALL Loc26pa      1 ALL Loc27pa FIRST 1 sub_accel
ALL Loc27pa      1 ALL Loc28pa FIRST 1 sub_accel
ALL Loc28pa      1 ALL Loc29pa FIRST 1 sub_accel
ALL Loc29pa      1 ALL Loc30pa FIRST 1 sub_accel
ALL Loc30pa WAIT UNTIL v_semaforo_horizontalpa=1
    IF Car THEN BEGIN
        IF (RAND(1)<0.8) THEN BEGIN
            sub_quadrante(1,2)
            ROUTE 1
        END
        ELSE BEGIN
            sub_quadrante(1,3)
            ROUTE 2
        END
    END
    ELSE BEGIN
        sub_quadrante(1,2)
        ROUTE 1
    END
        1 ALL Loc31pa FIRST 1 a_sentido=1
            sub_accel
        2 Car Loc41pa FIRST 1 a_sentido=2
                sub_accel
ALL Loc31pa      1 ALL Loc32pa FIRST 1 sub_accel
ALL Loc32pa      1 ALL Loc33pa FIRST 1 sub_accel
ALL Loc33pa      1 ALL Loc34pa FIRST 1 sub_accel
ALL Loc34pa      1 ALL Loc35pa FIRST 1 sub_accel
ALL Loc35pa      1 ALL Loc36pa FIRST 1 sub_accel
ALL Loc36pa      1 ALL Loc37pa FIRST 1 sub_accel
ALL Loc37pa      1 ALL Loc38pa FIRST 1 sub_accel
ALL Loc38pa      1 ALL Loc39pa FIRST 1 sub_accel
ALL Loc39pa      1 ALL Loc40pa FIRST 1 sub_accel
Car Loc41pa      1 Car Loc42pa FIRST 1 sub_accel
Car Loc42pa      1 Car Loc43pa FIRST 1 sub_accel
Car Loc43pa      1 Car Loc44pa FIRST 1 sub_accel
Car Loc44pa      1 Car Loc45pa FIRST 1 sub_accel

```

```

Car Loc52pa           1 Car Loc51pa FIRST 1 sub_accel
Car Loc51pa WAIT UNTIL v_semaforo_verticalpa=1

IF (RAND(1)<0.4) THEN BEGIN
    sub_quadrante(1,3)
    ROUTE 1
END

ELSE BEGIN
    sub_quadrante(1,3)
    ROUTE 2
END           1 Car Loc41pa FIRST 1 a_sentido=2

                                sub_accel
2 Car Loc11pa FIRST 1 a_sentido=1

                                sub_accel
Car Loc1sb           1 Car Loc2sb FIRST 1 sub_accel
Car Loc2sb           1 Car Loc3sb FIRST 1 sub_accel
Car Loc3sb           1 Car Loc4sb FIRST 1 sub_accel
Car Loc4sb           1 Car Loc5sb FIRST 1 sub_accel
Car Loc5sb           1 Car Loc6sb FIRST 1 sub_accel
Car Loc6sb           1 Car Loc7sb FIRST 1 sub_accel
Car Loc7sb           1 Car Loc8sb FIRST 1 sub_accel
Car Loc8sb           1 Car Loc9sb FIRST 1 sub_accel
Car Loc9sb           1 Car Loc10sb FIRST 1 sub_accel
Car Loc10sb WAIT UNTIL v_semaforo_horizontalsb=1

IF (RAND(1)<0.8) THEN BEGIN
    sub_quadrante(2,2)
    ROUTE 1
END

ELSE BEGIN
    sub_quadrante(1,1)
    ROUTE 2
END           1 Car Loc11sb FIRST 1 a_sentido=1

                                sub_accel
2 Car Loc31sb FIRST 1 a_sentido=2

                                sub_accel
Car Loc11sb          1 Car Loc12sb FIRST 1 sub_accel
Car Loc12sb          1 Car Loc13sb FIRST 1 sub_accel
Car Loc13sb          1 Car Loc14sb FIRST 1 sub_accel
Car Loc14sb          1 Car Loc15sb FIRST 1 sub_accel
Car Loc15sb          1 Car Loc16sb FIRST 1 sub_accel

```

Car	Loc16sb	1	Car	Loc17sb	FIRST 1	sub_accel
Car	Loc17sb	1	Car	Loc18sb	FIRST 1	sub_accel
Car	Loc18sb	1	Car	Loc19sb	FIRST 1	sub_accel
Car	Loc19sb	1	Car	Loc20sb	FIRST 1	sub_accel
Car	Loc21sb	1	Car	Loc22sb	FIRST 1	sub_accel
Car	Loc22sb	1	Car	Loc23sb	FIRST 1	sub_accel
Car	Loc23sb	1	Car	Loc24sb	FIRST 1	sub_accel
Car	Loc24sb	1	Car	Loc25sb	FIRST 1	sub_accel
Car	Loc25sb	1	Car	Loc26sb	FIRST 1	sub_accel
Car	Loc26sb	1	Car	Loc27sb	FIRST 1	sub_accel
Car	Loc27sb	1	Car	Loc28sb	FIRST 1	sub_accel
Car	Loc28sb	1	Car	Loc29sb	FIRST 1	sub_accel
Car	Loc29sb	1	Car	Loc30sb	FIRST 1	sub_accel
Car	Loc30sb	WAIT UNTIL v_semaforo_verticalsb=1				

```

IF (RAND(1)<0.8) THEN BEGIN
    sub_quadrante(1,1)
    ROUTE 1
END

ELSE BEGIN
    sub_quadrante(2,2)
    ROUTE 2
END      1   Car   Loc31sb   FIRST 1 a_sentido=2

```

sub_accel

2	Car	Loc11sb	FIRST 1	a_sentido=1
---	-----	---------	---------	-------------

Car	Loc31sb	1	Car	Loc32sb	FIRST 1	sub_accel
Car	Loc32sb	1	Car	Loc33sb	FIRST 1	sub_accel
Car	Loc33sb	1	Car	Loc34sb	FIRST 1	sub_accel
Car	Loc34sb	1	Car	Loc35sb	FIRST 1	sub_accel
Car	Loc35sb	1	Car	Loc36sb	FIRST 1	sub_accel
Car	Loc36sb	1	Car	Loc37sb	FIRST 1	sub_accel
Car	Loc37sb	1	Car	Loc38sb	FIRST 1	sub_accel
Car	Loc38sb	1	Car	Loc39sb	FIRST 1	sub_accel
Car	Loc39sb	1	Car	Loc40sb	FIRST 1	sub_accel
Car	Loc1sh	1	Car	Loc2sh	FIRST 1	sub_accel
Car	Loc2sh	1	Car	Loc3sh	FIRST 1	sub_accel
Car	Loc3sh	1	Car	Loc4sh	FIRST 1	sub_accel
Car	Loc4sh	1	Car	Loc5sh	FIRST 1	sub_accel
Car	Loc5sh	1	Car	Loc6sh	FIRST 1	sub_accel
Car	Loc6sh	1	Car	Loc7sh	FIRST 1	sub_accel
Car	Loc7sh	1	Car	Loc8sh	FIRST 1	sub_accel
Car	Loc8sh	1	Car	Loc9sh	FIRST 1	sub_accel
Car	Loc9sh	1	Car	Loc10sh	FIRST 1	sub_accel

```

Car Loc10sh WAIT UNTIL v_semaforo_horizontalsh=1

    IF (RAND(1)<0.8) THEN BEGIN
        sub_quadrante(2,3)
        ROUTE 1
    END

    ELSE BEGIN
        sub_quadrante(2,2)
        ROUTE 2
    END      1 Car Loc11sh FIRST 1 a_sentido=1

                sub_accel
    2 Car Loc30sh FIRST 1 a_sentido=2

```

Car Loc11sh	1 Car Loc12sh FIRST 1 sub_accel
Car Loc12sh	1 Car Loc13sh FIRST 1 sub_accel
Car Loc13sh	1 Car Loc14sh FIRST 1 sub_accel
Car Loc14sh	1 Car Loc15sh FIRST 1 sub_accel
Car Loc15sh	1 Car Loc16sh FIRST 1 sub_accel
Car Loc16sh	1 Car Loc17sh FIRST 1 sub_accel
Car Loc17sh	1 Car Loc18sh FIRST 1 sub_accel
Car Loc18sh	1 Car Loc19sh FIRST 1 sub_accel
Car Loc19sh	1 Car Loc20sh FIRST 1 sub_accel
Car Loc40sh	1 Car Loc39sh FIRST 1 sub_accel
Car Loc39sh	1 Car Loc38sh FIRST 1 sub_accel
Car Loc37sh	1 Car Loc36sh FIRST 1 sub_accel
Car Loc38sh	1 Car Loc37sh FIRST 1 sub_accel
Car Loc36sh	1 Car Loc35sh FIRST 1 sub_accel
Car Loc35sh	1 Car Loc34sh FIRST 1 sub_accel
Car Loc34sh	1 Car Loc33sh FIRST 1 sub_accel
Car Loc33sh	1 Car Loc32sh FIRST 1 sub_accel
Car Loc32sh	1 Car Loc31sh FIRST 1 sub_accel

Car Loc31sh WAIT UNTIL v_semaforo_verticalsh=1

```

    IF (RAND(1)<0.8) THEN BEGIN
        sub_quadrante(2,2)
        ROUTE 1
    END

    ELSE BEGIN
        sub_quadrante(2,3)
        ROUTE 2
    END      1 Car Loc30sh FIRST 1 a_sentido=2

                sub_accel

```

2 Car Loc11sh FIRST 1 a_sentido=1

			sub_accel
Car	Loc30sh	1 Car	Loc29sh FIRST 1 sub_accel
Car	Loc29sh	1 Car	Loc28sh FIRST 1 sub_accel
Car	Loc28sh	1 Car	Loc27sh FIRST 1 sub_accel
Car	Loc27sh	1 Car	Loc26sh FIRST 1 sub_accel
Car	Loc26sh	1 Car	Loc25sh FIRST 1 sub_accel
Car	Loc25sh	1 Car	Loc24sh FIRST 1 sub_accel
Car	Loc24sh	1 Car	Loc23sh FIRST 1 sub_accel
Car	Loc23sh	1 Car	Loc22sh FIRST 1 sub_accel
Car	Loc22sh	1 Car	Loc21sh FIRST 1 sub_accel
Car	Loc1sa	1 Car	Loc2sa FIRST 1 sub_accel
Car	Loc2sa	1 Car	Loc3sa FIRST 1 sub_accel
Car	Loc3sa	1 Car	Loc4sa FIRST 1 sub_accel
Car	Loc4sa	1 Car	Loc5sa FIRST 1 sub_accel
Car	Loc5sa	1 Car	Loc6sa FIRST 1 sub_accel
Car	Loc6sa	1 Car	Loc7sa FIRST 1 sub_accel
Car	Loc7sa	1 Car	Loc8sa FIRST 1 sub_accel
Car	Loc8sa	1 Car	Loc9sa FIRST 1 sub_accel
Car	Loc9sa	1 Car	Loc10sa FIRST 1 sub_accel
Car	Loc10sa	WAIT UNTIL v_semaforo_horizontalsa=1	

