

UNIVERSIDADE ESTADUAL DO OESTE DO PARANÁ
UNIOESTE – CAMPUS DE FOZ DO IGUAÇU

GUARDIAN COMPUTER

CRISTIANO LEHRER

FOZ DO IGUAÇU – 1998

GUARDIAN COMPUTER

UNIOESTE – Universidade Estadual do Oeste do Paraná

Campus de Foz do Iguaçu

Centro de Ciências Exatas e Tecnológicas

Departamento de Ciência da Computação

Curso de Ciência da Computação

GUARDIAN COMPUTER

Autor: Cristiano Lehrer

Orientadores: Prof. Jorge Habib Hanna El Khouri, Ms.

Prof. José Luíz Pereira dos Santos

Projeto de Graduação apresentado ao curso de
Ciência da Computação da UNIOESTE –
Universidade Estadual do Oeste do Paraná
como parte dos requisitos exigidos para a
obtenção do título de Bacharel em Ciência da
Computação.

Dezembro 1998

“Vê mais longe a gaivota que
voa mais alto”.

RICHARD BACH

Aos meus familiares.

Agradeço a todos que de uma ou outra forma auxiliaram no desenvolvimento deste trabalho, tornando-o uma realidade ao invés de uma pretensão.

RESUMO

A segurança residencial tem incorporado mais intensamente recursos de automação. As grandes empresas de produtos de informática investem maciçamente neste ramo, em busca de novos clientes e serviços.

Os clientes desejam trazer cada dia mais conforto e tranquilidade para dentro de suas casas, tornando-as verdadeiros refúgios do modo de vida atual.

Guardian Computer propõem ser o guardião da residência, protegendo-a e controlando-a em busca de uma melhor qualidade de vida para seus habitantes.

O usuário interage com o sistema através de módulos de interface, conectados ao computador central, onde está localizado o núcleo do sistema, responsável pelo controle dos módulos dos equipamentos, controlando assim a segurança da residência.

ABSTRACT

Home security has incorporated more intensively automation resources. Big companies of computer products have been investing heavily in this area, searching for new clients and services.

Clients hope to get each day more comfort and tranquility for their homes, making them real shelters from the modern way of living.

Guardian Computer proposes to be the home guardian, protecting it and controlling it, in search of a best quality of life for their inhabitants.

The user interacts with the system through the interface modules, connected to the central computer, where the kernel system is located. This system is responsible for the modules control of the equipments, controlling that way home security.

SUMÁRIO

1. INTRODUÇÃO.....	1
1.1. AUTOMAÇÃO RESIDENCIAL.....	1
1.1.1. Segurança.....	1
1.1.2. Iluminação.....	2
1.1.3. Condicionamento ambiental.....	2
1.1.4. Setor elétrico.....	3
1.1.5. Setor hidráulico.....	3
1.1.6. Aquisição de imagens.....	4
1.1.7. Aparelhos eletrônicos.....	4
1.2. OBJETIVO.....	4
1.3. ORGANIZAÇÃO DO TRABALHO.....	5
2. ARQUITETURA PROPOSTA.....	6
2.1. ARQUITETURA FÍSICA.....	6
2.2. ARQUITETURA LÓGICA.....	9
2.3. DISPOSITIVOS SUPORTADOS PELO SISTEMA.....	10
2.3.1. Sensor de presença.....	11
2.3.2. Sensor de incêndio.....	12
2.3.3. Máquina fotográfica digital.....	12
2.3.4. Cortina.....	12
2.3.5. Porta.....	13
2.3.6. Lâmpada.....	13
2.3.7. Sensor de ruído.....	13
2.3.8. Tomada de força.....	14
2.3.9. Sensor infravermelho.....	14
2.3.10. Interruptor.....	14
2.3.11. Torneira.....	15
2.3.12. Janela.....	15
3. KERNEL.....	17
3.1. COMUNICAÇÃO COM AS UTR'S.....	18
3.1.1. Estrutura dos pacotes.....	19
3.1.2. Protocolo.....	20
3.1.3. Comunicação serial.....	21

3.1.4. <i>Buffer de pacotes</i>	22
3.2. COMUNICAÇÃO COM A ESTAÇÃO	22
3.2.1. <i>Estrutura dos pacotes</i>	22
3.2.2. <i>Protocolo</i>	24
3.2.3. <i>Comunicação TCP/IP</i>	24
3.2.4. <i>Buffer de pacotes</i>	25
3.3. INTERPRETADOR DE REGRAS	25
3.4. GERENCIADOR CENTRAL	27
3.4.1. <i>Arquivos de Log</i>	28
3.4.2. <i>Processo de comunicação</i>	29
3.4.3. <i>Configurações</i>	30
3.4.4. <i>Funcionamento</i>	30
4. BUILDER & VIEWER	32
4.1. BUILDER	32
4.1.1. <i>Projeto</i>	33
4.1.2. <i>Paleta</i>	33
4.1.3. <i>Editor de propriedades</i>	34
4.1.4. <i>Formação da Tela</i>	35
4.2. VIEWER	36
5. CASO BASE	39
5.1. BUILDER	41
5.2. KERNEL	42
5.3. VIEWER	43
6. CONCLUSÃO	46
7. REFERÊNCIAS BIBLIOGRÁFICAS	49

LISTA DE FIGURAS

Figura 2.1 : Arquitetura física.....	6
Figura 2.2 : Esquema de comunicação entre o Master e a UTR	7
Figura 2.3 : Diagrama da Arquitetura Lógica.....	9
Figura 2.4 : Sensor de presença	11
Figura 2.5 : Sensor de incêndio	12
Figura 2.6 : Máquina fotográfica digital.....	12
Figura 2.7 : Cortina.....	12
Figura 2.8 : Porta	13
Figura 2.9 : Lâmpada.....	13
Figura 2.10 : Sensor de ruído.....	13
Figura 2.11 : Tomada de força.....	14
Figura 2.12 : Sensor infravermelho	14
Figura 2.13 : Interruptor	14
Figura 2.14 : Torneira.....	15
Figura 2.15 : Janela.....	15
Figura 2.16 : Árvore de componentes que representam os equipamentos.	16
Figura 3.1 : Diagrama lógico do Kernel.....	17
Figura 3.2 : Diagrama lógico da Comunicação com as UTR's	18
Figura 3.3 : Pacote de Dados da COM	19
Figura 3.4 : Pacote de controle da COM	20
Figura 3.5 : Pacote de confirmação da COM	20
Figura 3.6 : Protocolo de comunicação entre o Master e a UTR.....	21
Figura 3.7 : Diagrama lógico da Comunicação com as Estações.....	23
Figura 3.8 : Pacote de Dados do TCP.....	23
Figura 3.9 : Pacote de Controle do TCP	24
Figura 3.10 : Pacote de Confirmação do TCP.....	24
Figura 3.11 : Protocolo de comunicação entre o Viewer e o Kernel.....	24
Figura 3.12 : Diagrama lógica do Interpretador de Regras	25
Figura 3.13 : Estrutura de uma regra	27

Figura 3.14: Diagrama lógico do Gerenciador Central do Kernel	28
Figura 3.15 : Exemplo das trocas de pacotes entre as máquinas	29
Figura 3.16 : Conexões do Kernel	30
Figura 3.17 : Tela de Setup do Kernel.....	31
Figura 4.1 : Diagrama lógico do Builder.....	32
Figura 4.2 : Paleta de componentes	33
Figura 4.3 : Editor de Propriedades	34
Figura 4.4 : Editor de regras	34
Figura 4.5 : Grade de destaque	35
Figura 4.6 : Exemplo de tela.....	35
Figura 4.7 : Diagrama lógico do Viewer	36
Figura 4.8 : Tela de Log do Sistema.....	37
Figura 4.9 : Tela de Setup do Viewer.....	38
Figura 5.1 : Planta Baixa	43
Figura 5.2 : Cozinha	44
Figura 5.3 : Banheiro	44
Figura 5.4 : Quarto.....	45

1. Introdução

As inovações tecnológicas na área de informática e sistemas digitais estão possibilitando o aparecimento de várias aplicações até então não imagináveis, que marcam algumas vezes pelo ineditismo, mas rapidamente são aceitas pelas pessoas, incorporando-se ao dia a dia, e tornado-se um condicionante do conforto e da praticidade essenciais atualmente, assim como o relógio digital, o forno de microondas, os microcomputadores, a Internet e atualmente em fase de expansão, a automação residencial.

1.1. Automação Residencial

Qual a pessoa que não desejaria possuir a residência da família Jetsons, do desenho animado The Jetsons, toda automatizada e computadorizada, proporcionando uma residência com alto nível de conforto, praticidade e segurança. Infelizmente, tais residências ainda são um sonho futurista, mas já existem sistemas que provêem uma certa automação, como por exemplo as que utilizam a arquitetura X-10.

Dentre as áreas cobertas pela automação residencial, estão o controle da segurança, da iluminação, do condicionamento ambiental, do setor elétrico, do setor hidráulico, aquisição de imagens, e em alguns casos o controle de aparelhos eletrônicos, como secretárias eletrônicas, televisores, rádios entre outros.

1.1.1. Segurança

Com a atual falta de segurança encontrada nas grandes cidades, a utilização de sistemas de segurança tornou-se essencial, sendo uma das áreas da automação residencial mais desenvolvidas atualmente.

O monitoramento da segurança é obtido através da instalação de uma série de sensores diferentes espalhados por diversos pontos da residência, monitorando os estados de portas, janelas, e a detecção de estranhos, incêndio e inundações. Os sensores podem ser magnéticos,

infravermelhos, sísmicos, vibração, microondas, sonoros, temperatura entre outros disponíveis.

Quando um sensor dispara, procedimentos devem ser executados a fim de agir sobre a ocorrência, como por exemplo, disparar um sinal sonoro imitando uma sirene ou o latido de um cachorro, simular a presença de pessoas na residência, informar centrais de monitoramento sobre a ocorrência entre outros procedimentos que podem ser especificados.

1.1.2. Iluminação

Permite o acionamento automático de lâmpadas quando se entra em determinados ambientes e o seu desligamento quando ausentar-se do mesmo, através dos sensores de presença instalados nos ambientes que indicam ao sistema a presença ou não de pessoas no mesmo.

Através de sensores que captam a intensidade luminosa, o sistema pode permitir o ajuste de uma claridade uniforme para cada ambiente, economizando energia elétrica através da melhor utilização da luz ambiente.

As lâmpadas também podem ser acessas e apagadas através de uma programação de horários ou do nível de claridade do ambiente, simulando a presença de pessoas na residência.