```

real x = RAND(1)
IF (x<0.3) THEN BEGIN
    sub_quadrante(2,3)
    ROUTE 1
END

ELSE IF (x<0.6) THEN BEGIN
    sub_quadrante(2,3)
    ROUTE 2
END

ELSE BEGIN
    sub_quadrante(1,3)
    ROUTE 3
END
    1 Car Loc11sa FIRST 1 a_sentido=1
        sub_accel
    2 Car Loc51sa FIRST 1 a_sentido=2
        sub_accel
    3 Car Loc31sa FIRST 1 a_sentido=2
        sub_accel

```

Car	Loc11sa	1	Car	Loc12sa	FIRST 1	sub_accel
Car	Loc12sa	1	Car	Loc13sa	FIRST 1	sub_accel
Car	Loc13sa	1	Car	Loc14sa	FIRST 1	sub_accel
Car	Loc14sa	1	Car	Loc15sa	FIRST 1	sub_accel
Car	Loc15sa	1	Car	Loc16sa	FIRST 1	sub_accel
Car	Loc16sa	1	Car	Loc17sa	FIRST 1	sub_accel
Car	Loc17sa	1	Car	Loc18sa	FIRST 1	sub_accel
Car	Loc18sa	1	Car	Loc19sa	FIRST 1	sub_accel
Car	Loc19sa	1	Car	Loc20sa	FIRST 1	sub_accel
Car	Loc21sa	1	Car	Loc22sa	FIRST 1	sub_accel
Car	Loc22sa	1	Car	Loc23sa	FIRST 1	sub_accel
Car	Loc23sa	1	Car	Loc24sa	FIRST 1	sub_accel
Car	Loc24sa	1	Car	Loc25sa	FIRST 1	sub_accel
Car	Loc25sa	1	Car	Loc26sa	FIRST 1	sub_accel
Car	Loc26sa	1	Car	Loc27sa	FIRST 1	sub_accel
Car	Loc27sa	1	Car	Loc28sa	FIRST 1	sub_accel
Car	Loc28sa	1	Car	Loc29sa	FIRST 1	sub_accel
Car	Loc29sa	1	Car	Loc30sa	FIRST 1	sub_accel
Car	Loc30sa	WAIT UNTIL v_semaforo_verticalsa=1				

IF (RAND(1)<0.8) THEN BEGIN
 sub_quadrante(1,3)
 ROUTE 1
 END

ELSE BEGIN
 sub_quadrante(2,3)
 ROUTE 2
 END 1 Car Loc31sa FIRST 1 a_sentido=2

2 Car Loc11sa FIRST 1 a_sentido=1 sub_accel

Car	Loc31sa	1	Car	Loc32sa	FIRST 1	sub_accel
Car	Loc32sa	1	Car	Loc33sa	FIRST 1	sub_accel
Car	Loc33sa	1	Car	Loc34sa	FIRST 1	sub_accel
Car	Loc34sa	1	Car	Loc35sa	FIRST 1	sub_accel
Car	Loc35sa	1	Car	Loc36sa	FIRST 1	sub_accel
Car	Loc36sa	1	Car	Loc37sa	FIRST 1	sub_accel
Car	Loc37sa	1	Car	Loc38sa	FIRST 1	sub_accel
Car	Loc38sa	1	Car	Loc39sa	FIRST 1	sub_accel
Car	Loc39sa	1	Car	Loc40sa	FIRST 1	sub_accel
Car	Loc41sa	1	Car	Loc42sa	FIRST 1	sub_accel
Car	Loc42sa	1	Car	Loc43sa	FIRST 1	sub_accel
Car	Loc43sa	1	Car	Loc44sa	FIRST 1	sub_accel
Car	Loc44sa	1	Car	Loc45sa	FIRST 1	sub_accel

Car	Loc45sa	1	Car	Loc46sa	FIRST 1	sub_accel
Car	Loc46sa	1	Car	Loc47sa	FIRST 1	sub_accel
Car	Loc47sa	1	Car	Loc48sa	FIRST 1	sub_accel
Car	Loc48sa	1	Car	Loc49sa	FIRST 1	sub_accel
Car	Loc49sa	1	Car	Loc50sa	FIRST 1	sub_accel
Car	Loc50sa	WAIT UNTIL v_semaforo_verticalsa=1				

sub_quadrante(2,3)

1	Car	Loc51sa	FIRST 1	a_sentido=2
---	-----	---------	---------	-------------

sub_accel						
Car	Loc51sa	1	Car	Loc52sa	FIRST 1	sub_accel
Car	Loc52sa	1	Car	Loc53sa	FIRST 1	sub_accel
Car	Loc53sa	1	Car	Loc54sa	FIRST 1	sub_accel
Car	Loc54sa	1	Car	Loc55sa	FIRST 1	sub_accel
Car	Loc55sa	1	Car	Loc56sa	FIRST 1	sub_accel
Car	Loc56sa	1	Car	Loc57sa	FIRST 1	sub_accel
Car	Loc57sa	1	Car	Loc58sa	FIRST 1	sub_accel
Car	Loc58sa	1	Car	Loc59sa	FIRST 1	sub_accel
Car	Loc59sa	1	Car	Loc60sa	FIRST 1	sub_accel
ALL	Loc40ph	1	ALL	Loc21pb	FIRST 1	sub_accel
ALL	Loc20pb	1	ALL	Loc1ph	FIRST 1	sub_accel
ALL	Loc40pa	1	ALL	Loc21ph	FIRST 1	sub_accel
ALL	Loc20ph	1	ALL	Loc1pa	FIRST 1	sub_accel
ALL	Loc40sb	1	ALL	Loc60pb	FIRST 1	sub_accel
ALL	Loc20sb	1	ALL	Loc1sh	FIRST 1	sub_accel
ALL	Loc60ph	1	ALL	Loc40sh	FIRST 1	sub_accel
ALL	Loc20sh	1	ALL	Loc1sa	FIRST 1	sub_accel
ALL	Loc61pa	1	ALL	Loc41sa	FIRST 1	sub_accel
ALL	Loc40sa	1	ALL	Loc60pa	FIRST 1	sub_accel
ALL	Loc40pb	1	ALL	EXIT	FIRST 1	sub_accel
ALL	Loc50pb	1	ALL	EXIT	FIRST 1	sub_accel
ALL	Loc21sh	1	ALL	EXIT	FIRST 1	sub_accel
ALL	Loc60sa	1	ALL	EXIT	FIRST 1	sub_accel
ALL	Loc20sa	1	ALL	EXIT	FIRST 1	sub_accel
ALL	Loc20pa	1	ALL	EXIT	FIRST 1	sub_accel
ALL	Loc50pa	1	ALL	EXIT	FIRST 1	sub_accel

* Arrivals *

Entity	Location	Qty each	First Time Occurrences	Frequency	Logic
--------	----------	----------	------------------------	-----------	-------

Car	Loc1pb	1; paulista_cb 0	inf	a_quadrante_x=1
				a_quadrante_y=1
				a_sentido=1
GRAPHIC 1				
Ev	Loc1pb	1	1830 sec	1
				a_quadrante_x=1
				a_quadrante_y=1
				a_sentido=1
Car	Loc21pa	1; paulista_bc 0	inf	a_quadrante_x=1
				a_quadrante_y=3
				a_sentido=1
GRAPHIC 2				
Car	Loc1sb	1; santos 0	inf	a_quadrante_x=2
				a_quadrante_y=1
				a_sentido=1
GRAPHIC 3				
Car	Loc21sb	1; bela_cinta 0	inf	a_quadrante_x=2
				a_quadrante_y=1
				a_sentido=2
GRAPHIC 4				
Car	Loc50ph	1; haddock_lobo 0	inf	a_quadrante_x=1
				a_quadrante_y=2
				a_sentido=2
GRAPHIC 5				
Car	Loc80pa	1; augusta_cb 0	inf	a_quadrante_x=1
				a_quadrante_y=3
				a_sentido=2
GRAPHIC 1				
Car	Loc21sa	1; augusta_bc 0	inf	a_quadrante_x=2
				a_quadrante_y=3
				a_sentido=2

GRAPHIC 2

* **Attributes** *

ID	Type	Classification
a_distperc	Integer	Entity
a_velocidade	Real	Entity
#		
#sentido em que o carro esta se deslocando, horizontal=1, vertical=2		
a_sentido	Integer	Entity
a_quadrante_x	Integer	Entity
a_quadrante_y	Integer	Entity

* Variables (global) *

ID	Type	Initial value	Stats
v_semaforo_horizontalpb	Integer	1	Time Series
v_semaforo_verticalpb	Integer	0	Time Series
v_semaforo_horizontalph	Integer	1	Time Series
v_semaforo_verticalph	Integer	0	Time Series
v_semaforo_horizontalpa	Integer	1	Time Series
v_semaforo_verticalpa	Integer	0	Time Series
v_semaforo_horizontalsb	Integer	1	Time Series
v_semaforo_verticalsb	Integer	0	Time Series
v_semaforo_horizontalsh	Integer	1	Time Series
v_semaforo_verticalsh	Integer	0	Time Series
v_semaforo_horizontalsa	Integer	1	Time Series
v_semaforo_verticalsa	Integer	0	Time Series
v_tempo_inicial	Integer	0	Time Series
v_tempo_final	Integer	0	Time Series
v_estadopb	Integer	0	Time Series
v_presencapb	Integer	0	Time Series
v_presencapa	Integer	0	Time Series
v_presencaph	Integer	0	Time Series
v_tempo_percurso	Integer	0	Time Series
v_pb	Integer	0	Time Series
v_ph	Integer	0	Time Series
v_pa	Integer	0	Time Series

* Arrays *

ID	Dimensions	Type
m_semaforo	2,3,2	Integer

* Macros *

ID	Text
VM	19
AC	1.9

* Subroutines *

ID	Type	Parameter	Type	Logic
sub_accel	None			IF a_velocidade<VM THEN BEGIN INC a_distperc, 6 a_velocidade=SQRT(2*AC*a_distperc) WAIT (6/a_velocidade-6/VM) sec END
m_semaforo	[a_quadrante_x,a_quadrante_y,a_sentido]	=0		THEN BEGIN a_distperc=0 a_velocidade=0 END
sub_quadrante	None	x	Integer	a_quadrante_x=x
		y	Integer	a_quadrante_y=y
sub_semaforop	None			inicio: WHILE v_pb<75 DO { v_semaforo_horizontalpb=1 v_semaforo_verticalpb=0 WAIT 1 sec}

```

v_pb=v_pb +1
IF v_presencapb=1 THEN BEGIN
    v_semaforo_horizontalpb=1
    v_semaforo_verticalpb=0
    WHILE v_pb<100 DO {
        v_semaforo_horizontalpb=1
        v_semaforo_verticalpb=0
        WAIT 1 sec
        v_pb=v_pb +1
    }
    GOTO saida1
END
}

saida1:
v_pb=0

WHILE v_pb<3 DO {
    v_semaforo_horizontalpb=2
    v_semaforo_verticalpb=0
    WAIT 1 sec
    v_pb=v_pb +1
    IF v_presencapb=1 THEN BEGIN
        v_semaforo_horizontalpb=2
        v_semaforo_verticalpb=0
        WAIT 3 sec
        GOTO saida2
    END
}
v_pb=0
saida2:

WHILE v_pb<2 DO {
    v_semaforo_horizontalpb=0
    v_semaforo_verticalpb=0
    WAIT 1 sec
    v_pb=v_pb +1
    IF v_presencapb=1 THEN BEGIN
        v_semaforo_horizontalpb=0
        v_semaforo_verticalpb=0
        WAIT 2 sec
        GOTO saida3
    END
}
v_pb=0
saida3:

```

```

WHILE v_pb<55 DO {
    v_semaforo_horizontalpb=0
    v_semaforo_verticalpb=1
    WAIT 1 sec
    v_pb=v_pb +1
    IF v_presencapb=1 AND v_pb>=25 THEN BEGIN
        GOTO saida4
    END
    IF v_presencapb=1 AND v_pb<25 THEN BEGIN
        WHILE v_pb<25 DO {
            v_semaforo_horizontalpb=0
            v_semaforo_verticalpb=1
            WAIT 1 sec
            v_pb=v_pb +1
        }
        GOTO saida4
    END
}
v_pb=0
saida4:

WHILE v_pb<3 DO {
    v_semaforo_horizontalpb=0
    v_semaforo_verticalpb=2
    WAIT 1 sec
    v_pb=v_pb +1
    IF v_presencapb=1 THEN BEGIN
        v_semaforo_horizontalpb=0
        v_semaforo_verticalpb=2
        WAIT 3 sec
        GOTO saida5
    END
}
v_pb=0
saida5:

WHILE v_pb<2 DO {
    v_semaforo_horizontalpb=0
    v_semaforo_verticalpb=0
    wait 1 sec
    v_pb=v_pb +1
    IF v_presencapb=1 THEN BEGIN
        v_semaforo_horizontalpb=0
        v_semaforo_verticalpb=0
        WAIT 2 sec
        GOTO saida6
    END
}

```

```

    }
v_pb=0
saida6:

goto inicio

sub_semaforoph None           inicio:

WHILE v_ph<80 DO {
    v_semaforo_horizontalph=1
    v_semaforo_verticalph=0
    WAIT 1 sec
    v_ph=v_ph +1
    IF v_presencaph=1 THEN BEGIN
        v_semaforo_horizontalph=1
        v_semaforo_verticalph=0
        WHILE v_ph<100 DO {
            v_semaforo_horizontalph=1
            v_semaforo_verticalph=0
            WAIT 1 sec
            v_ph=v_ph +1
        }
        //      WAIT UNTIL v_presencaph=0
        GOTO saida1
    END
}

saida1:
v_ph=0

WHILE v_ph<3 DO {
    v_semaforo_horizontalph=2
    v_semaforo_verticalph=0
    WAIT 1 sec
    v_ph=v_ph +1
    IF v_presencaph=1 THEN BEGIN
        v_semaforo_horizontalph=2
        v_semaforo_verticalph=0
        WAIT 3 sec
        GOTO saida2
    END
}
v_ph=0
saida2:

WHILE v_ph<2 DO {
    v_semaforo_horizontalph=0

```

```

v_semaforo_verticalalph=0
WAIT 1 sec
v_ph=v_ph +1
IF v_presencaph=1 THEN BEGIN
    v_semaforo_horizontalph=0
    v_semaforo_verticalph=0
    WAIT 2 sec
    GOTO saida3
END
}
v_ph=0
saida3:

WHILE v_ph<50 DO {
    v_semaforo_horizontalph=0
    v_semaforo_verticalph=1
    WAIT 1 sec
    v_ph=v_ph +1
    IF v_presencaph=1 AND v_ph>=25 THEN BEGIN
        GOTO saida4
    END
    IF v_presencaph=1 AND v_ph<25 THEN BEGIN
        WHILE v_ph<25 DO {
            v_semaforo_horizontalph=0
            v_semaforo_verticalph=1
            WAIT 1 sec
            v_ph=v_ph +1
        }
        GOTO saida4
    END
}
v_ph=0
saida4:

WHILE v_ph<3 DO {
    v_semaforo_horizontalph=0
    v_semaforo_verticalph=2
    WAIT 1 sec
    v_ph=v_ph +1
    IF v_presencaph=1 THEN BEGIN
        v_semaforo_horizontalph=0
        v_semaforo_verticalph=2
        WAIT 3 sec
        GOTO saida5
    END
}
v_ph=0

```

saida5:

```
WHILE v_ph<2 DO {
    v_semaforo_horizontalph=0
    v_semaforo_verticalph=0
    WAIT 1 sec
    v_ph=v_ph +1
    IF v_presencaph=1 THEN BEGIN
        v_semaforo_horizontalph=0
        v_semaforo_verticalph=0
        WAIT 2 sec
        GOTO saida6
    END
}
v_ph=0
saida6:
```

goto inicio

sub_semaforopa None

inicio:

```
WHILE v_pa<70 DO {
    v_semaforo_horizontalpa=1
    v_semaforo_verticalpa=0
    WAIT 1 sec
    v_pa=v_pa +1
    IF v_presencapa=1 THEN BEGIN
        v_semaforo_horizontalpa=1
        v_semaforo_verticalpa=0
        WHILE v_pa<100 DO {
            v_semaforo_horizontalpa=1
            v_semaforo_verticalpa=0
            WAIT 1 sec
            v_pa=v_pa +1
        }
    //      WAIT UNTIL v_presencapa=0
    GOTO saida1
}
END
}
```

saida1:

v_pa=0

```
WHILE v_pa<3 DO {
    v_semaforo_horizontalpa=2
    v_semaforo_verticalpa=0
    WAIT 1 sec
    v_pa=v_pa +1
```

```

IF v_presencapa=1 THEN BEGIN
    v_semaforo_horizontalpa=2
    v_semaforo_verticalpa=0
    WAIT 3 sec
    GOTO saida2
END
}
v_pa=0
saida2:

WHILE v_pa<2 DO {
    v_semaforo_horizontalpa=0
    v_semaforo_verticalpa=0
    WAIT 1 sec
    v_pa=v_pa +1
    IF v_presencapa=1 THEN BEGIN
        v_semaforo_horizontalpa=0
        v_semaforo_verticalpa=0
        WAIT 2 sec
        GOTO saida3
    END
}
v_pa=0
saida3:

WHILE v_pa<50 DO {
    v_semaforo_horizontalpa=0
    v_semaforo_verticalpa=1
    WAIT 1 sec
    v_pa=v_pa +1
    IF v_presencapa=1 AND v_pa>=25 THEN BEGIN
        GOTO saida4
    END
    IF v_presencapa=1 AND v_pa<25 THEN BEGIN
        WHILE v_pa<25 DO {
            v_semaforo_horizontalpa=0
            v_semaforo_verticalpa=1
            WAIT 1 sec
            v_pa=v_pa +1
        }
        GOTO saida4
    END
}
v_pa=0
saida4:

```

```

WHILE v_pa<3 DO {
    v_semaforo_horizontalpa=0
    v_semaforo_verticalpa=2
    WAIT 1 sec
    v_pa=v_pa +1
    IF v_presencapa=1 THEN BEGIN
        v_semaforo_horizontalpa=0
        v_semaforo_verticalpa=2
        WAIT 3 sec
        GOTO saida5
    END
}
v_pa=0
saida5:

WHILE v_pa<2 DO {
    v_semaforo_horizontalpa=0
    v_semaforo_verticalpa=0
    WAIT 1 sec
    v_pa=v_pa +1
    IF v_presencapa=1 THEN BEGIN
        v_semaforo_horizontalpa=0
        v_semaforo_verticalpa=0
        WAIT 2 sec
        GOTO saida6
    END
}
v_pa=0
saida6:

goto inicio
sub_semaforosb None
    inicio:
        v_semaforo_horizontalsb=1
        v_semaforo_verticalsb=0
        wait 40 sec

        v_semaforo_horizontalsb=2
        v_semaforo_verticalsb=0
        wait 3 sec

        v_semaforo_horizontalsb=0
        v_semaforo_verticalsb=0
        wait 2 sec

        v_semaforo_horizontalsb=0
        v_semaforo_verticalsb=1

```

```

wait 35 sec

v_semaforo_horizontalsb=0
v_semaforo_verticalsb=2
wait 3 sec

v_semaforo_horizontalsb=0
v_semaforo_verticalsb=0
wait 2 sec

goto inicio
sub_semaforosh None
inicio:

v_semaforo_horizontalsh=1
v_semaforo_verticalsh=0
wait 40 sec

v_semaforo_horizontalsh=2
v_semaforo_verticalsh=0
wait 3 sec

v_semaforo_horizontalsh=0
v_semaforo_verticalsh=0
wait 2 sec

v_semaforo_horizontalsh=0
v_semaforo_verticalsh=1
wait 35 sec

v_semaforo_horizontalsh=0
v_semaforo_verticalsh=2
wait 3 sec

v_semaforo_horizontalsh=0
v_semaforo_verticalsh=0
wait 2 sec

goto inicio
sub_semaforosa None
inicio:

v_semaforo_horizontalsa=1
v_semaforo_verticalsa=0
wait 40 sec

v_semaforo_horizontalsa=2
v_semaforo_verticalsa=0
wait 3 sec

```

v_semaforo_horizontalsa=0
v_semaforo_verticalsa=0
wait 2 sec

v_semaforo_horizontalsa=0
v_semaforo_verticalsa=1
wait 35 sec

v_semaforo_horizontalsa=0
v_semaforo_verticalsa=2
wait 3 sec

v_semaforo_horizontalsa=0
v_semaforo_verticalsa=0
wait 2 sec

goto inicio

sub_cronometro None

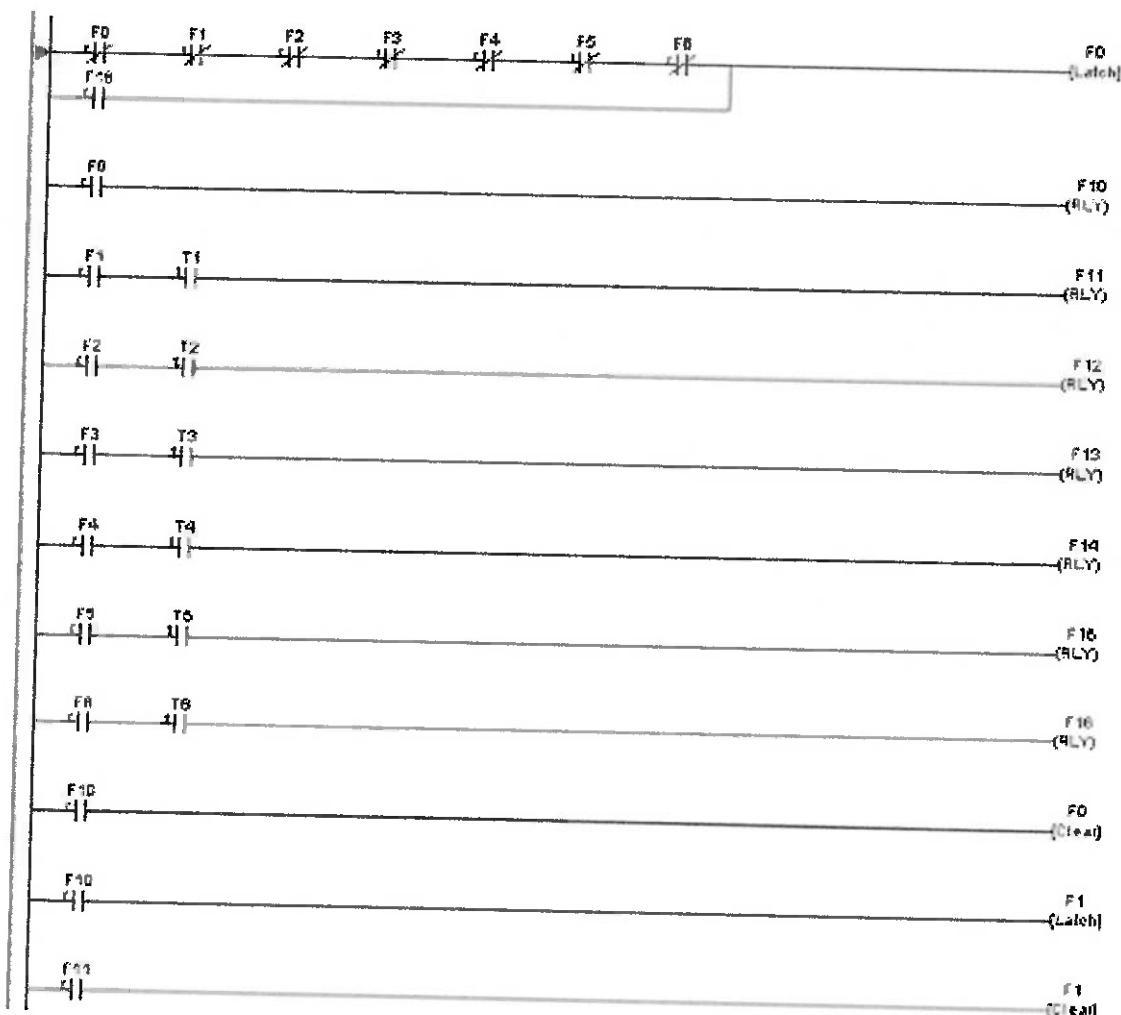
* Arrival Cycles *

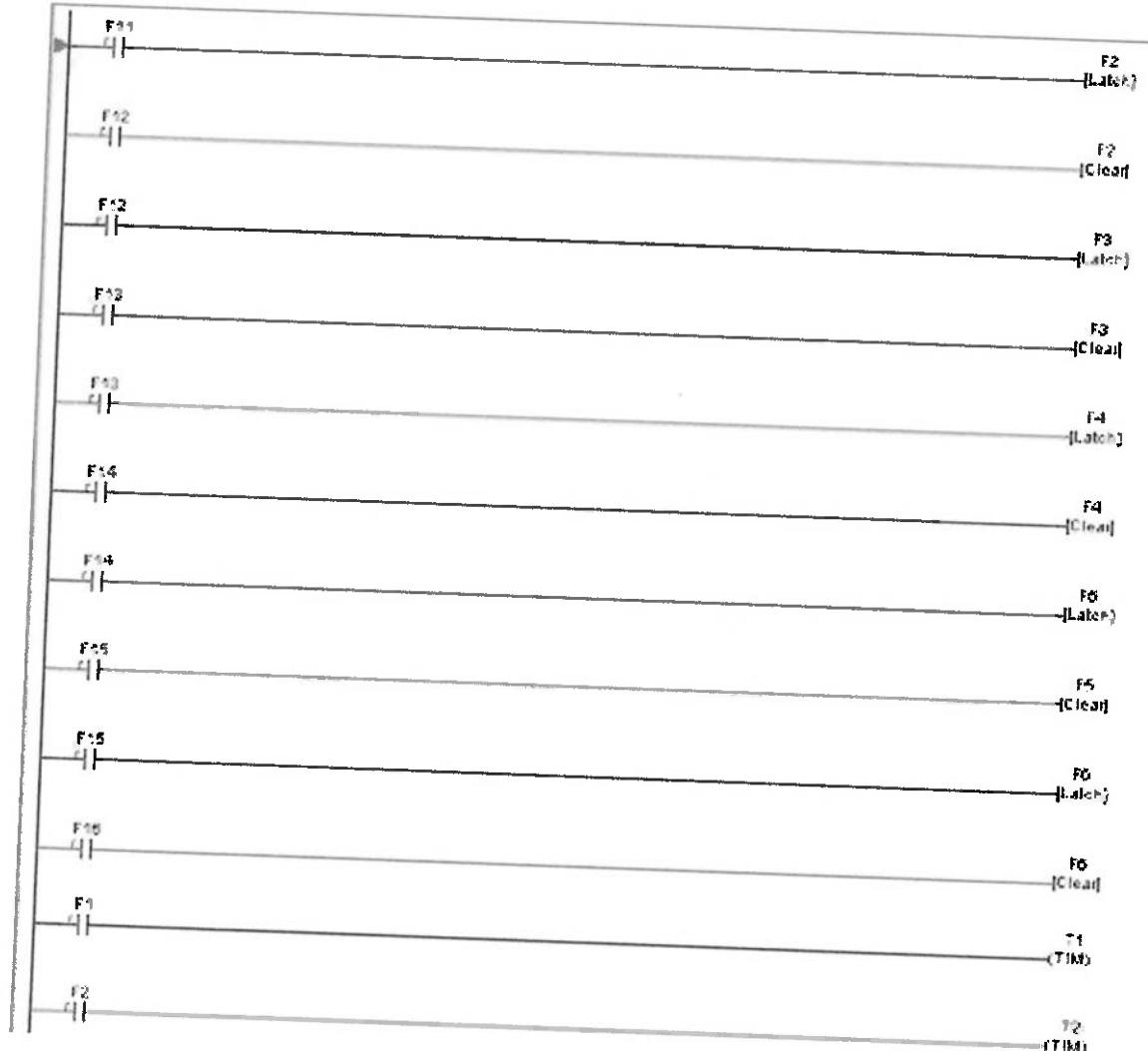
ID	Qty / %	Cumulative	Time (Hours)	Value
paulista_bc	Quantity	No	1	2500
		2	3200	
		3	3800	
		4	4000	
		5	3800	
paulista_cb	Quantity	No	1	2500
		2	3200	
		3	3800	
		4	4000	
		5	2700	
augusta_bc	Quantity	No	1	600
		2	650	
		3	679	
		4	650	
		5	600	
augusta_cb	Quantity	No	1	400
		2	450	
		3	475	

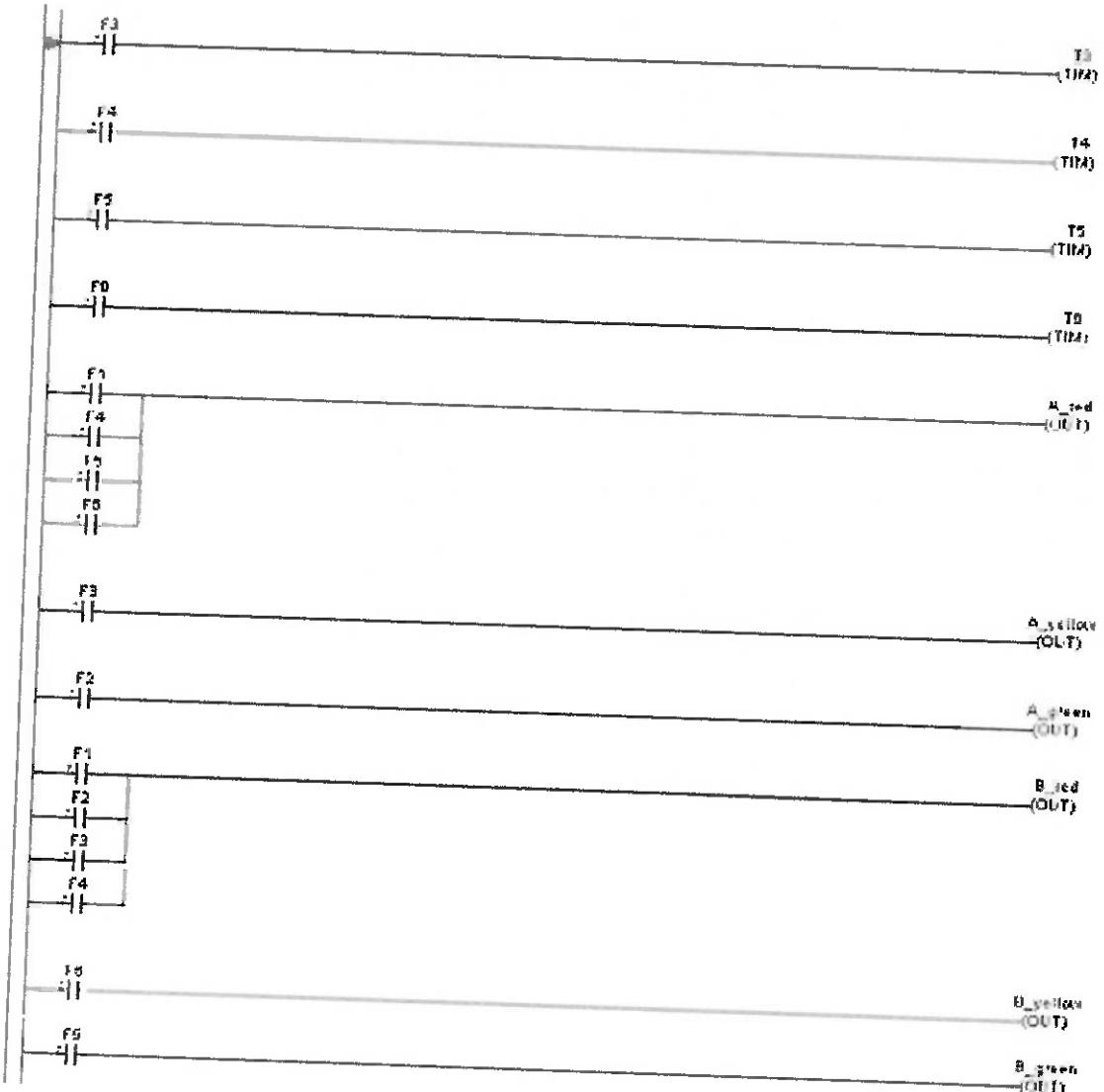
		4	508
		5	500
santos	Quantity	No	1
		2	700
		3	750
		4	900
		5	775
bela_cinta	Quantity	No	1
		2	900
		3	950
		4	975
		5	925
haddock_lobo	Quantity	No	1
		2	975
		3	1150
		4	1250
		5	1200

ANEXO B - Diagrama Ladder do ciclo de luzes de um semáforo

O diagrama Ladder a seguir é a transcrição do SFC (Figura 6) do ciclo de semáforos de apenas um cruzamento. Nota-se no entanto, que na programação final do CLP foram implementados seis cruzamentos. Portanto, considerando o padrão hierárquico do modelo considerado, escreve-se seis vezes este mesmo diagrama apenas com a alteração das variáveis.







ANEXO C - Documentação do supervisório

Necessidade dos Stakeholder	Versão: 1.0	Data: 07/11/2005
Responsável: Fagner Ghingaro	Supervisório de Semáforos	Status: Em desenv.

1. Introdução

A finalidade deste documento é coletar as necessidades dos usuários/stakeholders para implementar o sistema de inventário de sistemas.

Grandes oportunidades podem ser detectadas visando melhorar o desenvolvimento da aplicação. Muitas técnicas estão disponíveis para eliciar pedidos dos stakeholders ou do usuário. Uma técnica simples e barata que é apropriada, é a entrevista genérica. A entrevista genérica pode ajudar o analista a compreender os stakeholders ou os objetivos e os problemas do usuário. A entrevista genérica caracteriza as perguntas projetadas para eliciar uma compreensão dos stakeholders ou problemas e ambiente de usuário. Estas perguntas exploram os requisitos de funcionalidade, de usabilidade, de confiabilidade, de desempenho e de suportabilidade para a aplicação. Em consequência de usar a entrevista genérica, o analista adquirirá o conhecimento do problema que está sendo resolvido, tão bem quanto a compreensão por parte dos stakeholders das características de soluções bem sucedidas.

a. Escopo

Levantar os requisitos levantados nas visitas as dependências da Central de Controle de Tráfego da Companhia de Engenharia de Tráfego da cidade de São Paulo.

b. Referências

Não se aplica

2. Stakeholder ou Perfil do Usuário

Nome do Stakeholder	Área do Stakeholder	Cargo do Stakeholder
Controlador de Tráfego	Central de Controle de Tráfego	Controlador

3. Oportunidades de Melhoria

Número	Descrição	Solicitante
STRQ1	O sistema deve ter um meio de controlar o acesso ao sistema através de login e senhas cadastrados	Controlador de Tráfego
STRQ2	O sistema deve permitir a consulta do estados dos semáforos através de interface gráfica de fácil visualização	Controlador de Tráfego
STRQ3	O sistema deve permitir a alteração do valor do tempo de verde de um área determinada e o tempo de mudança semafórica	Controlador de Tráfego

Documento de Visão do Produto	Versão: 1.0	Data: 07/11/2005
Responsável: Fagner Ghingaro	Supervisório de Semáforos	Status: Em desenv.

1. Introdução

A finalidade deste documento é coletar, analisar e definir necessidade e características de alto nível do Supervisório de Semáforos. Focalizar as capacidades necessárias aos envolvidos, e os usuários-alvo, e porque estas necessidades existem. Os detalhes sobre como o Supervisório de Semáforos atende estas necessidades estão descritos nos casos de uso.

a. Escopo

Desenvolver e implementar o Supervisório de Sistemas que permita que os usuários acompanhem a situação em tempo real do estados de variáveis internas de um Controlador Lógico Programável responsável pelo controle de um protótipo de vias controladas por semáforos.

b. Referencias

Como referência para este documento temos os documentos de Necessidades dos Stakeholders do sistema IVS.

2. Problema/Oportunidade

A implementação de um modelo para controle de semáforos gerou a necessidade de construção de supervisório que permita que os controladores de tráfego posicionados na central de controle de trânsito da Companhia de Engenharia de Trânsito (CET) acompanhem em tempo real o estado dos semáforos de uma determinada área.

3. Descrição do Produto

Um aplicativo em linguagem de programação que permita o acesso via comunicação serial a um ou mais Controladores Lógicos Programáveis e que possua boa interface visual.

4. Descrição dos Envolvidos e Usuários

Esta seção apresenta um perfil dos envolvidos (pessoas ou entidades diretamente influenciadas pelo sucesso do projeto) e usuários do produto, bem como os principais problemas que os afetam e que devem ser resolvidos pela solução da proposta. Esta seção não descreve requisitos específicos (detalhados adiante).

a. Descrição dos Envolvidos

i. Analista

Representado por	Fagner Ghingaro
Descrição do Envolvido	Analista
Tipo	Conhecimento técnico e funcional
Responsabilidades	Mapear as necessidades das interfaces entre o supervisório e o CLP e entre o supervisório e usuário, além de possíveis controles sobre o fluxo lógico implementado no CLP para atender necessidades funcionais

Comentários/Questões	
-----------------------------	--

ii. Usuário Entrevistado e operador do sistema

Representado por	Controlador de Tráfego CET
Descrição do Envolvido	Usuário
Tipo	Usuário contínuo do sistema e com conhecimento das necessidades
Responsabilidades	Informar as necessidades do supervisório
Comentários/Questões	

iii. Analista CLP

Representado por	Dan Okawa
Descrição do Envolvido	Analista
Tipo	Conhecimento técnico e funcional
Responsabilidades	Informar as necessidades para interface entre o supervisório e o CLP
Comentários/Questões	

b. Ambiente do usuário

O Supervisório de Sistemas será instalado diretamente na máquina do usuário.

5. Visão Geral do Produto

a. Contexto do Produto

Este sistema é dependente da conexão do Controlador Lógico Programável com o computador ou rede onde o mesmo está instalado.

b. Características/Requisitos do Produto

i. REQ1 – Acompanhar o estado dos semáforos

Acompanhar o estado dos semáforos através de interface gráfica de fácil entendido os estados dos semáforos de uma área determinada.

ii. REQ2 – Consultar tempo para mudança semafórica

Consultar o tempo de verde de um semáforo e o tempo restante para a mudança semafórica

iii. REQ3 – Alterar tempo de verde

Alterar o tempo de verde de um semáforo ou de um grupo de semáforos caso estes tenham o seu funcionamento dependente dentro do bloco.

iv. REQ4 – Alterar tempo de mudança semafórica

Alterar o tempo de mudança semafórica de um semáforo ou de um grupo de semáforos caso estes tenham o seu funcionamento dependente dentro do bloco.

c. Requisitos funcionais do produto a serem implementados em versões futuras
Fora do escopo do projeto

d. Premissas e dependências

A analista será responsável pelo desenvolvimento do sistema.

e. Restrições

Não existem restrições.

f. Riscos

Não existem riscos.

Caso de Uso Geral	Versão: 1.0	Data: 07/11/2005
Responsável: Fagner Ghingaro	Supervisório de Semáforos	Status: Em desenv.

1. Introdução

A finalidade deste documento é apresentar os Casos de Uso e Atores envolvidos no Supervisório de Sistemas, bem como seus relacionamentos.

2. Atores



CLP



CONTROLADOR



OPERADOR

a. CLP

O CLP é um componente externo ao Supervisório de Semáforos utilizado para o controle eletrônico dos semáforos de uma determinada área, nela está programado toda a lógica do controle, sendo o responsável por enviar o estado dos semáforos para o Supervisório.

b. Operador

O usuário Operador tem permissão de consultar uma área determinada verificando o estado dos semáforos, consultar tempo para mudança semafórica e alterar o tempo de verde de um semáforo.

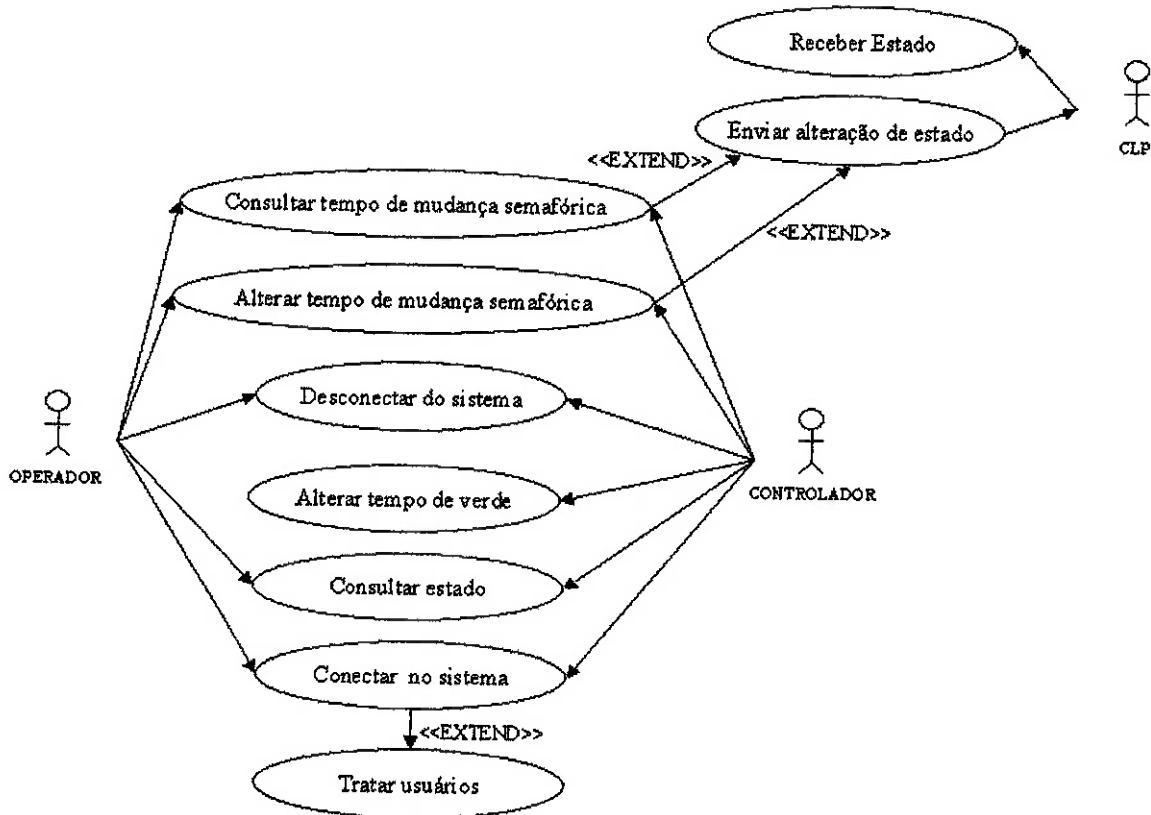
c. Controlador

O controlador possui as mesmas permissões dadas ao operador, além de alterar o tempo de mudança semafórica.

3. Casos de Uso

a. Casos de Uso

Nesta seção será elaborado o diagrama dos casos de uso do sistema. Isto significa incluir somente aqueles casos de uso referentes às regras de negócio e funcionalidades do sistema. Qualquer funcionalidade de apoio aos casos de uso deverá ser incluída no diagrama de Casos de Uso.



i. UC1-Receber Estado

O supervisório de semáforos recebe um determinado estado do CLP, trata este estado para posterior exibição através da interface visual.

ii. UC2-Enviar Alteração de Estado

O supervisório de semáforos envia para o CLP um determinado estado que tenha sido alterado pelo usuário.

iii. UC3-Conectar no Sistema

Usuário digita login e senha para permitir acesso ao sistema, filtrando este usuário como um operador ou um controlador e exibindo na tela somente as funções que lhe são permitidas.

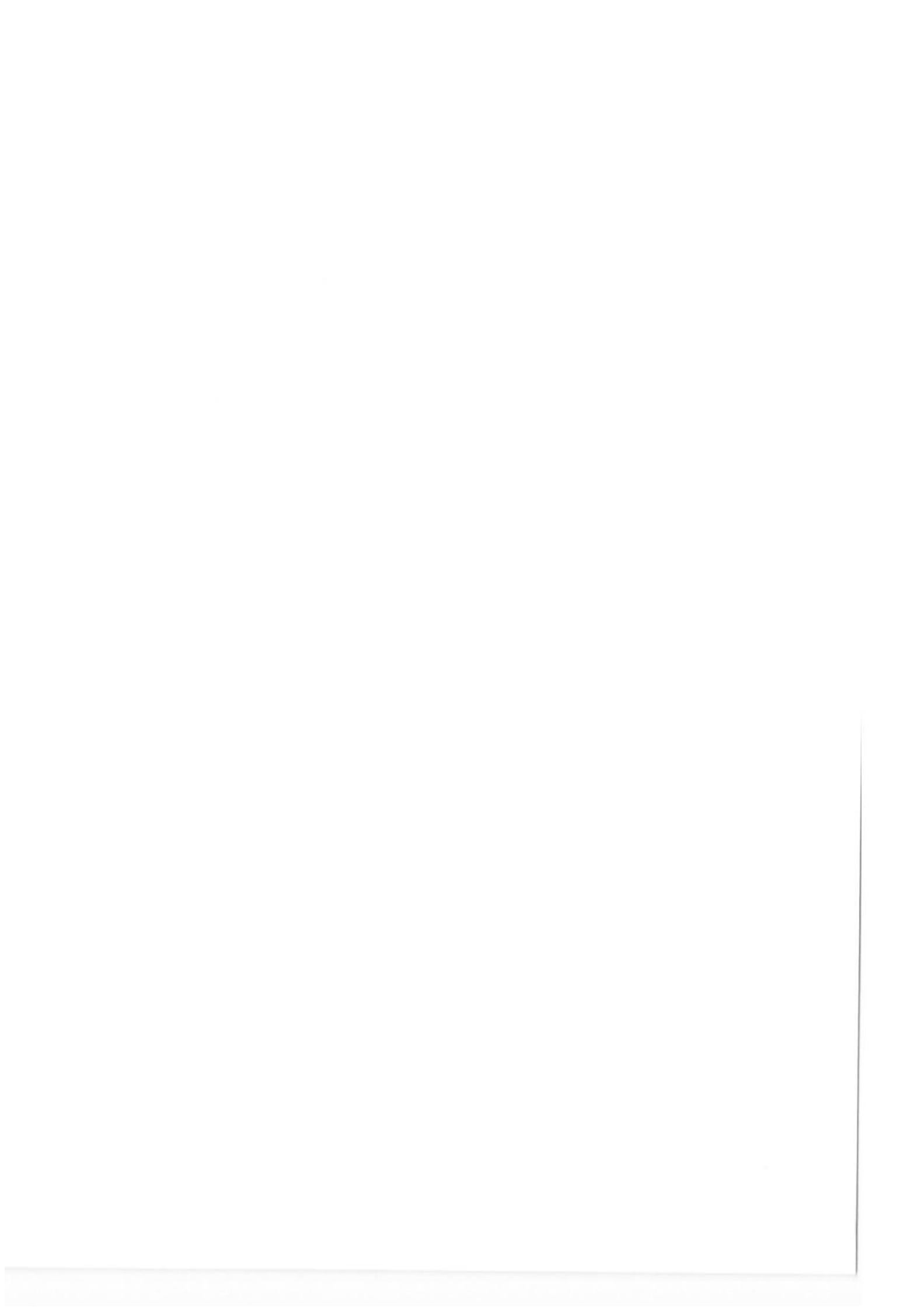
iv. UC4-Consultar Estado

Usuário acessa a função de exibição de uma determinada área cadastrada no sistema, possibilitando consultar o estado dos semáforos.

v. UC5-Consultar Tempo de Mudança Semaforica

Usuário consulta o tempo de verde de um semáforo ou de um grupo de semáforos caso estes possuam algum tipo de dependência.







vi. UC6-Alterar Tempo de Mudança Semaforica

Usuário altera o tempo de verde de um semáforo ou de um grupo de semáforos manualmente para mais ou para menos do valor atual de acordo com suas próprias avaliações.

vii. UC7-Alterar Tempo de Verde

Usuário altera o tempo de mudança semafórica de um semáforo ou de um grupo de semáforos manualmente para mais ou para menos do valor atual de acordo com suas próprias avaliações.

viii. UC8-Tratar acesso dos usuários

Sistema trata o acesso dos usuários entre Operador e Controlador para as funções do sistema.

ix. UC9-Desconectar do Sistema

Usuário efetua a desconexão do sistema.

Especificações Suplementares		Versão: 1.0	Data: 07/11/2005
Responsável: Fagner Ghingaro	Supervisório de Semáforos	Status: Em desenv.	

1. Introdução

O propósito deste documento é coletar, analisar e definir as especificações suplementares para o Supervisório de Semáforos.

a. Escopo

O projeto associado a este documento é o projeto do Supervisório de Semáforos e nada será afetado por este.

b. Referências

Como referência para este documento temos os documentos:

- Necessidade dos stakeholders
- Visão do Produto

2. Regras de negócio ou de Processo

Esta seção detalha regras de negócio importantes para o sistema, mas que, por não interagirem com o usuário, podem não estar claras no levantamento de requisitos do sistema.

Não possui

3. Requisitos Gerais do Sistema

Esta seção detalha requisitos que não descrevem as funcionalidades do sistema. Na verdade, eles descrevem características de infra-estrutura que suportarão o sistema, requisitos de ambiente, e também a lista de padrões aplicáveis ao sistema.

a. Requisito do Sistema

O Supervisório de Sistemas será instalado diretamente na máquina do usuário, portanto para utilização do mesmo, somente será necessário que a máquina seja tenha como Sistema Operacional o Microsoft Windows 98® ou versão superior.

4. Requisitos Não-Funcionais

Esta seção detalha os Requisitos não-funcionais do sistema: usabilidade, confiabilidade, performance e suportabilidade.

a. Requisitos de Usabilidade

Os usuários finais do sistema não necessitarão ter um treinamento específico, pois a interface com o usuário deve ser bem intuitiva. A exigência é que o usuário tenha conhecimento do negócio para o manejo das funções a ele permitidas.

b. Requisitos de Confiabilidade

Não se aplica

c.Requisitos de Performance
Não se aplica.

d.Requisitos de Suportabilidade

5. Restrições do desenho do Sistema/Produto
Não se aplica

6. Documentação Online do Usuário e Requisitos de Sistema de Ajuda
Não se aplica

7. Componentes Adquiridos

Nenhum componente foi adquirido para ser utilizado juntamente com este sistema.

8. Interfaces

a.Interface do usuário

Não se aplica.

b.Interface de Hardware

Não se aplica.

c.Interface de Software

Não se aplica.

d.Interface de Comunicação

O sistema necessita de comunicação via porta serial com o controlador programável Micrologix 1500.

9. Padrões Aplicáveis

Não se aplica.

Caso de Uso Detalhado - Receber Estado	Versão: 1.0	Data: 07/11/2005
Responsável: Fagner Ghingaro	Supervisório de Semáforos	Status: Em desenv.

1. Nome do Caso de Uso

Receber Estado

a. Breve Descrição

O supervisório de semáforos recebe um determinado estado do CLP, trata este estado para posterior exibição através da interface visual.

2. Fluxo de Eventos

a. Fluxo Básico

O fluxo básico do caso de uso é representado pela melhor seqüência de eventos ou “Caminho Ideal”, ou seja, imaginamos uma seqüência, então os eventos ocorrem de maneira transparente ou se problemas.

i. {Receber Estado}

- P1. Armazenar estado em variável temporária
- P2. Tratar valor recebido
- P3. Exibir no objeto correspondente o valor desta variável temporária

b. Fluxos Alternativos

O fluxo alternativo do Caso de Uso é representado pelas exceções imaginadas ao longo de nossa seqüência de eventos

i. Não existem

3. Requisitos Especiais

Esta seção lista os requisitos especiais que são tipicamente requisitos não-funcionais e que é difícil de serem capturados na visão do produto, pois neste documento há um detalhamento de como as tarefas serão executadas para atender aos requisitos funcionais. São os requisitos do caso de uso em estudo.

i. Não existem

4. Pré-Condições

Nesta seção listados os pré-requisitos necessários para que um Caso de Uso seja realizado.

- a. Conexão estabelecida entre o Supervisório de Semáforos e o CLP via comunicação serial
- b. Mapeamento das variáveis do CLP com variáveis do sistema já efetuado

5. Pós-condições

Nesta seção são listadas as pós-condições que garantem a conclusão da realização do Caso de Uso.

a. Não existem

6. Pontos de Extensão

a. Não existem

7. Diagrama de Caso de Uso

Diagrama apresentado no Caso de Uso Geral do sistema Supervisório de Semáforos.

8. Outros Diagramas

Não existem.

Caso de Uso Detalhado - Enviar Alteração de Estado	Versão: 1.0	Data: 07/11/2005
Responsável: Fagner Ghingaro	Supervisório de Semáforos	Status: Em desenv.

1. Nome do Caso de Uso

Enviar Alteração de Estado

a. Breve Descrição

O supervisório de semáforos envia para o CLP um determinado estado que tenha sido alterado pelo usuário.

2. Fluxo de Eventos

a. Fluxo Básico

O fluxo básico do caso de uso é representado pela melhor seqüência de eventos ou “Caminho Ideal”, ou seja, imaginamos uma seqüência, então os eventos ocorrem de maneira transparente ou se problemas.

i. { Enviar Alteração de Estado}

- P1. Determinar para qual variável do sistema está mapeada para a variável do CLP onde será feita a alteração