1.1.3. Condicionamento ambiental

Através de sensores que captam a temperatura ambiente e do controle de equipamentos de refrigeração e aquecimento, pode-se controlar a temperatura ambiente de forma a mantê-la sempre num mesmo nível. Os aparelhos de ar condicionado mais modernos já realizam esta tarefa sozinhos com o uso de termostatos, mas integrando-os num sistema, consegue-se um melhor aproveitamento dos equipamentos.

Por exemplo, o ar condicionado pode ser acionado remotamente ou em horários programados, e as portas e janelas do ambiente podem ser programados para se fecharem quando o ar condicionado entrar em funcionamento.

1.1.4. Setor elétrico

Permite a diminuição do consumo de energia elétrica, através da utilização racional da eletricidade, obtido com o desligamento dos aparelhos que não estejam em uso efetivo, como lâmpadas acessas durante o dia entre outros desperdícios.

A segurança da residência aumenta com o controle da eletricidade, prevenindo acidentes elétricos com crianças e incêndios provocados por um curto-circuito, através do corte da carga das tomadas ou do ambiente.

Aparelhos eletrônicos simples cujo funcionamento está apenas limitado a presença ou ausência de energia, como cafeteiras elétricas, podem ser controlados através do controle da eletricidade, especificando-se o horário que as tomadas devem ser ligadas ou desligadas.

1.1.5. Setor hidráulico

A água potável é um dos recursos naturais mais valiosos para a humanidade, porque sem água nenhum ser vivo sobrevive. As fontes naturais de água estão nos últimos anos sendo poluídas, e custa muito caro limpar esta água para que torne-se apta ao consumo humano ou mesmo animal. Por essa razão o uso inteligente dos recursos hídricos é essencial nos tempos atuais, economizando-se os recursos naturais e minimizando-se os custos.

A automação neste setor restringe-se a racionar o consumo de água, através de sensores de proximidade, o sistema pode saber quando atuar nas torneiras, evitando-se deixar a torneira aberta enquanto escova-se os dentes, por exemplo.

Também é possível programar o sistema para irrigar o gramado, liberando apenas a quantidade de água especificado, verificar a existência de vazamentos através da monitoração do consumo, e se a residência possuir um sistema anti-incêndio, aciona-lo no caso de incêndio e fechando-se as demais torneiras.

1.1.6. Aquisição de imagens

Câmeras fotográficas digitais ou circuitos fechados de televisão (CFTV), disponibilizam a visualização da situação real da residência, através da conexão de um computador externo com o computador responsável pelo gerenciamento da residência.

Com a gravação das imagens, pode-se identificar casos de invasões ou controlar o acesso de pessoas, mantendo-se um controle do fluxo dos habitantes.

1.1.7. Aparelhos eletrônicos

Com a crescente popularização dos microcontroladores, microprocessadores desenvolvidos especialmente para o controle de outros equipamentos, sendo mais simples e baratos que os microprocessadores dos computadores, os equipamentos eletrônicos estão tornando-se cada dia mais inteligentes, como os televisores com bloqueio de canais e registros de recados, fornos de microondas que calculam automaticamente o tempo necessário para aquecer os alimentos, entre tantas outras áreas de atuação.

O problema para a automação residencial é integrar aparelhos eletrônicos ao sistema, porque a maioria deles não possui a capacidade de comunicar-se com outros equipamentos.

Diversas empresas estão tentando definir um padrão de comunicação para aparelhos, permitindo a sua comunicação com outros aparelhos ou com um sistema central. Dentro destes padrões que estão em desenvolvimento, encontra-se o protocolo Jini proposto pela Sun Microsystems, Inc.

1.2. Objetivo

Existem diversas áreas numa residência que podem ser automatizadas de forma a deixá-la quase que totalmente auto-suficiente. Existem atualmente no mercado diversos tipos de equipamentos para automatização, desde equipamentos físicos, como sensores, controladores, até software para o gerenciamento dos mesmos.

Guardian Computer se propõem a ser um sistema de automatização residencial computadorizado, integrando as diversas áreas que podem ser automatizadas em uma residência, utilizando-se de uma interface amigável com o usuário, que normalmente não domina a área de informática.

1.3. Organização do trabalho

O presente trabalho será dividido em cinco capítulos. O primeiro descrevendo a arquitetura proposta para solucionar o problema e as tarefas a serem automatizadas dentro de cada uma das áreas.

O segundo capítulo descreve detalhadamente o núcleo do sistema, onde são tomadas as decisões sobre o gerenciamento da residência.

O terceiro capítulo contém as ferramentas que auxiliam o usuário na implantação e supervisão do sistema.

O quarto capítulo descreve um cenário de atuação do sistema, com a demonstração dos resultados alcançados.

O quinto capítulo apresenta as conclusões sobre o desenvolvimento do sistema e os possíveis acréscimos a serem feitos para deixá-lo mais funcional.

2. Arquitetura Proposta

Como o sistema extrapola a barreira de um único computador, interagindo com outros equipamentos, dividiu-se a arquitetura em duas, arquitetura física e a arquitetura lógica a fim de facilitar a compreensão.

O sistema é implementado na linguagem de programação Delphi 2.0 da Inprise (antiga Borland), utilizando-se dos conceitos de orientação a objetos para facilitar o desenvolvimento, através do encapsulamento de certas funcionalidades em componentes da linguagem.

2.1. Arquitetura física

O sistema é composto por estações de interface com o usuário (Estações), que realizam a interface homem-máquina, uma estação de operação (Master), responsável pelo controle do sistema, e várias unidades terminais remotas (UTR) que controlam os equipamentos assistidos pelo sistema.

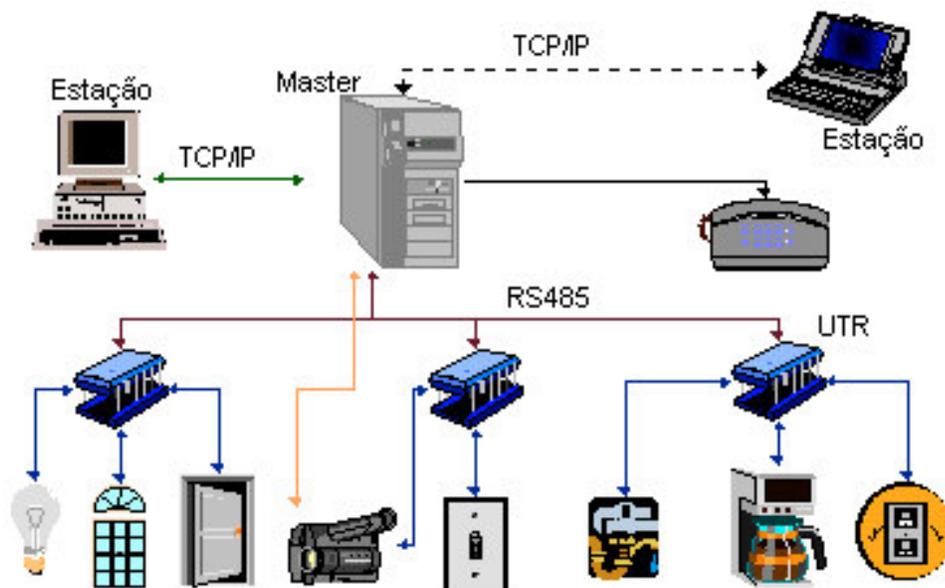


Figura 2.1 : Arquitetura física

O número de dispositivos instalados pode ser elevado, ocasionando um problema a conexão destes dispositivos em um único controlador. A conexão dos dispositivos em um controlador pode ser realizada de duas formas : ponto a ponto ou em rede. Para a conexão em rede é necessário a utilização de dispositivos capazes de gerenciarem o protocolo de comunicação, enquanto que na conexão ponto a ponto os dispositivos não são responsáveis pelo protocolo utilizado.

Em muitas residências, em especial as mais antigas, não existem tubulações suficientes para passar todos os cabos até o controlador, além do custo dos fios. Para contornar este problema, adotou-se a utilização de UTR's que são responsáveis pela aquisição dos dados provenientes dos dispositivos, bem como no envio das ações aos mesmos, não utilizando-se nesta fase dispositivos inteligentes capazes de serem interligados numa rede.

Cada UTR é responsável por um certo número de dispositivos e estão ligados ao Master. Exemplificando, se uma residência possuir 100 dispositivos monitorados, número fácil de ser alcançado atualmente, o Master iria receber 100 pares de fios a serem conectados às suas placas. Utilizando UTR's, e estimando que cada UTR suporte 20 dispositivos, iríamos precisar de 5 UTR, que estariam localizados perto dos dispositivos, e o Master precisaria de uma placa com apenas 5 entradas.

As UTR são baseadas em microcontroladores da família 8051, que foram escolhidos por serem encontrados facilmente no mercado, além da vasta documentação existente. Estes microcontroladores possuem uma frequência de clock de 12 MHz, trabalham com uma palavra de 8 bits e possuem uma porta serial e duas portas paralelas para comunicação. As portas paralelas são usadas na comunicação com os dispositivos e a porta serial é utilizada na comunicação com o Master, numa taxa de transferência de 9600 bps que pode ser aumentada conforme necessidade.

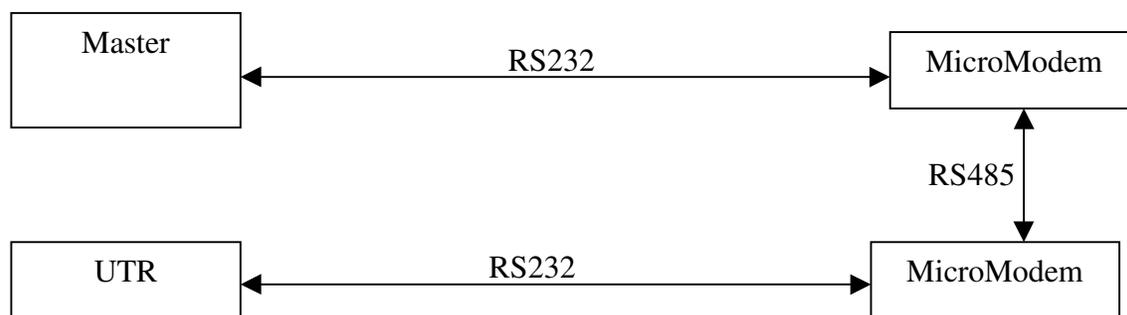


Figura 2.2 : Esquema de comunicação entre o Master e a UTR

Para realizar a transferência de dados entre as UTR's e o Master são utilizados micromodems, que possibilitam distâncias maiores de comunicação. A interface utilizada pelos micromodems é o RS485, uma interface simples, de fácil utilização e que exige um número reduzido de fios. Os micromodems são conectados as portas seriais das UTR's e do Master, que utilizam a interface RS232, cabendo aos micromodems a conversão entre RS232 / RS485 / RS232.