- P2. Tratar valor recebido de forma que um valor não seja enviado tipos de dados que possam apresentar conflitos
- P3. Enviar estado para a variável interno do CLP
- P4. Verificar se ocorreu algum tipo de erro durante o envio do estado

b. Fluxos Alternativos

O fluxo alternativo do Caso de Uso é representado pelas exceções imaginadas ao longo de nossa seqüência de eventos

i. {Enviar Alteração de Estado}-Erro durante envio

- P1. Determinar para qual variável do sistema está mapeada para a variável do CLP onde será feita a alteração
- P2. Tratar valor recebido de forma que um valor não seja enviado tipos de dados que possam apresentar conflitos
- P3. Enviar estado para a variável interno do CLP
- P4. Verificar se ocorreu algum tipo de erro durante o envio do estado
- P5. Exibir mensagem informando que ouve um Erro durante o envio

3. Requisitos Especiais

Esta seção lista os requisitos especiais que são tipicamente requisitos não-funcionais e que é difícil de serem capturados na visão do produto, pois neste documento há um detalhamento de como as tarefas serão executadas para atender aos requisitos funcionais. São os requisitos do caso de uso em estudo.

i. Não existem

4. Pré-Condições

- Nesta seção listados os pré-requisitos necessários para que um Caso de Uso seja realizado.
- a. Conexão estabelecida entre o Supervisório de Semáforos e o CLP via comunicação serial
 - b. Mapeamento das variáveis do CLP com variáveis do sistema já efetuado

5. Pós-condições

Nesta seção são listadas as pós-condições que garantem a conclusão da realização do Caso de Uso.

- a. Não existem

6. Pontos de Extensão

- a. Não existem

7. Diagrama de Caso de Uso

Diagrama apresentado no Caso de Uso Geral do sistema Supervisório de Semáforos.

8. Outros Diagramas

Não existem.

Caso de Uso Detalhado – Conectar no Sistema	Versão: 1.0	Data: 07/11/2005
Responsável: Fagner Ghingaro	Supervisório de Semáforos	Status: Em desenv.

1. Nome do Caso de Uso

Logar no Sistema

a. Breve Descrição

Usuário digita login e senha para permitir acesso ao sistema, filtrando este usuário como um operador ou um controlador e exibindo na tela somente as funções que lhe são permitidas.

2. Fluxo de Eventos

a. Fluxo Básico

O fluxo básico do caso de uso é representado pela melhor seqüência de eventos ou “Caminho Ideal”, ou seja, imaginamos uma seqüência, então os eventos ocorrem de maneira transparente ou se problemas.

i. {Logar no Sistema}

- P1. No menu <Conexão> clicar sobre a função <Conectar>
- P2. Digitar o login no sistema
- P3. Digitar a senha no sistema
- P4. Confirmar login e senha

b. Fluxos Alternativos

O fluxo alternativo do Caso de Uso é representado pelas exceções imaginadas ao longo de nossa seqüência de eventos

i. {Logar no Sistema}-Login Errado

- P1. No menu <Conexão> clicar sobre a função <Conectar>
- P2. Digitar o login no sistema
- P3. Digitar a senha no sistema
- P4. Confirmar login e senha
- P5. Exibir mensagem informando que o login não foi encontrado

ii. {Logar no Sistema}-Senha Errada

- P1. No menu <Conexão> clicar sobre a função <Conectar>
- P2. Digitar o login no sistema
- P3. Digitar a senha no sistema
- P4. Confirmar login e senha
- P5. Exibir mensagem informando que a senha não foi encontrada

3. Requisitos Especiais

Esta seção lista os requisitos especiais que são tipicamente requisitos não-funcionais e que é difícil de serem capturados na visão do produto, pois neste documento há um detalhamento de como as tarefas serão executadas para atender aos requisitos funcionais. São os requisitos do caso de uso em estudo.

- i. Não existem

4. Pré-Condições

Nesta seção listados os pré-requisitos necessários para que um Caso de Uso seja realizado.

- a. Usuário deve estar cadastrado

5. Pós-condições

Nesta seção são listadas as pós-condições que garantem a conclusão da realização do Caso de Uso.

- a. Não existem

6. Pontos de Extensão

- a. Não existem

7. Diagrama de Caso de Uso

Diagrama apresentado no Caso de Uso Geral do sistema Supervisório de Semáforos.

8. Outros Diagramas

Não existem.

Caso de Uso Detalhado – Consultar estado	Versão: 1.0	Data: 07/11/2005
Responsável: Fagner Ghingaro	Supervisório de Semáforos	Status: Em desenv.

1. Nome do Caso de Uso

Consultar Estado

a. Breve Descrição

Usuário acessa a função de exibição de uma determinada área cadastrada no sistema, possibilitando consultar o estado dos semáforos.

2. Fluxo de Eventos

a. Fluxo Básico

O fluxo básico do caso de uso é representado pela melhor seqüência de eventos ou “Caminho Ideal”, ou seja, imaginamos uma seqüência, então os eventos ocorrem de maneira transparente ou se problemas.

i. {Logar no Sistema}

- P1. No menu <Arquivo> clicar sobre a função <Consultar>
- P2. Escolher uma das áreas cadastradas na lista de dados
- P3. Exibir a área escolhida

b. Fluxos Alternativos

O fluxo alternativo do Caso de Uso é representado pelas exceções imaginadas ao longo de nossa seqüência de eventos

i. Não existem

3. Requisitos Especiais

Esta seção lista os requisitos especiais que são tipicamente requisitos não-funcionais e que é difícil de serem capturados na visão do produto, pois neste documento há um detalhamento de como as tarefas serão executadas para atender aos requisitos funcionais. São os requisitos do caso de uso em estudo.

i. Não existem

4. Pré-Condições

Nesta seção listados os pré-requisitos necessários para que um Caso de Uso seja realizado.

a. Área deve estar cadastrada

5. Pós-condições

Nesta seção são listadas as pós-condições que garantem a conclusão da realização do Caso de Uso.

a. Não existem

6. Pontos de Extensão

a. Não existem

7. Diagrama de Caso de Uso

Diagrama apresentado no Caso de Uso Geral do sistema Supervisório de Semáforos.

8. Outros Diagramas

Não existem.

Caso de Uso Detalhado – Consultar Tempo de Mudanças Semaforica	Versão: 1.0	Data: 07/11/2005
Responsável: Fagner Ghingaro	Supervisório de Semáforos	Status: Em desenv.

1. Nome do Caso de Uso

Consultar Tempo de Mudança Semaforica

a. Breve Descrição

Usuário consulta o tempo de Mudança Semaforica de um semáforo ou de um grupo de semáforos caso estes possuam algum tipo de dependência

2. Fluxo de Eventos

a. Fluxo Básico

O fluxo básico do caso de uso é representado pela melhor seqüência de eventos ou “Caminho Ideal”, ou seja, imaginamos uma seqüência, então os eventos ocorrem de maneira transparente ou se problemas.

i. {Consultar Tempo de Mudança Semaforica }

- P1. Consultar uma área
- P2. Clicar sobre o botão alterar tempos
- P3. Escolher o cruzamento desejado
- P4. Exibir tempo de mudança semafórica

b. Fluxos Alternativos

O fluxo alternativo do Caso de Uso é representado pelas exceções imaginadas ao longo de nossa seqüência de eventos

i. Não existem

3. Requisitos Especiais

Esta seção lista os requisitos especiais que são tipicamente requisitos não-funcionais e que é difícil de serem capturados na visão do produto, pois neste documento há um detalhamento de como as tarefas serão executadas para atender aos requisitos funcionais. São os requisitos do caso de uso em estudo.

a. Não existem

4. Pré-Condições

Nesta seção listados os pré-requisitos necessários para que um Caso de Uso seja realizado.

a. Não existem

5. Pós-condições

Nesta seção são listadas as pós-condições que garantem a conclusão da realização do Caso de Uso.

a. Não existem

6. Pontos de Extensão

a. Não existem

7. Diagrama de Caso de Uso

Diagrama apresentado no Caso de Uso Geral do sistema Supervisório de Semáforos.

8. Outros Diagramas

Não existem.

Caso de Uso Detalhado – Alterar Tempo de Mudança Semaforica	Versão: 1.0	Data: 07/11/2005
Responsável: Fagner Ghingaro	Supervisório de Semáforos	Status: Em desenv.

1. Nome do Caso de Uso

Alterar Tempo de Mudança Semaforica

a. Breve Descrição

Usuário altera o tempo de Mudança Semaforica de um semáforo ou de um grupo de semáforos manualmente para mais o para menos do valor atual de acordo com suas próprias avaliações.

2. Fluxo de Eventos

a. Fluxo Básico

O fluxo básico do caso de uso é representado pela melhor seqüência de eventos ou “Caminho Ideal”, ou seja, imaginamos uma seqüência, então os eventos ocorrem de maneira transparente ou se problemas.

i. {Alterar Tempo de Mudança Semaforica}

- P1. Clicar sobre o tempo de mudança semafórica exibido
- P5. Digitar valor
- P6. Clicar sobre o botão OK
- P7. Pedir confirmação de alteração
- P8. Enviar alteração

b. Fluxos Alternativos

O fluxo alternativo do Caso de Uso é representado pelas exceções imaginadas ao longo de nossa seqüência de eventos

i. {Alterar Tempo de Mudança Semaforica}- Usuário não confirmar alteração

- P1. Clicar sobre o tempo de mudança semafórica exibido
- P2. Alterar o valor mostrado
- P3. Clicar no botão alterar
- P4. Pedir confirmação
- P5. Usuário cancela alteração
- P6. Não enviar alteração

3. Requisitos Especiais

Esta seção lista os requisitos especiais que são tipicamente requisitos não-funcionais e que é difícil de serem capturados na visão do produto, pois neste documento há um detalhamento de como as tarefas serão executadas para atender aos requisitos funcionais. São os requisitos do caso de uso em estudo.

- a. Não existem

4. Pré-Condições

Nesta seção listados os pré-requisitos necessários para que um Caso de Uso seja realizado.

- a. Ter feito consulta a um tempo de mudança semafórica

5. Pós-condições

Nesta seção são listadas as pós-condições que garantem a conclusão da realização do Caso de Uso.

- a. Não existem

6. Pontos de Extensão

- a. Não existem

7. Diagrama de Caso de Uso

Diagrama apresentado no Caso de Uso Geral do sistema Supervisório de Semáforos.

8. Outros Diagramas

Não existem.

Alterar Tempo de Mudança Semaforica

Caso de Uso Detalhado – Alterar Tempo de Verde		Versão: 1.0	Data: 07/11/2005
Responsável: Fagner Ghingaro	Supervisório de Semáforos	Status: Em desenv.	

1. Nome do Caso de Uso

Alterar Tempo de Verde

a. Breve Descrição

Usuário altera o tempo de Verde de um semáforo ou de um grupo de semáforos manualmente para mais ou para menos do valor atual de acordo com suas próprias avaliações.

2. Fluxo de Eventos

a. Fluxo Básico

O fluxo básico do caso de uso é representado pela melhor seqüência de eventos ou “Caminho Ideal”, ou seja, imaginamos uma seqüência, então os eventos ocorrem de maneira transparente ou se problemas.

i. {Alterar Tempo Verde}

- P1. Clicar sobre o tempo de verde
- P2. Posicionar Text Box onde será digitado o valor do tempo
- P3. Focar no Text Box onde será digitado o valor do tempo
- P4. Digitar o tempo desejado
- P5. Apertar a tecla <ENTER>
- P6. Enviar alteração

b. Fluxos Alternativos

O fluxo alternativo do Caso de Uso é representado pelas exceções imaginadas ao longo de nossa seqüência de eventos

i. {Alterar Tempo de Verde}- Tempo desejado maior que o Valor PRE do timer

- P1. Clicar sobre o tempo de verde
- P2. Posicionar Text Box onde será digitado o valor do tempo
- P3. Focar no Text Box onde será digitado o valor do tempo
- P4. Digitar o tempo desejado
- P5. Apertar a tecla <ENTER>
- P6. Tempo desejado maior que o Valor PRE do timer
- P7. Solicitar alteração do tempo

3. Requisitos Especiais

Esta seção lista os requisitos especiais que são tipicamente requisitos não-funcionais e que é difícil de serem capturados na visão do produto, pois neste documento há um detalhamento de como as tarefas serão executadas para atender aos requisitos funcionais. São os requisitos do caso de uso em estudo.

a. Não existem

4. Pré-Condições

Nesta seção listados os pré-requisitos necessários para que um Caso de Uso seja realizado.

a. Não existem

5. Pós-condições

Nesta seção são listadas as pós-condições que garantem a conclusão da realização do Caso de Uso.

a. Não existem

6. Pontos de Extensão

a. Não existem

7. Diagrama de Caso de Uso

Diagrama apresentado no Caso de Uso Geral do sistema Supervisório de Semáforos.

Caso de Uso Detalhado – Tratar acesso dos usuários	Versão: 1.0	Data: 07/11/2005
Responsável: Fagner Ghingaro	Supervisório de Semáforos	Status: Em desenv.

1. Nome do Caso de Uso

Tratar acesso dos usuários

a. Breve Descrição

Sistema trata o acesso dos usuários entre Operador e Controlador para as funções do sistema.

2. Fluxo de Eventos

a. Fluxo Básico

O fluxo básico do caso de uso é representado pela melhor seqüência de eventos ou “Caminho Ideal”, ou seja, imaginamos uma seqüência, então os eventos ocorrem de maneira transparente ou se problemas.

i. {Tratar acesso dos usuários}

P1. Verificar se o usuário está cadastrado como Operador ou Controlador

P2. Habilita as funções conforme o tipo de usuário

b. Fluxos Alternativos

O fluxo alternativo do Caso de Uso é representado pelas exceções imaginadas ao longo de nossa seqüência de eventos

3. Requisitos Especiais

Esta seção lista os requisitos especiais que são tipicamente requisitos não-funcionais e que é difícil de serem capturados na visão do produto, pois neste documento há um detalhamento de como as tarefas serão executadas para atender aos requisitos funcionais. São os requisitos do caso de uso em estudo.

c. Não existem

4. Pré-Condições

Nesta seção listados os pré-requisitos necessários para que um Caso de Uso seja realizado.

a. Não existem

5. Pós-condições

Nesta seção são listadas as pós-condições que garantem a conclusão da realização do Caso de Uso.

a. Não existem

6. Pontos de Extensão

a. Não existem

7. Diagrama de Caso de Uso

Diagrama apresentado no Caso de Uso Geral do sistema Supervisório de Semáforos.

8. Outros Diagramas

Não existem.

Caso de Uso Detalhado – Desconectar do Sistema	Versão: 1.0	Data: 07/11/2005
Responsável: Fagner Ghingaro	Supervisório de Semáforos	Status: Em desenv.

1. Nome do Caso de Uso

Desconectar do Sistema

a. Breve Descrição

Usuário efetua a desconexão do sistema.

2. Fluxo de Eventos

a. Fluxo Básico

O fluxo básico do caso de uso é representado pela melhor seqüência de eventos ou “Caminho Ideal”, ou seja, imaginamos uma seqüência, então os eventos ocorrem de maneira transparente ou se problemas.

i. { Desconectar do Sistema}

P1. Clicar no menu <Conexão>

P2. Clicar sobre a função <Desconectar>

b. Fluxos Alternativos

O fluxo alternativo do Caso de Uso é representado pelas exceções imaginadas ao longo de nossa seqüência de eventos

3. Requisitos Especiais

Esta seção lista os requisitos especiais que são tipicamente requisitos não-funcionais e que é difícil de serem capturados na visão do produto, pois neste documento há um detalhamento de como as tarefas serão executadas para atender aos requisitos funcionais. São os requisitos do caso de uso em estudo.

a. Não existem

4. Pré-Condições

Nesta seção listados os pré-requisitos necessários para que um Caso de Uso seja realizado.

a. O usuário estar conectado

5. Pós-condições

Nesta seção são listadas as pós-condições que garantem a conclusão da realização do Caso de Uso.

a. Não existem

6. Pontos de Extensão

a. Não existem

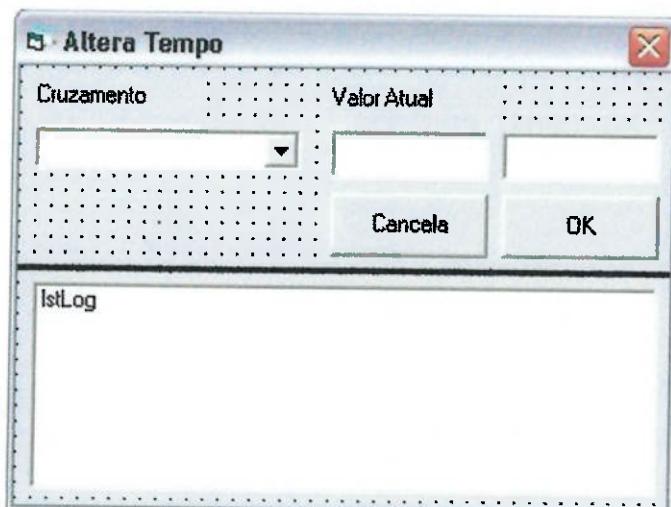
7. Diagrama de Caso de Uso

Diagrama apresentado no Caso de Uso Geral do sistema Supervisório de Semáforos.

8. Outros Diagramas

Não existem.

ANEXO D - Codificação do supervisório