As imagens captadas pelas câmeras de vídeo ou por máquinas fotográficas digitais são transmitidas diretamente ao Master através da sua própria cablagem, evitando-se monopolizar as UTR's com o grande volume de dados a serem transmitidos. Os controles destes equipamentos continuam passando pelas UTR's.

No Master está localizado o núcleo do sistema, possuindo como principais atribuições:

- supervisionar e controlar a rede de UTR's, e conseqüentemente, os equipamentos assistidos pelo sistema;
- centralizar e consolidar os dados adquiridos pelo sistema;
- disponibilizar os dados para as Estações, possibilitando a interação entre o usuário e o sistema.

O Master é baseado em microcomputadores PC's, devendo ser uma máquina de dedicação exclusiva em virtude da carga de processamento e da ininterruptividade de seu funcionamento. Com a instalação de um modem conectado a uma linha telefônica ou de uma placa de rede, pode-se receber comandos remotamente das Estações, bem como informa-lhes a situação da residência e ainda enviar mensagens gravadas a telefones comuns.

As Estações são compostas por microcomputadores PC's, podendo ser o computador doméstico do usuário, seu computador de trabalho ou seu notebook. Nele são executadas todas as operações de interface homem-máquina com o sistema. Através das estações o usuário pode interagir com o sistema, programando novas tarefas, alterando os estados dos equipamentos e recebendo os relatórios de funcionamento do sistema.

O protocolo de rede TCP/IP é utilizado na comunicação entre o Master e as Estações, já que o usuário pode conectar-se ao sistema através de uma linha telefônica ou via rede. A utilização do protocolo TCP/IP consegue padronizar a comunicação entre os computadores, independentemente do meio utilizado para realizar a conexão, além de implementar uma entrega confiável dos dados.

2.2. Arquitetura Lógica

Cada um dos computadores utilizados na arquitetura física contém um aplicativo específico para realizar a tarefa pelo qual o equipamento ficou responsável. O sistema é composto por quatro aplicativos distintos que trabalham integrados:

- Software das UTR's: localizado nas UTR's, é o responsável pelo processamento local das informações e do protocolo de rede utilizado nas comunicações entre as UTR's e o Master;
- Kernel: localizado no Master, é o cérebro do sistema. É o responsável pelo gerenciamento do sistema, sendo o intermediador das requisições dos usuários aos dispositivos instalados;
- Viewer: localizado na Estação, é o responsável pela interface homem-máquina do sistema, permitindo ao usuário interagir com o sistema;
- Builder: localizado na Estação, é o responsável pela construção da representação gráfica do sistema.

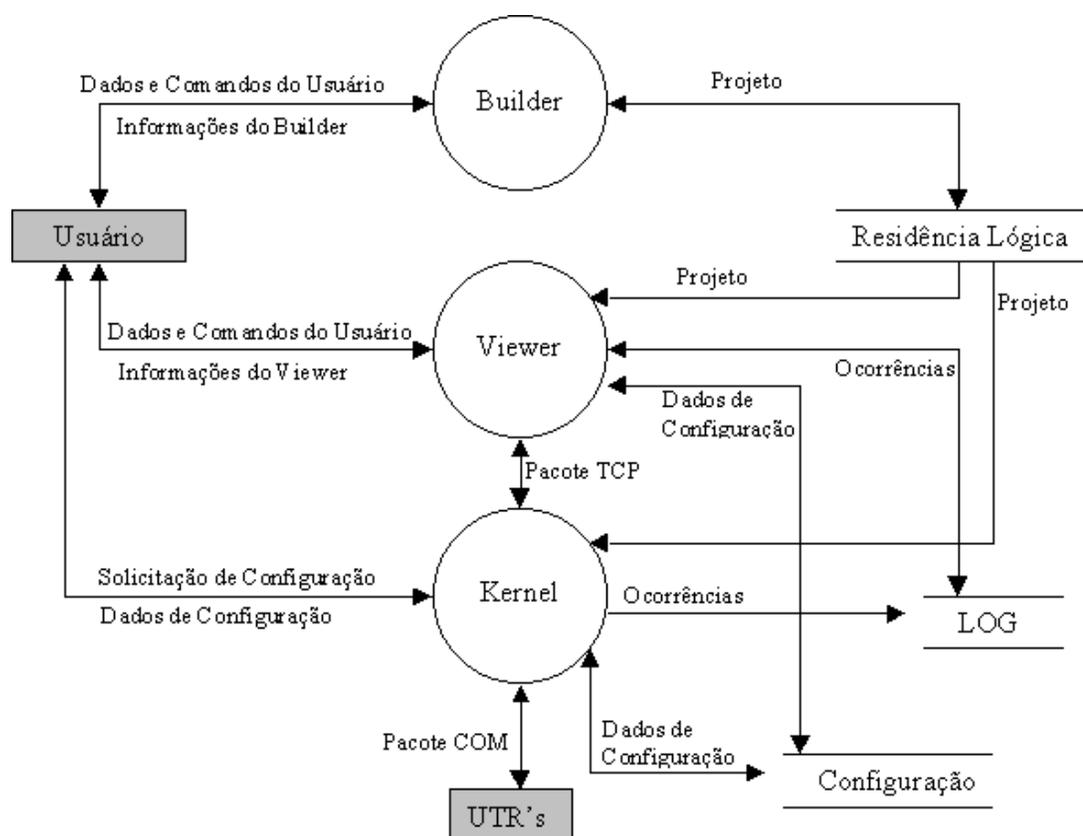


Figura 2.3 : Diagrama da Arquitetura Lógica

2.3. Dispositivos suportados pelo sistema

Existem atualmente disponíveis no mercado uma série de equipamentos que podem ser conectados num sistema de automação residencial. Nesta primeira fase de desenvolvimento, optou-se em escolher um subconjunto de equipamentos que seriam os necessários para automatizar uma residência de forma conveniente.

Para que o sistema atue sob os equipamentos, é necessário que o sistema conheça suas características. Cada equipamento suportado pelo sistema foi encapsulado dentro de um componente, herdando suas características básicas de um outro componente denominado TBaseDevice, que contém as propriedades, métodos e eventos básicos para todos os equipamentos.

Cada equipamento pode encontrar-se em quatro estados, sendo eles :

- **Ativado** : cada grupo de equipamentos possui um significado diferente para este estado. Para os equipamentos que só recebem sinais do ambiente, no caso de sensores, este estado ocorre quando um sinal é recebido. No caso dos equipamentos que realizam alguma tarefa no ambiente, como portas, luzes, torneiras, este estado indica que estes dispositivos estão abertos ou acesos;
- **Conectado** : este estado indica que o equipamento está conectado ao sistema, de forma que o mesmo possa monitorá-lo;
- **Alarmado** : este estado indica que se o equipamento entrar no estado ativado, deve-se gerar um sinal de alarme. Por exemplo, os sensores de presença, depois que todos os habitantes da casa já foram dormir, entram no estado alarmado. Se um deles captar algum movimento, os mesmos devem emitir um alarme, já que a residência está sendo violada. Durante o dia, o alarme não pode disparar, já que quem está violando a residência são os próprios habitantes.

Cada equipamento possui uma representação gráfica única, que os identifica para o usuário e possuem uma variação de três cores que indicam três possíveis estados que o equipamento pode se encontrar : desconectado, conectado mas não ativado e conectado e ativado. O usuário é que escolhe as cores dos equipamentos e de seus respectivos estados.

O componente TBaseDevice possui as seguintes propriedades básicas comuns a todos os equipamentos:

- Connected : indica se o equipamento está ou não conectado ao sistema;
- Activated : indica se o equipamento está ou não ativado;
- Alarmed : indica se o equipamento está ou não alarmado;
- Address : indica qual a UTR que está controlando o equipamento;
- Port : indica a porta da UTR onde o equipamento está conectado;
- ColorConnected : indica a cor que o dispositivo deve assumir enquanto estiver conectado;
- ColorEdge : indica a cor do contorno dos equipamentos;
- ColorActivated : indica a cor que o equipamento deve assumir enquanto estiver ativado;
- ColorDisconnected : propriedade que indica a cor que o dispositivo deve assumir enquanto estiver desconectado;
- Regras : contêm as regras que os equipamentos devem executar quando alteram seus estados ou em períodos regulares de tempo.

Os equipamentos suportados pelo sistema, com o seu respectivo componente e suas funcionalidades serão descritos a seguir.

2.3.1. Sensor de presença

Encapsulado no componente TActivity, possui a funcionalidade de indicar a presença de alguma pessoa no ambiente.

Conectado



Ativado



Desconectado



Figura 2.4 : Sensor de presença

2.3.2. Sensor de incêndio

Encapsulado no componente TBurning, possui a funcionalidade de indicar a presença de fumaça ou calor excessivo no ambiente, indícios de incêndio.



Figura 2.5 : Sensor de incêndio

2.3.3. Máquina fotográfica digital

Encapsulada no componente TCamera, possui a funcionalidade de fotografar o ambiente, com a possibilidade de movimentar-se nos eixos horizontal e vertical.



Figura 2.6 : Máquina fotográfica digital

2.3.4. Cortina

Encapsulado no componente TCurtain, possui a funcionalidade de indicar se a cortina está aberta ou não, além de manipula-la automaticamente.



Figura 2.7 : Cortina

2.3.5. Porta

Encapsulado no componente TDoor, possui a funcionalidade de indicar se a porta está aberta ou não, além de manipulá-la automaticamente.



Figura 2.8 : Porta

2.3.6. Lâmpada

Encapsulado no componente TLamp, possui a funcionalidade de indicar se a lâmpada está acessa ou não, além de manipulá-la automaticamente, levando em consideração o nível de intensidade desejado.



Figura 2.9 : Lâmpada

2.3.7. Sensor de ruído

Encapsulado no componente TNoise, possui a funcionalidade de indicar a presença de ruídos no ambiente, identificando a quebra de uma vidraça por exemplo.



Figura 2.10 : Sensor de ruído

2.3.8. Tomada de força

Encapsulada no componente TPower, possui a funcionalidade de indicar a presença de energia elétrica na tomada, além de manipulá-la automaticamente.

Conectado



Ativado



Desconectado



Figura 2.11 : Tomada de força

2.3.9. Sensor infravermelho

Encapsulado no componente TRay, possui a funcionalidade de indicar a presença de pessoas através da interrupção do seu raio infravermelho.

Conectado



Ativado



Desconectado



Figura 2.12 : Sensor infravermelho

2.3.10. Interruptor

Encapsulado no componente TSwitch, possui a funcionalidade de indicar a posição do interruptor, além de manipular o equipamento com o qual está relacionado.

Conectado



Ativado



Desconectado



Figura 2.13 : Interruptor

2.3.11. Torneira

Encapsulado no componente TTap, possui a funcionalidade de indicar se a torneira está aberta ou fechada, além de manipulá-la automaticamente.

Conectado



Ativado



Desconectado



Figura 2.14 : Torneira

2.3.12. Janela

Encapsulado no componente TWindow, possui a funcionalidade de indicar a posição da janela, além de manipulá-la automaticamente.

Conectado



Ativado



Desconectado



Figura 2.15 : Janela

Diversos outros equipamentos podem ser agregados ao sistema, bastando que sejam encapsulados num componente que herde suas características do componente TBaseDevice. Atualmente nenhum dispositivo consegue realizar uma tarefa sobre o ambiente, como por exemplo, abrir a porta, mas todos conseguem demonstrar seus estados atuais pelo sistema.

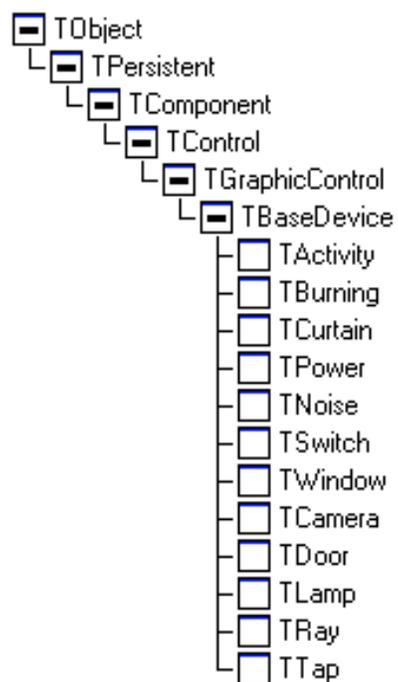


Figura 2.16 : Árvore de componentes que representam os equipamentos.

3. Kernel

É o aplicativo responsável pelo controle centralizado de todo o sistema, sendo responsável pela escolha do que fazer em certas situações, além de possuir o conhecimento dos estados de todos os equipamentos que perfazem o ambiente.

Para facilitar a apresentação do Kernel, o mesmo será dividido em quatro grupos, cada um responsável por um conjunto de componentes e objetos que realizam uma determinada atividade em particular.

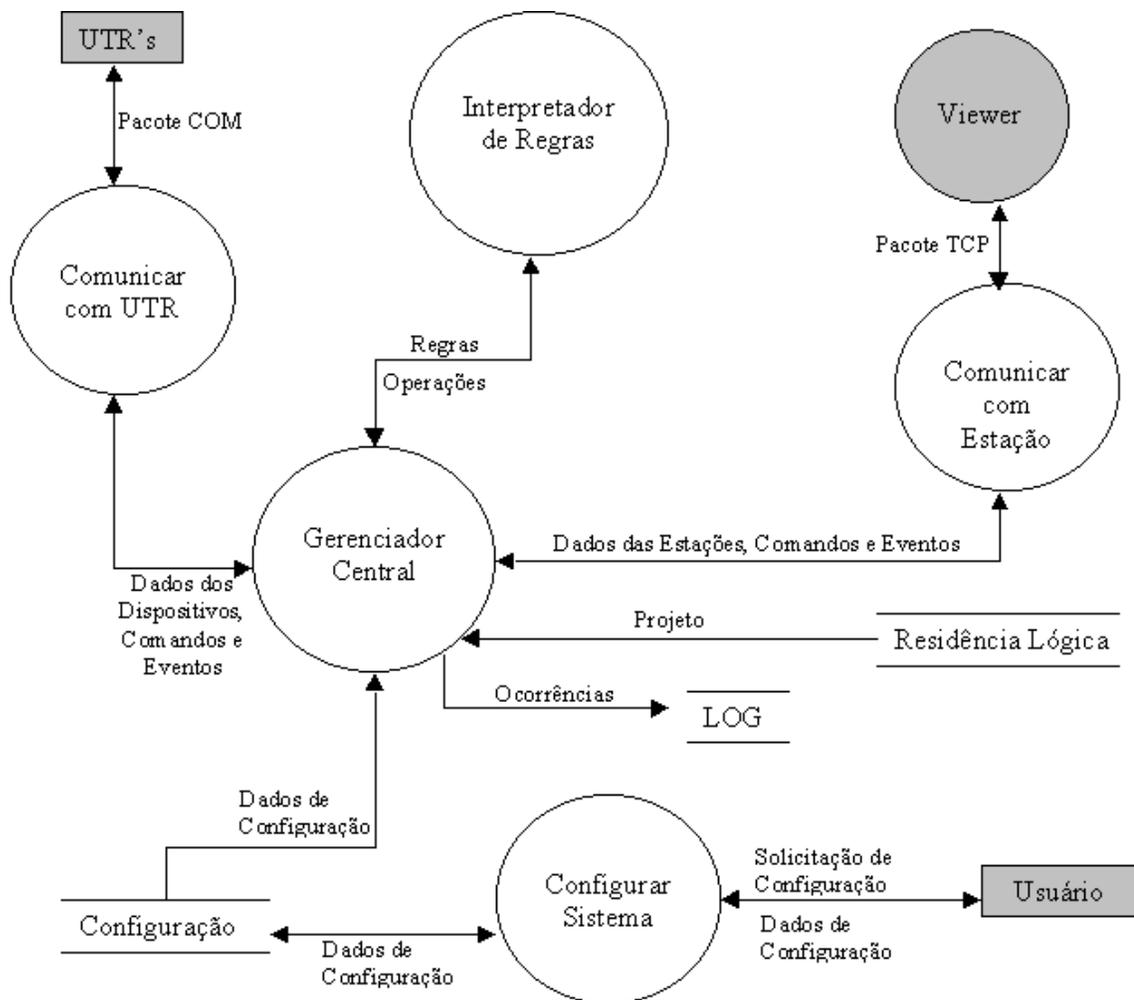


Figura 3.1 : Diagrama lógico do Kernel

3.1. Comunicação com as UTR's

Toda a comunicação realizada entre as UTR's e o Master, onde está localizado o Kernel, é realizado através de uma porta serial. Um protocolo de rede foi projetado para gerenciar esta comunicação, visando aumentar a confiabilidade dos dados transmitidos. Os dados são transmitidos dentro de pacotes, e cada pacote transmitido precisa receber do destino um pacote de confirmação, caso contrário o emissor retransmitirá o pacote.

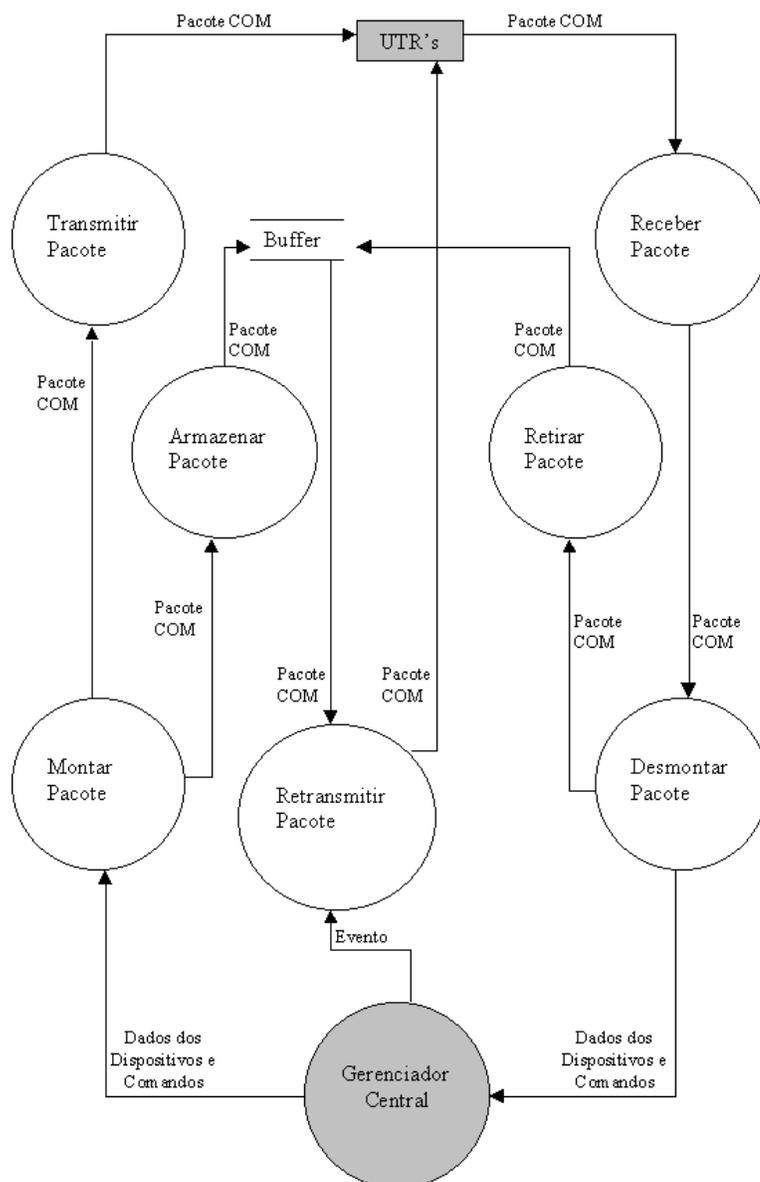


Figura 3.2 : Diagrama lógico da Comunicação com as UTR's

3.1.1. Estrutura dos pacotes

Foram especificados três tipos de pacotes, uma para cada tipo de informação transmitida:

- Pacote de dados : para informações sobre os estados dos equipamentos;
- Pacote de controle : para informações de controle entre o Kernel e o Viewer, que será descrito no próximo capítulo, e entre o Kernel e as UTR's;
- Pacote de confirmação : enviado pelo receptor ao emissor avisando que o pacote enviado chegou ao seu destino corretamente.