```
Option Explicit
```

```
Public x As Integer
```

```
Public v_Valor As String
```

```
Private Sub cmdCancela_Click()
```

```
Unload Me
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
f_Preenche_Combo
```

```
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)
```

```
frmPaulista.Enabled = True
```

```
End Sub
```

```
Public Sub f_Preenche_Combo()
```

```
cboCruzamento.AddItem "Paulista X Bela Cintra Verde", 0
```

```
cboCruzamento.AddItem "Bela Cintra X Paulista Verde", 1
```

```
cboCruzamento.AddItem "Paulista X Haddock Lobo Verde", 2
```

```
cboCruzamento.AddItem "Haddock Lobo X Paulista Verde", 3
```

```
cboCruzamento.AddItem "Paulista X Augusta Verde", 4
```

```
cboCruzamento.AddItem "Augusta X Paulista Verde", 5
```

```
End Sub
```

```
Private Sub txtValor_KeyPress(KeyAscii As Integer)
```

```
If (KeyAscii >= 48 And KeyAscii <= 57) Or KeyAscii = 8 Then
```

```
    KeyAscii = KeyAscii
```

```
Else
```

```

    KeyAscii = 0
End If
End Sub

*****
*!
Nome da Função: cboCruzamento_Click
' Caso de uso: Consultar Tempo de Mudanças Semaforica
' Responsável: Fagner Ghingaro
' Entrada: Nenhuma
' Saída: Nenhuma
' Observação:
*****
*!

Private Sub cboCruzamento_Click()
If cboCruzamento.ItemData(cboCruzamento.ListIndex) = 0 Then
    txtAtual.Text = Timer_av
ElseIf cboCruzamento.ItemData(cboCruzamento.ListIndex) = 1 Then
    txtAtual.Text = Timer_bv
ElseIf cboCruzamento.ItemData(cboCruzamento.ListIndex) = 2 Then
    txtAtual.Text = Timer_cv
ElseIf cboCruzamento.ItemData(cboCruzamento.ListIndex) = 3 Then
    txtAtual.Text = Timer_dv
ElseIf cboCruzamento.ItemData(cboCruzamento.ListIndex) = 4 Then
    txtAtual.Text = Timer_ev
ElseIf cboCruzamento.ItemData(cboCruzamento.ListIndex) = 5 Then
    txtAtual.Text = Timer_fv
End If
End Sub

*****
*!
Nome da Função: cmdOk_Click
' Caso de uso: Alterar Tempo de Mudança Semaforica
' Responsável: Fagner Ghingaro
' Entrada: Nenhuma
' Saída: Nenhuma
' Observação:
*****
*!

Private Sub cmdOk_Click()
If MsgBox("Deseja realmente efetuar esta alteração?", vbYesNo, "Confirmação") = vbYes
Then
    If cboCruzamento.ItemData(cboCruzamento.ListIndex) = 0 Then
        frmPaulista.f_Envia_Estado 3, txtValor.Text, lstLog
    ElseIf cboCruzamento.ItemData(cboCruzamento.ListIndex) = 1 Then
        frmPaulista.f_Envia_Estado 9, txtValor.Text, lstLog

```

```

ElseIf cboCruzamento.ItemData(cboCruzamento.ListIndex) = 2 Then
    frmPaulista.f_Envia_Estado 15, txtValor.Text, lstLog
ElseIf cboCruzamento.ItemData(cboCruzamento.ListIndex) = 3 Then
    frmPaulista.f_Envia_Estado 21, txtValor.Text, lstLog
ElseIf cboCruzamento.ItemData(cboCruzamento.ListIndex) = 4 Then
    frmPaulista.f_Envia_Estado 27, txtValor.Text, lstLog
ElseIf cboCruzamento.ItemData(cboCruzamento.ListIndex) = 5 Then
    frmPaulista.f_Envia_Estado 33, txtValor.Text, lstLog
End If
cboCruzamento_Click
txtValor.Text = Null
End If
End Sub

```



```

Private Sub Form_Load()
cboArea.AddItem "Paulista"
cboArea.ItemData(cboArea.NewIndex) = 1
End Sub

```

```
*****
*
```

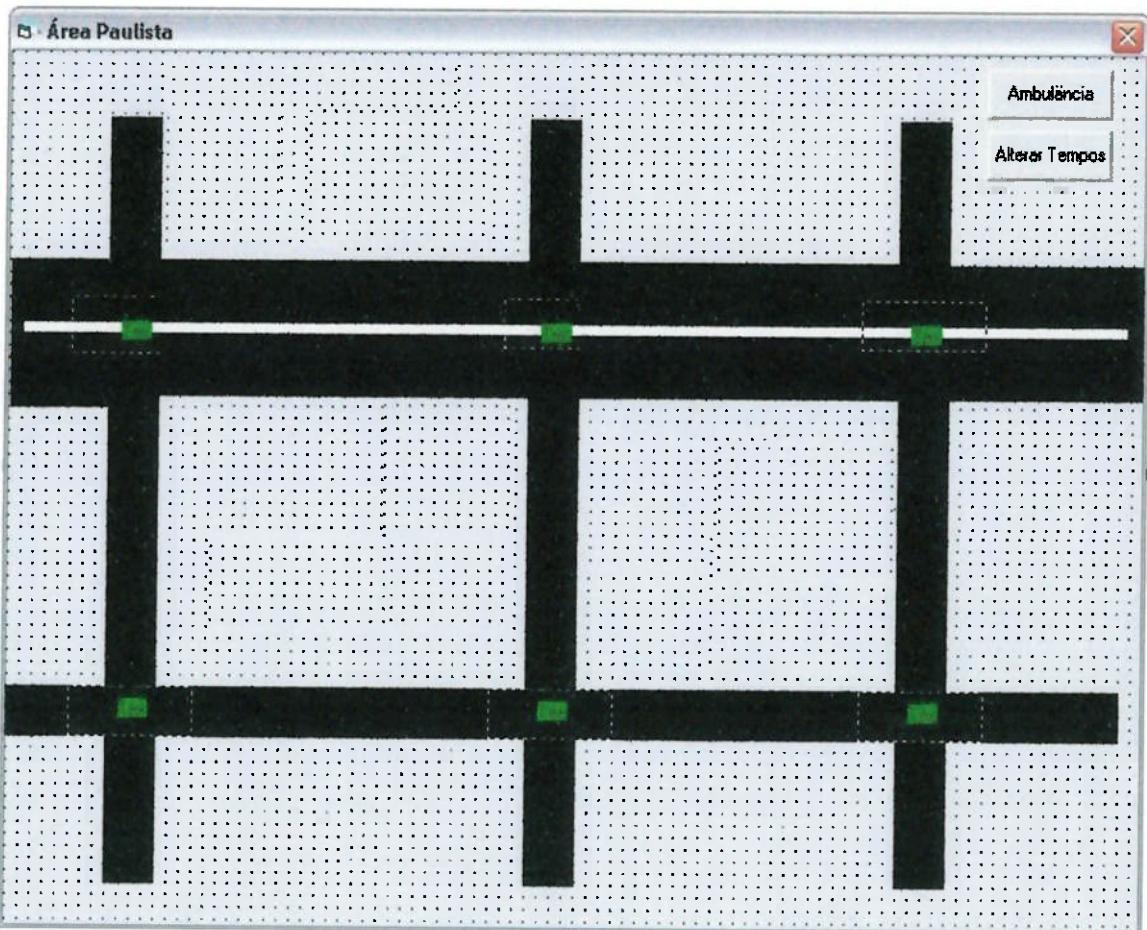
Nome da Função: cboArea_Click
 ' Caso de uso: Consultar Estado
 ' Responsável: Fagner Ghingaro
 ' Entrada: Nenhuma
 ' Saída: Nenhuma
 ' Observação:

```
*****
*
```

```

Private Sub cboArea_Click()
Dim v_Form As Control
If cboArea.Text = "Paulista" Then
    frmPaulista.Show
End If
frmArea.Hide
End Sub

```



```
Option Explicit
Dim TimerServer As OPCServer
Dim OutputServer As OPCServer
Dim InputServer As OPCServer
Dim WithEvents Timers As OPCGroup
Dim WithEvents Outputs As OPCGroup
Dim WithEvents Inputs As OPCGroup
Dim v_Vermelho, v_Verde, v_Amarelo, x As Double
Public v_Timer, v_Cruzamento As Integer
```

```
Private Sub txt11Verde_KeyPress(KeyAscii As Integer)
If 48 <= KeyAscii <= 57 Then
    v_Timer = 3
    v_Cruzamento = Timer_A
    f_Altera_Verde txt11Verde, KeyAscii
End If
End Sub
```

```
Private Sub txt11Verde2_KeyPress(KeyAscii As Integer)
If 48 <= KeyAscii <= 57 Then
    v_Timer = 9
End If
```

```

v_Cruzamento = Timer_B
f_Altera_Verde txt11Verde2, KeyAscii
End If
End Sub

Private Sub txt12Verde_KeyPress(KeyAscii As Integer)
If 48 <= KeyAscii <= 57 Then
    v_Timer = 15
    v_Cruzamento = Timer_C
    f_Altera_Verde txt12Verde, KeyAscii
End If
End Sub

Private Sub txt12Verde2_KeyPress(KeyAscii As Integer)
If 48 <= KeyAscii <= 57 Then
    v_Timer = 21
    v_Cruzamento = Timer_D
    f_Altera_Verde txt12Verde2, KeyAscii
End If
End Sub

Private Sub txt13Verde_KeyPress(KeyAscii As Integer)
If 48 <= KeyAscii <= 57 Then
    v_Timer = 27
    v_Cruzamento = Timer_E
    f_Altera_Verde txt13Verde, KeyAscii
End If
End Sub

Private Sub txt13Verde2_KeyPress(KeyAscii As Integer)
If 48 <= KeyAscii <= 57 Then
    v_Timer = 33
    v_Cruzamento = Timer_F
    f_Altera_Verde txt13Verde2, KeyAscii
End If
End Sub

Private Sub cmdAltera_Click()
frmPaulista.Enabled = False
frmAlteraTempo.Show
End Sub

Private Sub cmdAmbulancia_Click()

End Sub

Private Sub txtVerde_KeyPress(KeyAscii As Integer)

```

```

If (KeyAscii >= 48 And KeyAscii <= 57) Or KeyAscii = 13 Or KeyAscii = 8 Then
    If KeyAscii = 13 Then
        If txtVerde.Text < v_Cruzamento Then
            f_Envia_Estado v_Timer, txtVerde.Text, List1
        Else
            MsgBox "O valor deve ser menor que o tempo de verde do semáforo."
            txtVerde.SetFocus
        End If
    End If
Else
    KeyAscii = 0
End If
End Sub

```

```
Private Sub Form_Load()
```

```
'CORES
v_Vermelho = 255
v_Verde = 65280
v_Amarelo = 65535
```

```
f_PosObjetos
```

```
f_SetServer
```

```
f_SetData
```

```
End Sub
```

```
Private Sub Form_QueryUnload(Cancel As Integer, UnloadMode As Integer)
```

```
f_ServerUnload
```

```
End Sub
```

```
*****
*
```

```
Nome da Função:Inputs_DataChange
```

```
' Caso de uso:Receber Estado dos Inputs
' Responsável:Fagner Ghingaro
' Entrada:ByVal TransactionID As Long, ByVal NumItems As Long,
'          ClientHandles() As Long, ItemValues() As Variant,
'          ItemValues() As Long, ItemValues() As Date
' Saída:Nenhuma
```

```
' Observação:Entradas fixas enviadas pelo CLP, não podem ser alteradas
*****
```

```
*****
Private Sub Inputs_DataChange(ByVal TransactionID As Long, ByVal NumItems As
Long, ClientHandles() As Long, ItemValues() As Variant, Qualities() As Long,
TimeStamps() As Date)
```

```

Dim v_Data As String
For x = 1 To NumItems
    v_Data = ItemValues(x)
    If ClientHandles(x) = 1 Then
        If v_Data = 1 Then
        Elseif v_Data = 0 Then
        End If
    End If
    If ClientHandles(x) = 2 Then
        If v_Data = 1 Then
        Elseif v_Data = 0 Then
        End If
    End If
    If ClientHandles(x) = 3 Then
        If v_Data = 1 Then
        Elseif v_Data = 0 Then
        End If
    End If
Next x
End Sub

```