O pacote de dados possui uma estrutura bastante simples, facilitando a implementação do protocolo. A estrutura do pacote é composta por seis campos de um byte cada, totalizando em um tamanho fixo de seis bytes. O pacote de dados é formado pelos seguintes campos :

- Delimitador : indica o começo do pacote, e contem como valor fixo o caractere '[';
- Código : indica qual o tipo do pacote que está sendo transmitido, possuindo para o pacote de dados o valor fixo do caractere '#';
- Número : Contem um número único que identifica o pacote, utilizado no pacote de confirmação, aceitando valores entre 1 e 255;
- Porta : Indica a porta da UTR onde o equipamento está localizado. Aceita valores de 1 até 255, ou seja, nesta fase, cada UTR só consegue controlar no máximo 254 equipamentos;
- Valor : Indica o valor que o equipamento contém no caso da transmissão da UTR para o Master, ou o valor que deseja-se que o equipamento possua, no caso da transmissão do Master para a UTR;
- Delimitador : indica o fim do pacote, e contem como valor fixo o caractere ']';

Delimitador	Código	Número	Porta	Valor	Delimitador
[#	1 .. 255	1 .. 255	1 .. 255]

Figura 3.3 : Pacote de Dados da COM

No pacote de controle são enviadas informações e solicitações de controle do sistema, como por exemplo, o pedido do Master para as UTR's solicitando que estes lhes

envie o estado de todos os equipamentos. Possui uma estrutura de cinco campos de um byte, totalizando em um tamanho fixo de cinco bytes. O pacote de controle é formado pelos seguintes campos :

- Delimitador : o mesmo do pacote de dados;
- Código : possui o valor '\$' para identificar o pacote de controle;
- Número : o mesmo significado do pacote de dados;
- Operação : indica o que deseja-se que o receptor faça;
- Delimitador : o mesmo do pacote de dados.

Delimitador	Código	Número	Operação	Delimitador
[\$	1 .. 255	1 .. 255]

Figura 3.4 : Pacote de controle da COM

O pacote de confirmação retorna ao emissor a informação de que o pacote, indicado pelo campo número, chegou ao seu destino. Possui uma estrutura de quatro campos de um byte, totalizando em um tamanho fixo de quatro bytes. O pacote de confirmação é formado pelos seguintes campos :

- Delimitador : o mesmo do pacote de dados;
- Código : possui o valor '%' para identificar o pacote de confirmação;
- Número : indica qual o pacote está sendo confirmado;
- Delimitador : o mesmo do pacote de dados.

Delimitador	Código	Número	Delimitador
[%	1 .. 255]

Figura 3.5 : Pacote de confirmação da COM

3.1.2. Protocolo

O protocolo de rede utilizado na comunicação pela porta serial é bastante simples. Todos os dados transmitidos são encapsulados dentro de pacotes, e cada pacote enviado pelo emissor deve receber uma confirmação do receptor.

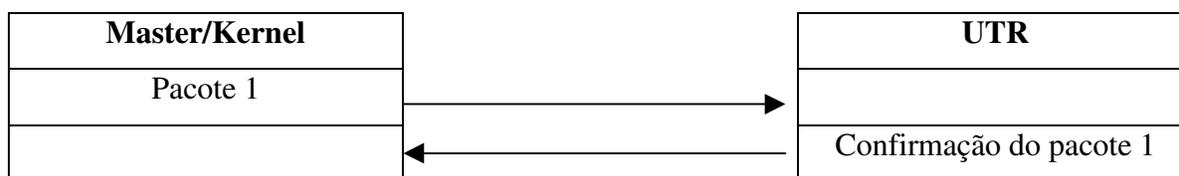


Figura 3.6 : Protocolo de comunicação entre o Master e a UTR

Os pacotes enviados possuem um tempo t para receberem a confirmação, caso contrário serão retransmitidos até a chegada do pacote de confirmação ou até o limite máximo de retransmissões. É responsabilidade do usuário configurar o tempo de espera da confirmação e o número máximo de retransmissões.

Um buffer armazena e gerencia os pacotes enviados, a espera de confirmação, de forma a liberar o protocolo para enviar outros pacotes, não o deixando-o preso a espera da confirmação. Quando um pacote de confirmação é recebido, ele é repassado ao buffer que se responsabiliza de excluir o pacote.

Cada pacote enviado deve possuir um número único de identificação, que é utilizado na confirmação para indicar qual o pacote que está sendo confirmado. O protocolo de rede está encapsulado no componente TDeviceCOM, que isola o restante da aplicação do conhecimento explícito do seu funcionamento.

A interface do componente com a aplicação é realizada através de um método que envia os dados pela serial, responsabilizando-se pela implementação do protocolo, e os dados recebidos são enviados a aplicação através de um evento.

3.1.3. Comunicação serial

A comunicação serial está encapsulada em um componente, chamado de TCOM32, distribuído na Internet, desenvolvido sobre a API do Windows de 32 bits, utilizando duas Thread que realizam a comunicação, uma para o recebimento dos dados, e a outra para o envio dos dados.

O componente disponibiliza um método para enviar uma quantidade de bytes pela serial, e os bytes que chegam pela porta são repassados através de um evento.

3.1.4. *Buffer de pacotes*

Para organizar e armazenar os pacotes enviados pela serial a espera de confirmação, desenvolveu-se um componente, chamado TBufferCOM, responsável por esta função.

Os pacotes são armazenados numa fila de registros, onde cada registro contém o número do pacote, o pacote em si, a quantidade de vezes que já foi retransmitido e um apontador para o próximo registro da fila. Quando um novo pacote é enviado, ele é inserido na lista e só sairá quando sua confirmação for recebida ou quando esgotar-se o número de retransmissões, ocorrência que ocasionará um erro a ser relatado. Quando o pacote é retransmitido, o seu contador é incrementado e ele não é inserido de novo na lista.

3.2. **Comunicação com a Estação**

A comunicação realizada entre o Kernel e o Viewer, localizado na Estação, é feita sobre o protocolo de rede TCP/IP, conseqüentemente, independente do meio físico utilizado para a transmissão dos dados.

3.2.1. *Estrutura dos pacotes*

É formado pelos mesmos tipos de pacotes utilizados na comunicação serial, diferindo apenas na estrutura dos mesmos.

Para que os pacotes fossem transmitidos pelo componente responsável pela comunicação TCP/IP, o mesmo deve ser transmitido como uma seqüência de caracteres. Por essa razão, os valores inteiros são transformados numa cadeia de caracteres para serem inseridos no pacote. Assim, os pacotes possuem um tamanho variado, dependendo do número a ser transmitido. Por exemplo, para transmitir o número 222 necessita-se de três caracteres, ao invés de um byte.

Um separador foi acrescentado aos pacotes de forma a delimitar o tamanho dos campos que transportam dados numéricos. O delimitador escolhido foi o caractere '!':

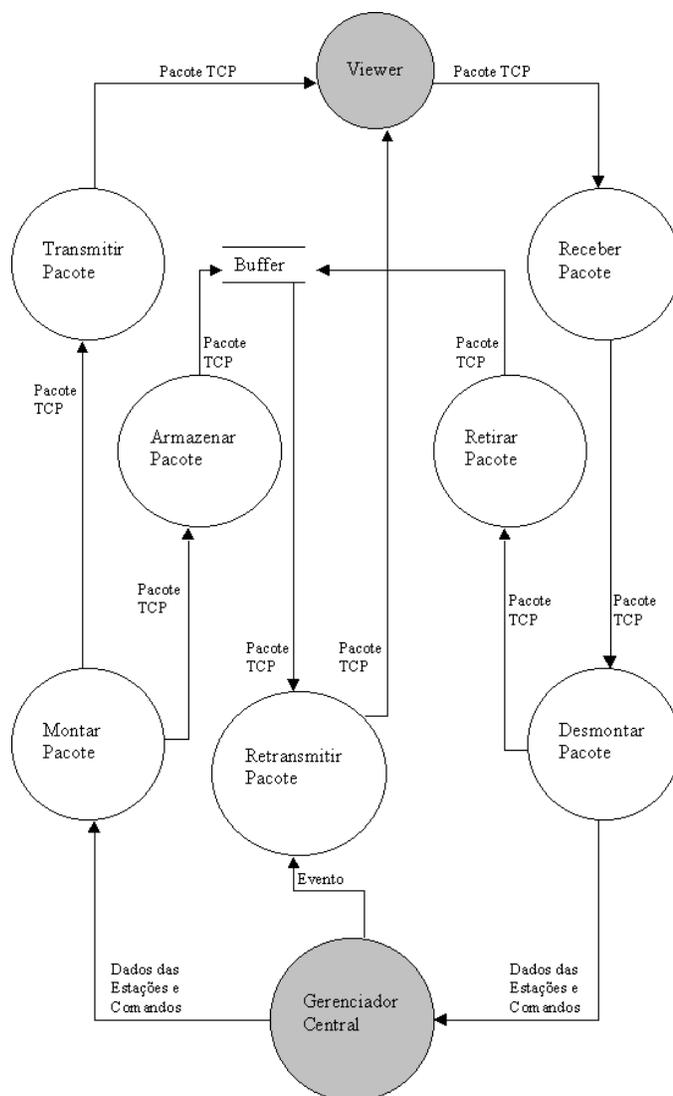


Figura 3.7 : Diagrama lógico da Comunicação com as Estações

O pacote de dados possui alguns campos a mais em relação ao pacote de dados da comunicação serial, como o campo Endereço que indica qual a UTR que controla o equipamento, e os separadores entre os campos número, endereço, porta e valor. O pacote de controle ganhou um separador entre o campo número e operação, e o pacote de confirmação não foi alterado.

Delimitador	Código	Número	Separador	Endereço	Separador	Porta	Separador	Valor	Delimitador
[#	1..32000	:	1 .. 255	:	1 .. 255	:	1 .. 255]

Figura 3.8 : Pacote de Dados do TCP

Delimitador	Código	Número	Separador	Operação	Delimitador
[\$	1 .. 32000	:	1 .. 255]

Figura 3.9 : Pacote de Controle do TCP

Delimitador	Código	Número	Delimitador
[%	1 .. 32000]

Figura 3.10 : Pacote de Confirmação do TCP

3.2.2. Protocolo

O protocolo de aplicação utilizado sobre o protocolo de rede TCP/IP é semelhante ao protocolo de rede utilizado na comunicação serial. Ele utiliza os mesmos valores do tempo de espera para retransmitir o pacote e o número máximo de retransmissões da comunicação serial.

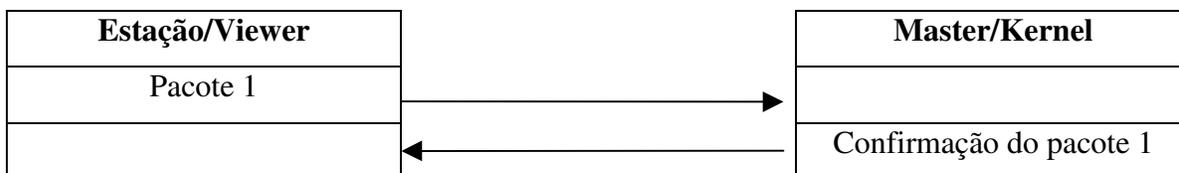


Figura 3.11 : Protocolo de comunicação entre o Viewer e o Kernel

3.2.3. Comunicação TCP/IP

O protocolo de rede TCP/IP está encapsulado no componente TTCP fornecido junto com a linguagem de programação, disponibilizando alguns métodos e eventos, sendo que um dos métodos é o responsável pelo envio dos dados pela conexão TCP, e um dos eventos é o responsável pelo recebimento dos dados que chegam pela conexão.

O usuário pode selecionar a porta desejada para a conexão TCP utilizada na comunicação, possibilitando uma flexibilidade quanto à escolha da porta. Para que a

comunicação seja realizada, é necessário que o usuário possua, no Master e nas Estações, o protocolo TCP/IP instalado, especificando um endereço IP para cada uma das máquinas.

3.2.4. Buffer de pacotes

A estrutura e função do buffer de pacotes da comunicação TCP/IP é a mesma do buffer da comunicação serial, alterando-se apenas a estrutura do pacote armazenado, que na comunicação TCP/IP é uma seqüência de caracteres não fixa, e na comunicação serial é uma seqüência de bytes com tamanho máximo de 6 bytes.

3.3. Interpretador de regras

Cada equipamento assistido pelo sistema possui uma base de regras com os procedimentos que o sistema deve executar quando as premissas das regras são satisfeitas. As regras são analisadas sempre que o equipamento trocar de estado ou em intervalos regulares de tempo de um minuto.

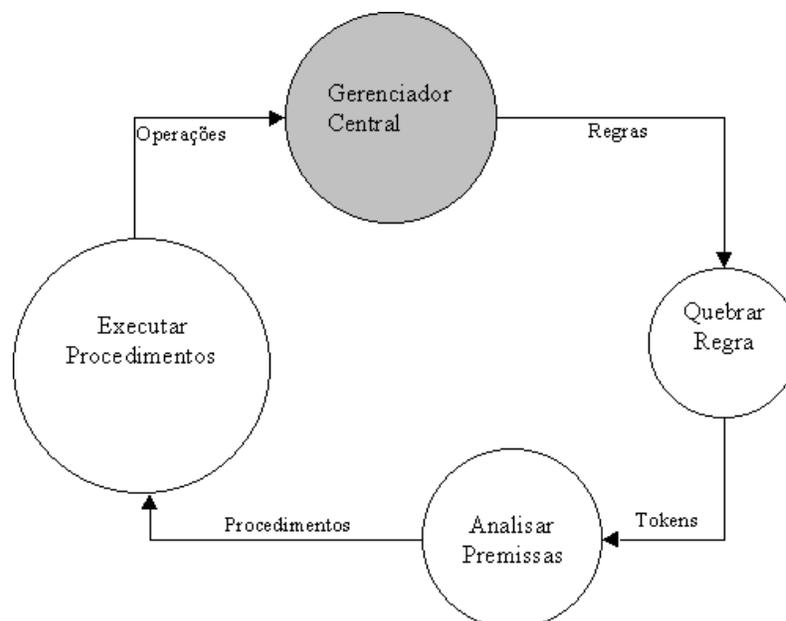


Figura 3.12 : Diagrama lógica do Interpretador de Regras

As regras servem para o usuário programar as ações que o sistema terá que executar quando determinados fatos ocorrerem. Por exemplo, pode-se programar uma lâmpada para acender às vinte horas e desligar-se às sete horas. Programar a cortina para fechar junto com a janela.

Para que o usuário possa programar as ações, uma pseudo-linguagem de programação foi implementada. Ela é composta por três palavras-chaves, três operadores lógicos e cinco operações. A linguagem de programação e o seu interpretador estão encapsulados do componente TRegra.

As palavras-chaves que compõem a linguagem são:

- Ifm : Indica o começo de uma regra, que deverá ser analisada na mudança de estado do equipamento;
- Ift : Indica o começo de uma regra, que deverá ser analisada em intervalos regulares de tempo;
- Then : Indica o começo do bloco de procedimentos.

Os operadores lógicos suportados pela linguagem são:

- ~ : equívale a operação de negação, invertendo o valor de uma operação;
- ^ : equívale a operação de multiplicação lógica;
- + : equívale a operação de soma lógica.

As operações suportadas pela linguagem são :

- ac[A, P] : retorna ou seta o valor da propriedade Actived do equipamento indicado pelo endereço (Address) e porta (Port). Se a indicação do equipamento não estiver presente, assumi-se que é o equipamento que contem a regra;
- co[A, P] : retorna ou seta o valor da propriedade Connected;
- al[A, P] : retorna ou seta o valor da propriedade Alarmed;
- t[hh, mm] : retorna verdadeiro caso o horário atual seja o especificado pela hora (hh) e minuto (mm) ;
- dt[dd, msms, hh, mm] : retorna verdadeiro caso a data atual seja a especificada pelo dia (dd) e mês (msms) , e o horário atual seja o especificado pela hora (hh) e minuto (mm);
- op[c] : executa a operação especificado pelo código (c).

A sintaxe da linguagem para a programação das regras é descrita a seguir:

- um regra deve inicia-se por *ifm*, caso deva ser analisada nas mudanças de estados, ou *ift*, caso deva ser analisada em períodos regulares de tempo, seguido de um espaço em branco;
- As premissas são compostas pelas operações da linguagem, com exceção da operação *op*, unidas pelos operadores lógicos;
- após as premissas vem a palavra-chave *Then*, indicando o início do bloco de procedimentos;
- o bloco de procedimento é formado pelas operações da linguagem, com exceção das operações *t* e *dt*, separadas por um vírgula.

Ifm ou Ift	Premissa	Then	Procedimentos
------------	----------	------	---------------

Figura 3.13 : Estrutura de uma regra

Exemplos de regras válidas:

- *Ifm ac ^ al[4,2] then al[4,4]* : Se o dispositivo atual mudou seu estado para conectado e o dispositivo localizado na UTR 4, na porta 2, estiver no estado alarmado, então o dispositivo localizado na UTR 4, na porta 4, deve entrar no estado alarmado;
- *Ift t[08, 00] + ~al then ~co* : Se for oito horas ou o dispositivo desta regra não estiver no estado alarmado, então o dispositivo deve ser desconectado do sistema;
- *Ifm ~co then op[1]* : Se o dispositivo mudar seu estado para desconectado, então deve-se realizar a operação identificada pelo número.

3.4. Gerenciador central

O núcleo do Kernel está encapsulado no objeto TFMMain, responsável pelo fluxo de dados e de controle entre seus diversos componentes.

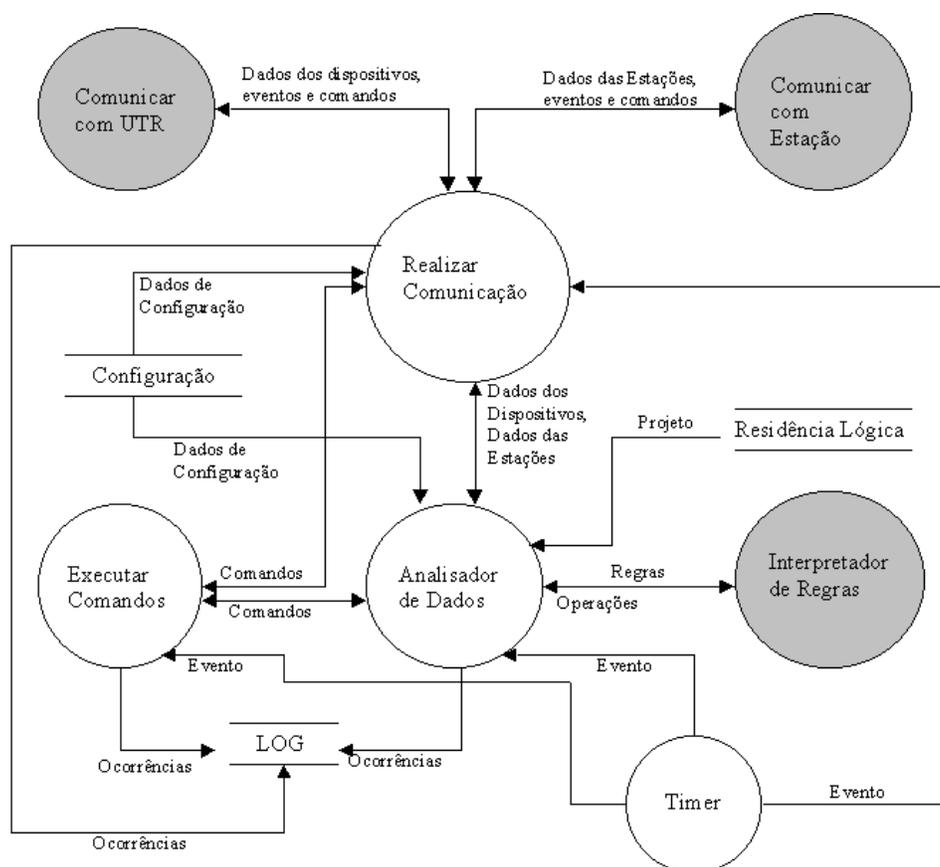


Figura 3.14: Diagrama lógico do Gerenciador Central do Kernel

3.4.1. Arquivos de Log

Tudo o que acontece no sistema é armazenado em arquivos de log para posterior análise. Os arquivos de log são criados diariamente, mantendo-se os registros de um dia em cada arquivo. Eles são armazenados em tabelas Paradox, facilitando a visualização e a procura dos registros. O objeto responsável pelo gerenciamento dos arquivos de log chama-se T Tabelas. São quatro os arquivos de log criados pelo Kernel :

- Log do sistema : armazena os erros que ocorreram na execução do aplicativo e os principais acontecimentos que ocorreram sobre o mesmo;
- Log dos comandos executados : armazena todos os comandos executados pelo Kernel;
- Log do alarmes : armazena todos os alarmes ocorridos;
- Log de operação : armazena todas as operações sobre os dispositivos.

3.4.2. Processo de comunicação

O Kernel suporta várias conexões seriais e TCP/IP, mantendo para isso uma lista com todas as conexões ativas no momento. As conexões com as COM's não podem ser realizadas a qualquer hora, elas são criadas na inicialização do Kernel. Para cada porta serial é criada uma instância do componente responsável pelo protocolo da porta serial.

O Kernel possui uma porta TCP que fica esperando as requisições de conexões. Quando recebe uma requisição cria um instância do componente responsável pelo protocolo da porta TCP para esta requisição, e a acrescenta na lista de conexões. As mudanças de estados dos equipamentos são transmitidos para todas as conexões TCP's presentes no momento, afim de mante-las atualizadas com o estado real do sistema.