```

*****
*!
'Nome da Função:Outputs_DataChange
'Caso de uso:Receber Estado dos Outputs
'Responsável:Fagner Ghingaro
'Entrada:ByVal TransactionID As Long, ByVal NumItems As Long,
'        ClientHandles() As Long, ItemValues() As Variant,
'        ItemValues() As Long, ItemValues() As Date
'Saída:Nenhuma
'Observação:Entradas fixas enviadas pelo CLP, não podem ser alteradas
*****
*
```

```

Private Sub Outputs_DataChange(ByVal TransactionID As Long, ByVal NumItems As
Long, ClientHandles() As Long, ItemValues() As Variant, Qualities() As Long,
TimeStamps() As Date)
Dim v_Data As String
For x = 1 To NumItems
    v_Data = ItemValues(x)
    'a_yellow
    If ClientHandles(x) = 1 Then
        If v_Data = 1 Then
            txt11Amar.Visible = True
            txt11Amar2.Visible = False
            txt11Verde.Visible = False
            txt11Verde2.Visible = False
            f_Pos_Imagen img11, H, 11
        End If
    End If
Next x
End Sub

```

```

        img11.Picture = imgHamar.Picture
    End If
'a_green
ElseIf ClientHandles(x) = 2 Then
    If v_Data = 1 Then
        txt11Amar.Visible = False
        txt11Amar2.Visible = False
        txt11Verde.Visible = True
        txt11Verde2.Visible = False
        f_Pos_Imagen img11, H, 11
        img11.Picture = imgHVerde.Picture
    End If
'b_yellow
ElseIf ClientHandles(x) = 3 Then
    If v_Data = 1 Then
        txt11Amar.Visible = False
        txt11Amar2.Visible = True
        txt11Verde.Visible = False
        txt11Verde2.Visible = False
        f_Pos_Imagen img11, V, 11
        img11.Picture = imgCAmar.Picture
    End If
'b_green
ElseIf ClientHandles(x) = 4 Then
    If v_Data = 1 Then
        txt11Amar.Visible = False
        txt11Amar2.Visible = False
        txt11Verde.Visible = False
        txt11Verde2.Visible = True
        f_Pos_Imagen img11, V, 11
        img11.Picture = imgCVerde.Picture
    End If
'c_yellow
ElseIf ClientHandles(x) = 5 Then
    If v_Data = 1 Then
        txt12Amar.Visible = True
        txt12Amar2.Visible = False
        txt12Verde.Visible = False
        txt12Verde2.Visible = False
        f_Pos_Imagen img12, H, 12
        img12.Picture = imgHamar.Picture
    End If
'c_green
ElseIf ClientHandles(x) = 6 Then
    If v_Data = 1 Then
        txt12Amar.Visible = False
        txt12Amar2.Visible = False

```

```

txt12Verde.Visible = True
txt12Verde2.Visible = False
f_Pos_Imagem img12, H, 12
img12.Picture = imgHVerde.Picture
End If
'd_yellow
ElseIf ClientHandles(x) = 7 Then
  If v_Data = 1 Then
    txt12Amar.Visible = False
    txt12Amar2.Visible = True
    txt12Verde.Visible = False
    txt12Verde2.Visible = False
    f_Pos_Imagem img12, V, 12
    img12.Picture = imgCAmar.Picture
  End If
'd_green
ElseIf ClientHandles(x) = 8 Then
  If v_Data = 1 Then
    txt12Amar.Visible = False
    txt12Amar2.Visible = False
    txt12Verde.Visible = False
    txt12Verde2.Visible = True
    f_Pos_Imagem img12, V, 12
    img12.Picture = imgCVerde.Picture
  End If
'e_yellow
ElseIf ClientHandles(x) = 9 Then
  If v_Data = 1 Then
    txt13Amar.Visible = True
    txt13Amar2.Visible = False
    txt13Verde.Visible = False
    txt13Verde2.Visible = False
    f_Pos_Imagem img13, H, 13
    img13.Picture = imgHamar.Picture
  End If
'e_green
ElseIf ClientHandles(x) = 10 Then
  If v_Data = 1 Then
    txt13Amar.Visible = False
    txt13Amar2.Visible = False
    txt13Verde.Visible = True
    txt13Verde2.Visible = False
    f_Pos_Imagem img13, H, 13
    img13.Picture = imgHVerde.Picture
  End If
'f_yellow
ElseIf ClientHandles(x) = 11 Then

```

```

If v_Data = 1 Then
    txt13Amar.Visible = False
    txt13Amar2.Visible = True
    txt13Verde.Visible = False
    txt13Verde2.Visible = False
    f_Pos_Imagen img13, V, 13
    img13.Picture = imgCAmar.Picture
End If
'f_green
ElseIf ClientHandles(x) = 12 Then
    If v_Data = 1 Then
        txt13Amar.Visible = False
        txt13Amar2.Visible = False
        txt13Verde.Visible = False
        txt13Verde2.Visible = True
        f_Pos_Imagen img13, H, 13
        img13.Picture = imgCVerde.Picture
    End If
End If
Next x
End Sub

```

```

*****
*'Nome da Função:Timers_DataChange
'Caso de uso:Receber Estado dos Timers
'Responsável:Fagner Ghingaro
'Entrada:ByVal TransactionID As Long, ByVal NumItems As Long,
'ClientHandles() As Long, ItemValues() As Variant,
'ItemValues() As Long, ItemValues() As Date
'Saída:Nenhuma
'Observação:Entradas fixas enviadas pelo CLP, não podem ser alteradas
*****
*
```

```

Private Sub Timers_DataChange(ByVal TransactionID As Long, ByVal NumItems As
Long, ClientHandles() As Long, ItemValues() As Variant, Qualities() As Long,
TimeStamps() As Date)
Dim v_Data As String
For x = 1 To NumItems
    If ClientHandles(x) = 3 Then
        Timer_A = ItemValues(x) 'pre_a_green
    ElseIf ClientHandles(x) = 4 Then
        txt11Verde.Text = ItemValues(x) 'acc_a_green
    ElseIf ClientHandles(x) = 6 Then
        txt11Amar.Text = ItemValues(x) 'acc_a_yellow
    ElseIf ClientHandles(x) = 9 Then
        Timer_B = ItemValues(x) 'pre_b_green
    End If
Next x

```

```

ElseIf ClientHandles(x) = 10 Then
    txt11Verde2.Text = ItemValues(x) 'acc_b_green
ElseIf ClientHandles(x) = 12 Then
    txt11Amar2.Text = ItemValues(x) 'acc_b_yellow
ElseIf ClientHandles(x) = 15 Then
    Timer_C = ItemValues(x) 'acc_c_green
ElseIf ClientHandles(x) = 16 Then
    txt12Verde.Text = ItemValues(x) 'acc_c_green
ElseIf ClientHandles(x) = 18 Then
    txt12Amar.Text = ItemValues(x) 'acc_b_green
ElseIf ClientHandles(x) = 22 Then
    Timer_D = ItemValues(x) 'pre_d_green
ElseIf ClientHandles(x) = 22 Then
    txt12Verde2.Text = ItemValues(x) 'acc_d_green
ElseIf ClientHandles(x) = 24 Then
    txt12Amar2.Text = ItemValues(x) 'acc_d_yellow
ElseIf ClientHandles(x) = 27 Then
    Timer_E = ItemValues(x) 'pre_e_green
ElseIf ClientHandles(x) = 28 Then
    txt13Verde.Text = ItemValues(x) 'acc_e_green
ElseIf ClientHandles(x) = 30 Then
    txt13Amar.Text = ItemValues(x) 'acc_e_yellow
ElseIf ClientHandles(x) = 33 Then
    Timer_F = ItemValues(x) 'pre_f_green
ElseIf ClientHandles(x) = 34 Then
    txt13Verde2.Text = ItemValues(x) 'acc_f_green
ElseIf ClientHandles(x) = 36 Then
    txt13Amar2.Text = ItemValues(x) 'acc_f_yellow
End If
Next x
End Sub

```

```

*****
*!
Nome da Função: f_SetData
' Caso de uso: Não se aplica
' Descrição: Mapeamento das variáveis do CLP associando uma variável
'           interna do CLP a uma variável do sistema.
' Responsável: Fagner Ghingaro
' Entrada: Nenhuma
' Saída: Nenhuma
' Observação:
*****
*!
Public Sub f_SetData()

```

```
Set Timers = TimerServer.OPCGroups.Add("MyOPCData")
```

'entreverde
Timers.OPCItems.AddItem "[UNTITLED]T4:0.PRE,L1,C1", 1
Timers.OPCItems.AddItem "[UNTITLED]T4:0.ACC,L1,C1", 2
'a_verde
Timers.OPCItems.AddItem "[UNTITLED]T4:1.PRE,L1,C1", 3
Timers.OPCItems.AddItem "[UNTITLED]T4:1.ACC,L1,C1", 4
'a_yellow
Timers.OPCItems.AddItem "[UNTITLED]T4:2.PRE,L1,C1", 5
Timers.OPCItems.AddItem "[UNTITLED]T4:2.ACC,L1,C1", 6
'entreverde
Timers.OPCItems.AddItem "[UNTITLED]T4:3.PRE,L1,C1", 7
Timers.OPCItems.AddItem "[UNTITLED]T4:3.ACC,L1,C1", 8
'b_green
Timers.OPCItems.AddItem "[UNTITLED]T4:4.PRE,L1,C1", 9
Timers.OPCItems.AddItem "[UNTITLED]T4:4.ACC,L1,C1", 10
'b_yellow
Timers.OPCItems.AddItem "[UNTITLED]T4:5.PRE,L1,C1", 11
Timers.OPCItems.AddItem "[UNTITLED]T4:5.ACC,L1,C1", 12
'entreverde
Timers.OPCItems.AddItem "[UNTITLED]T4:6.PRE,L1,C1", 13
Timers.OPCItems.AddItem "[UNTITLED]T4:6.ACC,L1,C1", 14
'c_green
Timers.OPCItems.AddItem "[UNTITLED]T4:7.PRE,L1,C1", 15
Timers.OPCItems.AddItem "[UNTITLED]T4:7.ACC,L1,C1", 16
'c_yellow
Timers.OPCItems.AddItem "[UNTITLED]T4:8.PRE,L1,C1", 17
Timers.OPCItems.AddItem "[UNTITLED]T4:8.ACC,L1,C1", 18
'entreverde
Timers.OPCItems.AddItem "[UNTITLED]T4:9.PRE,L1,C1", 19
Timers.OPCItems.AddItem "[UNTITLED]T4:9.ACC,L1,C1", 20
'd_green
Timers.OPCItems.AddItem "[UNTITLED]T4:10.PRE,L1,C1", 21
Timers.OPCItems.AddItem "[UNTITLED]T4:10.ACC,L1,C1", 22
'd_yellow
Timers.OPCItems.AddItem "[UNTITLED]T4:11.PRE,L1,C1", 23
Timers.OPCItems.AddItem "[UNTITLED]T4:11.ACC,L1,C1", 24
'entreverde
Timers.OPCItems.AddItem "[UNTITLED]T4:12.PRE,L1,C1", 24
Timers.OPCItems.AddItem "[UNTITLED]T4:12.ACC,L1,C1", 26
'e_green
Timers.OPCItems.AddItem "[UNTITLED]T4:13.PRE,L1,C1", 27
Timers.OPCItems.AddItem "[UNTITLED]T4:13.ACC,L1,C1", 28
'e_yellow
Timers.OPCItems.AddItem "[UNTITLED]T4:14.PRE,L1,C1", 29
Timers.OPCItems.AddItem "[UNTITLED]T4:14.ACC,L1,C1", 30
'entreverde
Timers.OPCItems.AddItem "[UNTITLED]T4:15.PRE,L1,C1", 31

```

Timers.OPCItems.AddItem "[UNTITLED]T4:15.ACC,L1,C1", 32
'f_green
Timers.OPCItems.AddItem "[UNTITLED]T4:16.PRE,L1,C1", 33
Timers.OPCItems.AddItem "[UNTITLED]T4:16.ACC,L1,C1", 34
'f_yellow
Timers.OPCItems.AddItem "[UNTITLED]T4:17.PRE,L1,C1", 35
Timers.OPCItems.AddItem "[UNTITLED]T4:17.ACC,L1,C1", 36

Timers.IsSubscribed = True

Set Outputs = OutputServer.OPCGroups.Add("MyOPCData")
'a_yellow
Outputs.OPCItems.AddItem "[UNTITLED]O:0/1,L1,C1", 1
'a_green
Outputs.OPCItems.AddItem "[UNTITLED]O:0/2,L1,C1", 2
'b_yellow
Outputs.OPCItems.AddItem "[UNTITLED]O:0/7,L1,C1", 3
'b_green
Outputs.OPCItems.AddItem "[UNTITLED]O:0/11,L1,C1", 4
'c_yellow
Outputs.OPCItems.AddItem "[UNTITLED]O:0/5,L1,C1", 5
'c_green
Outputs.OPCItems.AddItem "[UNTITLED]O:0/6,L1,C1", 6
'd_yellow
Outputs.OPCItems.AddItem "[UNTITLED]O:0/17,L1,C1", 7
'd_green
Outputs.OPCItems.AddItem "[UNTITLED]O:0/18,L1,C1", 8
'e_yellow
Outputs.OPCItems.AddItem "[UNTITLED]O:0/9,L1,C1", 9
'e_green
Outputs.OPCItems.AddItem "[UNTITLED]O:0/10,L1,C1", 10
'f_yellow
Outputs.OPCItems.AddItem "[UNTITLED]O:0/20,L1,C1", 11
'f_green
Outputs.OPCItems.AddItem "[UNTITLED]O:0/21,L1,C1", 12

Outputs.IsSubscribed = True

Set Inputs = InputServer.OPCGroups.Add("MyOPCData")
'botoeira_1
Inputs.OPCItems.AddItem "[UNTITLED]I:0/1,L1,C1", 1
'botoeira_2
Inputs.OPCItems.AddItem "[UNTITLED]I:0/2,L1,C1", 2
'botoeira_3
Inputs.OPCItems.AddItem "[UNTITLED]I:0/3,L1,C1", 3

Inputs.IsSubscribed = True

```

```
End Sub
```

```
*****
```

```
*
```

```
'Nome da Função: f_SetServer
```

```
' Caso de uso: Não se aplica
```

```
' Descrição: Configuração das variáveis dos Servidores para a Conexão
```

```
' Responsável: Fagner Ghingaro
```

```
' Entrada: Nenhuma
```

```
' Saída: Nenhuma
```

```
' Observação:
```

```
*****
```

```
*
```

```
Public Sub f_SetServer()
```

```
Set TimerServer = New OPCServer
```

```
TimerServer.Connect "RSLinx OPC Server"
```

```
Set OutputServer = New OPCServer
```

```
OutputServer.Connect "RSLinx OPC Server"
```

```
Set InputServer = New OPCServer
```

```
InputServer.Connect "RSLinx OPC Server"
```

```
End Sub
```

```
*****
```

```
*
```

```
'Nome da Função: f_ServerUnload
```

```
' Caso de uso: Não se aplica
```

```
' Descrição: Configuração das variáveis dos Servidores para a Desconexão
```

```
' Responsável: Fagner Ghingaro
```

```
' Entrada: Nenhuma
```

```
' Saída: Nenhuma
```

```
' Observação:
```

```
*****
```

```
*
```

```
Public Sub f_ServerUnload()
```

```
Set Timers = Nothing
```

```
TimerServer.Disconnect
```

```
Set Outputs = Nothing
```

```
OutputServer.Disconnect
```

```
Set Inputs = Nothing
```

```
InputServer.Disconnect
```

```
End Sub
```

```

*****
*'
'Nome da Função: f_PosObjetos
' Caso de uso: Não se aplica
'   Descrição: Posicionamento e configuração dos objetos da interface gráfica
'   dentro do forms
' Responsável: Fagner Ghingaro
' Entrada: Nenhuma
' Saída: Nenhuma
' Observação:
*****'
*' Private Sub f_PosObjetos()
With txt11Amar
    .BackColor = v_Amarelo
    .Left = txt11.Left
    .Top = txt11.Top
    .Width = txt11.Width
    .Height = txt11.Height
    .Visible = False
End With