Um temporizador fica responsável por avisar os buffers de pacotes a espera de confirmação das várias instâncias dos protocolos de comunicação à retransmitirem os pacotes depois de um tempo especificado. A existência de um único temporizador para todas as instancias reduz a carga de processamento do sistema, pois um temporizador é um processo que está sempre consumindo processamento.

Quando o Kernel recebe um pacote de um dos lados, imediatamente envia a confirmação, caso este pacote esteja correto. No exemplo abaixo, o pacote 1 contém um pedido para acender uma lâmpada, e o pacote 2 contém a indicação de que a lâmpada foi acesa.

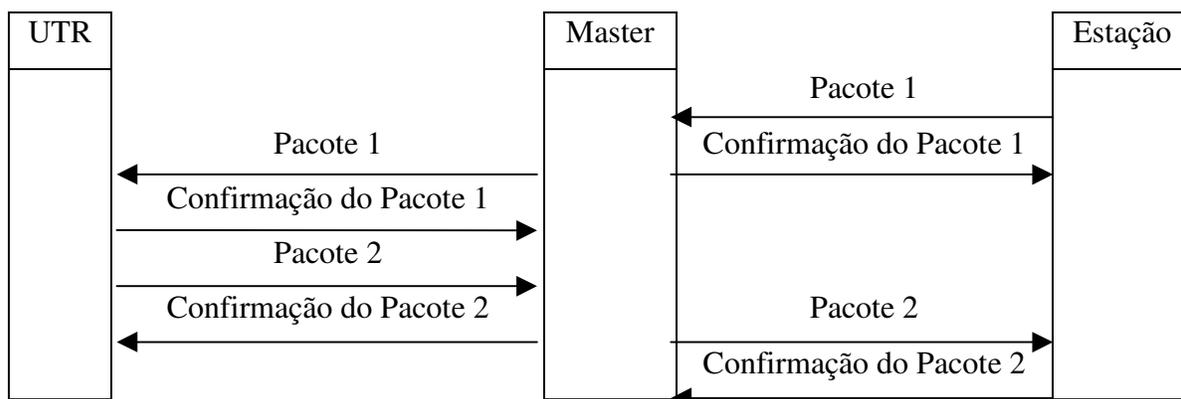


Figura 3.15 : Exemplo das trocas de pacotes entre as máquinas

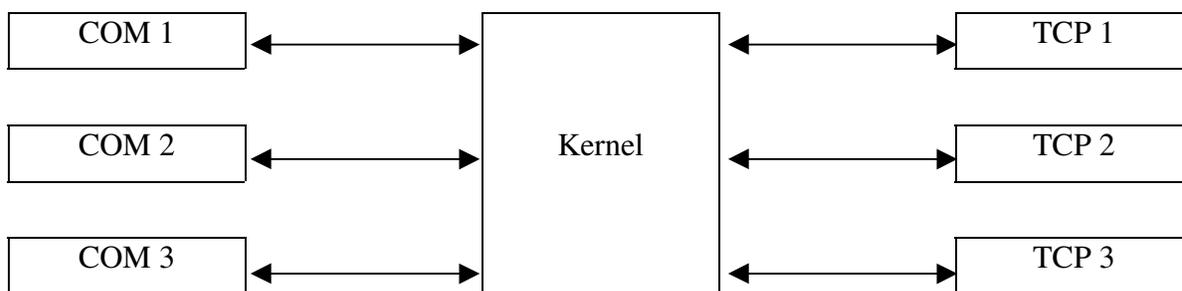


Figura 3.16 : Conexões do Kernel

3.4.3. Configurações

O usuário é o responsável pela configuração do sistema, sendo para isso disponibilizado uma tela onde são ajustados os parâmetros necessários ao Kernel. Estes parâmetros são armazenados num arquivo do tipo INI, chamado de Guardian.ini, localizado na pasta do sistema operacional (C:\Windows). Os parâmetros que o usuário pode configurar são :

- Projeto : Qual o projeto da residência que o Kernel controlará. No próximo capítulo será descrito com mais detalhes o conceito de projeto;
- Porta : Qual a porta TCP utilizada para que as Estações se conectem ao sistema;
- Intervalo : Qual o intervalo de tempo que deve-se esperar antes de retransmitir um pacote sem confirmação;
- Retransmitir : Indica o número de vezes que deve-se retransmitir um pacote antes de relatar uma mensagem de erro;
- Logs : Onde os arquivos de log devem ser armazenados;
- COM : Quais são as portas seriais disponíveis para o sistema.

3.4.4. Funcionamento

O Kernel envia as mudanças dos equipamentos vindas das UTR's para as Estações, e passa as solicitações de mudanças de estados das Estações para as UTR's. Antes de enviar as mudanças para as Estações, o Kernel submete estas mudanças ao interpretador de regras, de

modo a verificar a ocorrência de uma operação associada aquela mudança. Caso exista, os procedimentos serão encaminhados e a mudança enviada as Estações.

A cada minuto o Kernel chama o interpretador de regras para analisar e executar as regras temporais. Quando um procedimento altera o estado de um equipamento, este pedido é encaminhado a UTR responsável pelo equipamento.

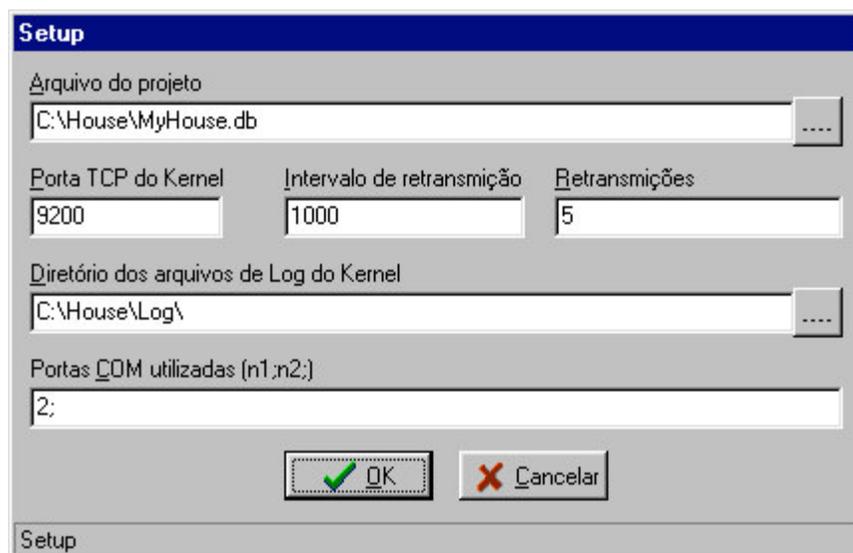


Figura 3.17 : Tela de Setup do Kernel

4. Builder & Viewer

Como já mencionado anteriormente, as estações são responsáveis pela interface do sistema com o usuário. Estes dois aplicativos estão localizados nas Estações, e fornecem a interface necessária para que o usuário possa visualizar o sistema e construí-lo.

O aplicativo Builder é o responsável pela construção lógica da residência, proporcionando ao usuário uma flexibilidade quanto a representação gráfica desejada para representar a sua residência na Estação.

O aplicativo Viewer é o responsável pela visualização dos estados atuais do sistema, possibilitando que o usuário altere os estados dos equipamentos através do clique do botão do mouse, ou seja, através da Estação o usuário consiga acender uma lâmpada, por exemplo.

É utilizado o conceito de janela filha, de forma que várias telas podem ser expostas ao mesmo tempo, além da inclusão de uma série de funcionalidades para trabalhar com janelas filhas, como minimizar todas, ordenar e assim por diante.

4.1. Builder

É o construtor do sistema no nível lógico, trabalhando off-line no sistema. Sua função é a de criar os ambientes sobre os quais o sistema assistirá, configurando cada equipamento de modo que o sistema conheça a sua localização e suas funcionalidades.

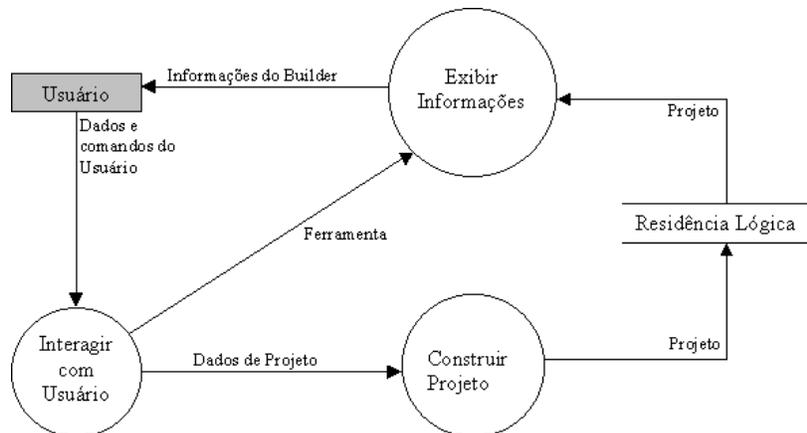


Figura 4.1 : Diagrama lógico do Builder

4.1.1. Projeto

As telas dos aplicativos são padronizadas de forma a facilitar o aprendizado para a utilização do sistema. Todas as telas que o compõem são construídas utilizando-se o mouse e o teclado, arrastando-se os equipamentos para as posições desejadas, e atualizando suas propriedades através de um editor de propriedades.

Quando o usuário constrói a sua residência lógica, ele o faz ambiente por ambiente, utilizando-se de várias telas para representar a residência, uma tela para cada ambiente. Para o sistema saber quais são as telas que compõem a residência, é necessário criar um projeto, que é uma tabela Paradox, onde estão armazenados todos os arquivos de tela que compõem o sistema.

O usuário possui a liberdade para acrescentar novas telas e retirar as antigas sempre que o desejar, mas deverá reiniciar o sistema, de forma que o mesmo reconheça as novas telas.

4.1.2. Paleta

Para que o usuário possa montar a sua residência, o Builder possui uma paleta com todos os componentes disponíveis no sistema. Esta paleta pode ser colocada em qualquer posição da área de trabalho.

O usuário seleciona o componente desejado, selecionando-o com o mouse, e então escolhe o lugar na tela onde o componente ficará. O objeto responsável pela paleta é o TFObjetos.

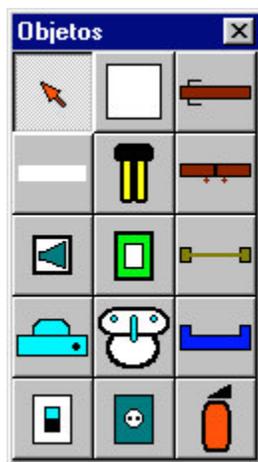


Figura 4.2 : Paleta de componentes

4.1.3. Editor de propriedades

As propriedades dos componentes são alteradas através de um editor de propriedades que disponibiliza ao usuário acesso a todas as propriedades que o objeto possui.