With txt11Amar2
    .BackColor = v_Amarelo
    .Left = txt11.Left
    .Top = txt11.Top
    .Width = txt11.Width
    .Height = txt11.Height
    .Visible = False
End With

With txt11Verde
    .BackColor = v_Amarelo
    .Left = txt11.Left
    .Top = txt11.Top
    .Width = txt11.Width
    .Height = txt11.Height
    .Visible = False
End With

With txt11Verde2
    .BackColor = v_Amarelo
    .Left = txt11.Left
    .Top = txt11.Top
    .Width = txt11.Width
    .Height = txt11.Height
    .Visible = False

```



End With

With txt12Amar

```
.BackColor = v_Amarelo  
.Left = txt12.Left  
.Top = txt12.Top  
.Width = txt12.Width  
.Height = txt12.Height  
.Visible = False
```

End With

With txt12Amar2

```
.BackColor = v_Amarelo  
.Left = txt12.Left  
.Top = txt12.Top  
.Width = txt12.Width  
.Height = txt12.Height  
.Visible = False
```

End With

With txt12Verde

```
.BackColor = v_Amarelo  
.Left = txt12.Left  
.Top = txt12.Top  
.Width = txt12.Width  
.Height = txt12.Height  
.Visible = False
```

End With

With txt12Verde2

```
.BackColor = v_Amarelo  
.Left = txt12.Left  
.Top = txt12.Top  
.Width = txt12.Width  
.Height = txt12.Height  
.Visible = False
```

End With

With txt13Amar

```
.BackColor = v_Amarelo  
.Left = txt13.Left  
.Top = txt13.Top  
.Width = txt13.Width  
.Height = txt13.Height  
.Visible = False
```

End With

With txt13Amar2

```
.BackColor = v_Amarelo  
.Left = txt13.Left  
.Top = txt13.Top  
.Width = txt13.Width  
.Height = txt13.Height  
.Visible = False
```

End With

With txt13Verde

```
.BackColor = v_Amarelo  
.Left = txt13.Left  
.Top = txt13.Top  
.Width = txt13.Width  
.Height = txt13.Height  
.Visible = False
```

End With

With txt13Verde2

```
.BackColor = v_Amarelo  
.Left = txt13.Left  
.Top = txt13.Top  
.Width = txt13.Width  
.Height = txt13.Height  
.Visible = False
```

End With

With txt21Amar

```
.BackColor = v_Amarelo  
.Left = txt21.Left  
.Top = txt21.Top  
.Width = txt21.Width  
.Height = txt21.Height  
.Visible = False
```

End With

With txt21Amar2

```
.BackColor = v_Amarelo  
.Left = txt21.Left  
.Top = txt21.Top  
.Width = txt21.Width  
.Height = txt21.Height  
.Visible = False
```

End With

With txt21Verde

```
.BackColor = v_Amarelo
```

```
.Left = txt21.Left  
.Top = txt21.Top  
.Width = txt21.Width  
.Height = txt21.Height  
.Visible = False  
End With
```

With txt21Verde2

```
.BackColor = v_Amarelo  
.Left = txt21.Left  
.Top = txt21.Top  
.Width = txt21.Width  
.Height = txt21.Height  
.Visible = False  
End With
```

With txt22Amar

```
.BackColor = v_Amarelo  
.Left = txt22.Left  
.Top = txt22.Top  
.Width = txt22.Width  
.Height = txt22.Height  
.Visible = False  
End With
```

With txt22Amar2

```
.BackColor = v_Amarelo  
.Left = txt22.Left  
.Top = txt22.Top  
.Width = txt22.Width  
.Height = txt22.Height  
.Visible = False  
End With
```

With txt22Verde

```
.BackColor = v_Amarelo  
.Left = txt22.Left  
.Top = txt22.Top  
.Width = txt22.Width  
.Height = txt22.Height  
.Visible = False  
End With
```

With txt22Verde2

```
.BackColor = v_Amarelo  
.Left = txt22.Left  
.Top = txt22.Top
```

```
.Width = txt22.Width  
.Height = txt22.Height  
.Visible = False  
End With
```

```
With txt23Amar  
.BackColor = v_Amarelo  
.Left = txt23.Left  
.Top = txt23.Top  
.Width = txt23.Width  
.Height = txt23.Height  
.Visible = False  
End With
```

```
With txt23Amar2  
.BackColor = v_Amarelo  
.Left = txt23.Left  
.Top = txt23.Top  
.Width = txt23.Width  
.Height = txt23.Height  
.Visible = False  
End With
```

```
With txt23Verde  
.BackColor = v_Amarelo  
.Left = txt23.Left  
.Top = txt23.Top  
.Width = txt23.Width  
.Height = txt23.Height  
.Visible = False  
End With
```

```
With txt23Verde2  
.BackColor = v_Amarelo  
.Left = txt23.Left  
.Top = txt23.Top  
.Width = txt23.Width  
.Height = txt23.Height  
.Visible = False  
End With  
End Sub
```

```
*****  
*!  
'Nome da Função: f_Altera_Verde  
' Caso de uso: Alterar Tempo de Verde  
' Descrição: Posicionamento do Text Box para digitação da alteração do tempo verde
```

```

' Entrada: v_Text = control referente ao text box onde será feito a alteração
' keyascii = valor asc do digito que foi digitado
' Saída: Nenhuma
' Observação:
'*****
'*'
Function f_Altera_Verde(ByVal v_Text As Control, KeyAscii As Integer)
With txtVerde
    .Left = v_Text.Left + 1000
    .Top = v_Text.Top
    .Visible = True
    .Text = Chr(KeyAscii)
    .SetFocus
End With
End Function

'*****
'*'
'Nome da Função: f_Pos_Imagen
'Caso de uso: Não se aplica
'Descrição: Posiciona a imagem para uma exibição gráfica correta
'Responsável: Fagner Ghingaro
'Entrada: v_Imagen = control referente a imagem que será alterada
'Saída: Nenhuma
'Observação:
'*****
'*'
Function f_Pos_Imagen(ByVal v_Imagen As Control, Pos As String, Cruz As Integer)
If Cruz = 11 Then
    If Pos = "V" Then
        With v_Imagen
            .Height = 735
            .Left = 540
            .Top = 2330
            .Width = 1335
        End With
    ElseIf Pos = "H" Then
        With v_Imagen
            .Height = 1335
            .Left = 840
            .Top = 2040
            .Width = 735
        End With
    End If
ElseIf Cruz = 12 Then
    If Pos = "V" Then

```



```

With v_Imagen
    .Height = 735
    .Left = 4620
    .Top = 2325
    .Width = 1335
End With
ElseIf Pos = "H" Then
    With v_Imagen
        .Height = 1335
        .Left = 4920
        .Top = 2040
        .Width = 735
    End With
End If
ElseIf Cruz = 13 Then
    If Pos = "V" Then
        With v_Imagen
            .Height = 735
            .Left = 8220
            .Top = 2325
            .Width = 1335
        End With
    ElseIf Pos = "H" Then
        With v_Imagen
            .Height = 1335
            .Left = 8520
            .Top = 2040
            .Width = 735
        End With
    End If
ElseIf Cruz = 21 Then
    If Pos = "V" Then
        With v_Imagen
            .Height = 735
            .Left = 540
            .Top = 6070
            .Width = 1335
        End With
    ElseIf Pos = "H" Then
        With v_Imagen
            .Height = 1335
            .Left = 840
            .Top = 5880
            .Width = 735
        End With
    End If
ElseIf Cruz = 22 Then

```

```

If Pos = "V" Then
    With v_Imagen
        .Height = 735
        .Left = 4620
        .Top = 6075
        .Width = 1335
    End With
ElseIf Pos = "H" Then
    With v_Imagen
        .Height = 1335
        .Left = 4920
        .Top = 5880
        .Width = 735
    End With
End If
ElseIf Cruz = 23 Then
    If Pos = "V" Then
        With v_Imagen
            .Height = 735
            .Left = 8220
            .Top = 6075
            .Width = 1335
        End With
    Elseif Pos = "H" Then
        With v_Imagen
            .Height = 1335
            .Left = 8520
            .Top = 5880
            .Width = 735
        End With
    End If
End If
End Function

```

```
*****
*
```

```

'Nome da Função: f_Envia_Estado
'Caso de uso: Envia Alteracao de Estado
'Responsável: Fagner Ghingaro
'Entrada: x = inteiro que referencia o GroupItema que será modificado
'          v_Valor = string que referencia o valor que está sendo enviado
'          v_Log = control que referencia o objeto que receberá o log
'Saída: Nenhuma
'Observação:
*****
```

```
*****
*
```

```
Public Sub f_Envia_Estado(x As Integer, v_Valor As String, ByVal v_Log As Control)
```

```

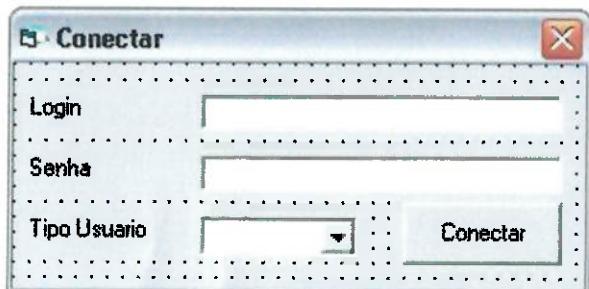
Dim arHandles(1 To 1) As Long
Dim arValues(1 To 1) As Variant
Dim arErrors() As Long

arHandles(x) = Timers.OPCItems(x).ServerHandle

On Error GoTo TextError
    arValues(x) = CInt(v_Valor)
    v_Log.AddItem "Início", 0
    Timers.SyncWrite 1, arHandles(), arValues(), arErrors()
    v_Log.AddItem "Fim", 0
    If arErrors(x) <> 0 Then
        v_Log.AddItem "arerror:" & arErrors(1), 0
    End If
Exit Sub

TextError:
    v_Log.AddItem "Texterror:Write not performed", 0
    Exit Sub
End Sub

```



```

Option Explicit
Dim v_Tipo As String
Private Sub Command1_Click()
If frmPrincipal.Acao = "Conectar" Then
    If Trim(txtLogin.Text) Is Not Null And Trim(cboTipo.Text) Is Not Null Then
        f_Valida_Senha
    End If
ElseIf frmPrincipal.Acao = "Adicionar" Then
    End If
End Sub
Private Sub Form_Load()
txtLogin.Text = ""
cboTipo.Text = ""
cmdAcao.Caption = frmPrincipal.Acao
If frmPrincipal.Acao = "Conectar" Then

```

```

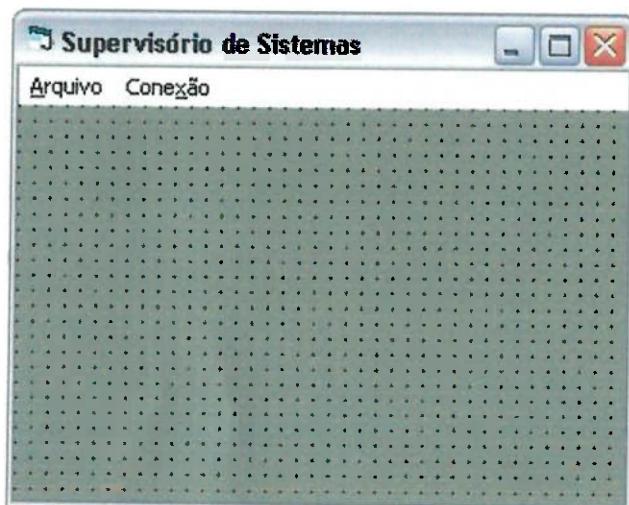
txtSenha.Locked = False
Label2.Visible = False
cboTipo.Visible = False
ElseIf frmPrincipal.Acao = "Adicionar" Then
    txtSenha.Locked = True
End If
End Sub
*****
*!
'Nome da Função: f_Valida_Senha
' Caso de uso: Conectar no sistema
' Responsável: Fagner Ghingaro
' Entrada: Nenhuma
' Saída: Nenhuma
' Observação:
*****
*!
Private Sub f_Valida_Senha()
If UCASE(txtLogin.Text) = "OPERADOR" Or UCASE(txtLogin.Text) =
"CONTROLADOR" Then
    If UCASE(txtLogin.Text) = "OPERADOR" And UCASE(txtSenha.Text) = "OPERADOR"
Then
        v_Tipo = "OPERADOR"
        f_Acesso
        Unload Me
    ElseIf UCASE(txtLogin.Text) = "CONTROLADOR" And UCASE(txtSenha.Text) =
"CONTROLADOR" Then
        v_Tipo = "CONTROLADOR"
        f_Acesso
        Unload Me
    Else
        v_Tipo = ""
        MsgBox "A senha é inválida! Digite Novamente.", vbCritical, "ERRO"
        txtSenha.Text = ""
        txtSenha.SetFocus
    End If
Else
    frmPrincipal.f_Desabilitar
    MsgBox "O login é inválido! Digite Novamente.", vbCritical, "ERRO"
    txtLogin.Text = ""
    txtSenha.Text = ""
    txtLogin.SetFocus
End If
End Sub
*****
*!
'Nome da Função: f_Acesso

```

' Caso de uso: Tratar acesso dos usuários
' Responsável: Fagner Ghingaro
' Entrada: Nenhuma
' Saída: Nenhuma
' Observação:

*'

```
Private Sub f_Acesso()
If v_Tipo = "OPERADOR" Then
    frmPrincipal.f_Operador
ElseIf v_Tipo = "CONTROLADOR" Then
    frmPrincipal.f_Controlador
Else
    frmPrincipal.f_Desabilitar
End If
End Sub
```



```
Public Usuário As String
Public Ação As String
```

```
Private Sub mnuSair_Click()
End
End Sub
```

```
Public Sub f_Operador()
mnuConsultar.Enabled = True
mnuConectar.Enabled = False
mnuDesconectar.Enabled = True
mnuUsuário.Enabled = False
Usuário = "OPERADOR"
End Sub
```

```
Public Sub f_Controlador()
mnuConsultar.Enabled = True
mnuConectar.Enabled = False
mnuDesconectar.Enabled = True
mnuUsuario.Enabled = False
Usuario = "CONTROLADOR"
End Sub
```

```
Private Sub mnuUsuario_Click()
frmUsuario.Show
Acao = "Adicionar"
End Sub
```

```
Private Sub MDIForm_Load()
f_Desabilitar
End Sub
```

```
Private Sub mnuConectar_Click()
Acao = "Conectar"
End Sub
```

```
Private Sub mnuConsultar_Click()
frmArea.Show
End Sub
```

```
Private Sub mnuDesconectar_Click()
f_Desabilitar
End Sub
```

```
*****
*' 
'Nome da Função: f_Acesso
' Caso de uso: Desconectar do Sistema
' Responsável: Fagner Ghingaro
' Entrada: Nenhuma
' Saída: Nenhuma
' Observação:
*****
```

```
*****
' 
Public Sub f_Desabilitar()
mnuConsultar.Enabled = False
mnuConectar.Enabled = True
mnuDesconectar.Enabled = False
mnuUsuario.Enabled = False
Usuario = "DESCONECTADO"
End Sub
```