Nele o usuário indica a cor desejada do componente, em qual UTR o equipamento está localizado, seus estados de inicialização, entre tantas outras propriedades disponíveis. O objeto responsável pelo editor de propriedades é o TCompEditForm.

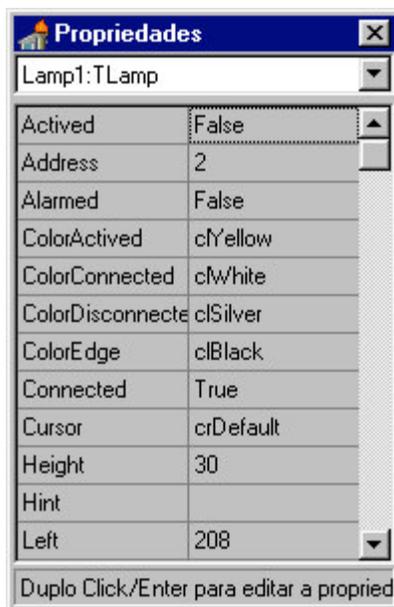


Figura 4.3 : Editor de Propriedades

As regras são programadas num módulo a parte, ativado pelo Editor de Propriedades.

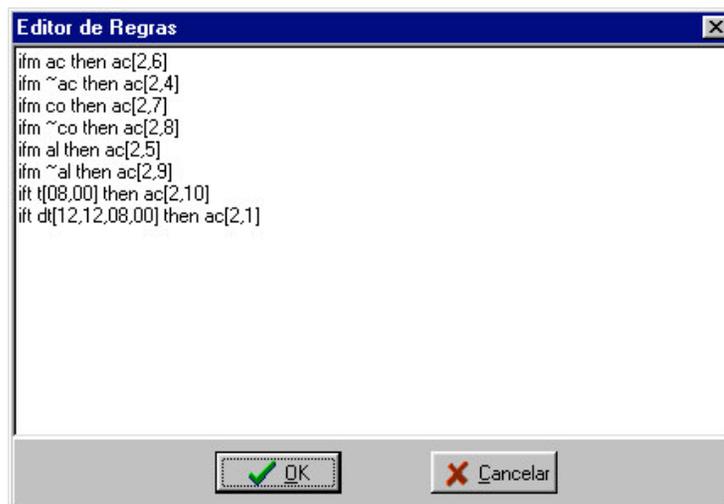


Figura 4.4 : Editor de regras

4.1.4. Formação da Tela

Uma tela é composta por uma série de componentes, cada um representando um equipamento específico suportado pelo sistema. Elas são armazenadas com a extensão .gc, através do objeto TFileStream, que possui como função armazenar e recuperar um fluxo de objetos do disco rígido.

Com a utilização deste objeto, evitou-se a criação de uma tabela relacional para armazenar as telas, já que não se dispõem de um banco de dados orientado a objetos.

Os objetos colocados na tela podem ser movidos, apagados, copiados, colados e recortados, proporcionado flexibilidade para o usuário construir suas telas. Para aumentar ou diminuir o tamanho, o componente selecionado recebe uma grade de destaque, e através dela o usuário pode aumentar ou diminuir o tamanho do componente sem recorrer ao editor de propriedades.



Figura 4.5 : Grade de destaque

Cada tela deve ser identificada por um número que indica a ordem de abertura das telas e a hierarquia das mesmas, possibilitando ao sistema a ordem de abertura das mesmas. As telas podem ser impressas ou capturadas para a área de transferência.

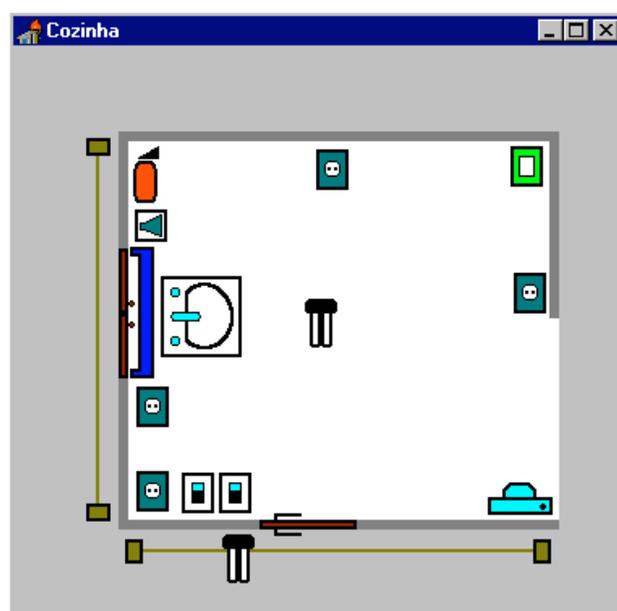


Figura 4.6 : Exemplo de tela

4.2. Viewer

Com a mesma interface da ferramenta Builder, possibilita ao usuário interagir com o sistema on-line, trocando informações com o Kernel. Nele não é permitido alterar as telas, apenas os estados dos objetos. Para isso, o editor de propriedades está presente, mas só é disponibilizado as propriedades possíveis de alteração, com exceção do nome do componente e das regras programadas, que estão presente para que o usuário visualize-as.

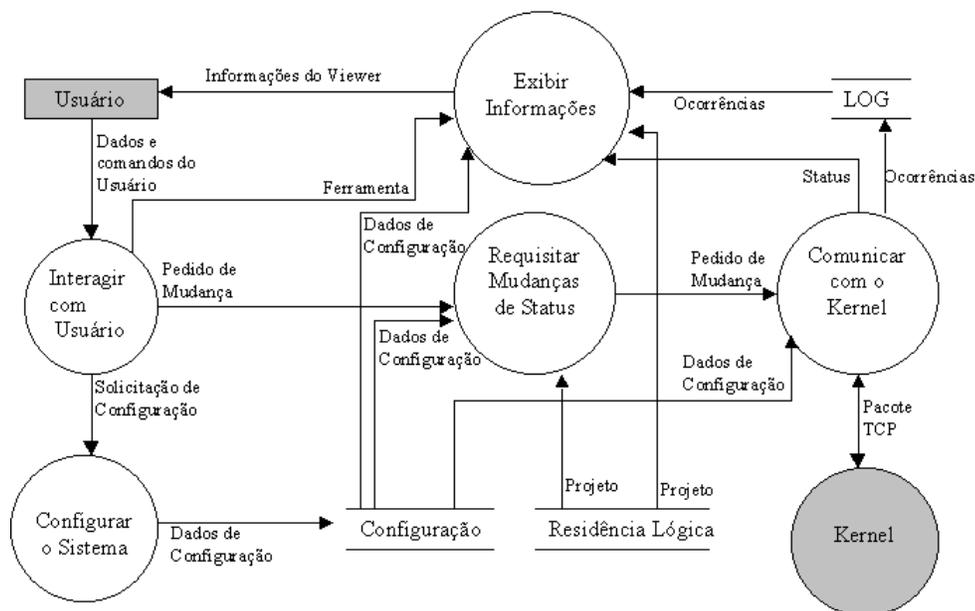


Figura 4.7 : Diagrama lógico do Viewer

O Viewer possibilita ver os arquivos de log gerados pelo Kernel, além do seu próprio arquivo de log, nos mesmos formatos dos arquivos do Kernel. Neste arquivo, estão armazenados os erros que ocorreram no Viewer e as principais operações realizadas pelo mesmos.

Para comunicar-se com o Kernel, ele possui uma instância do componente responsável pelo protocolo de aplicação sobre o TCP/IP. Para conectar-se, ele envia uma requisição de conexão para a porta TCP/IP responsável pelas requisições no Master, sendo atendida a requisição, a conexão é estabelecida e o Viewer fica on-line com o sistema.

É necessário armazenar algumas informações sobre o aplicativo para que o mesmo possa ser executado corretamente. Deste modo, existe uma tela de configuração, onde são configurados os parâmetros necessários a execução do mesmo.

Os parâmetros disponibilizados são:

- Projeto : Qual o projeto da residência controlado pelo sistema;
- Porta : Qual a porta TCP utilizada para conectar-se ao Kernel;
- Intervalo : Qual o intervalo de tempo que deve-se esperar antes de retransmitir um pacote sem confirmação;
- Retransmitir : Indica o número de vezes que deve-se retransmitir um pacote antes de relatar uma mensagem de erro;
- Endereço do Kernel : Indica o endereço IP ou o nome da máquina onde o Kernel está sendo executado;
- Log local : Indica a localização do arquivo de log do Viewer;
- Log do Kernel : Indica a localização dos arquivos de log mantidos pelo Kernel.

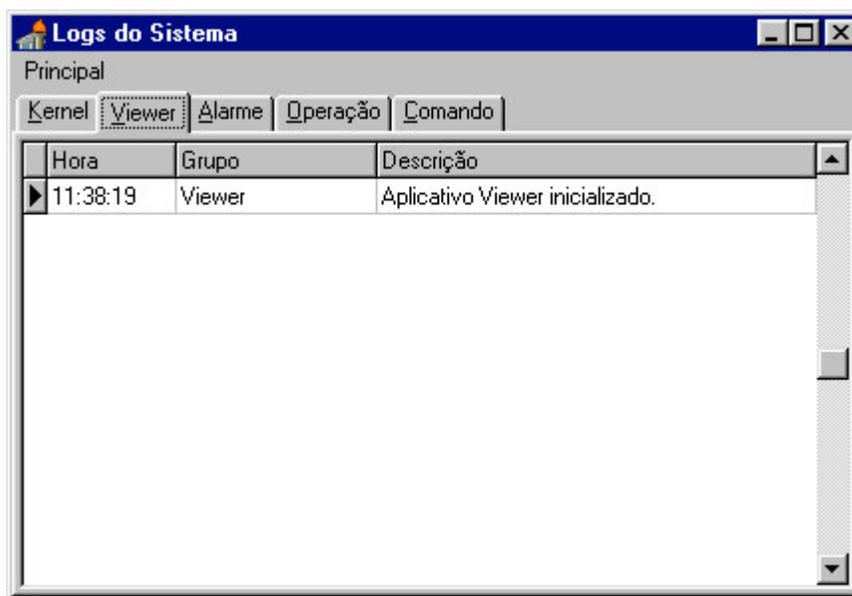


Figura 4.8 : Tela de Log do Sistema

Setup

Arquivo do projeto
C:\House\MyHouse.db

Porta TCP do Kernel Intervalo de retransmissão Retransmissões
9200 1000 5

Endereço TCP/IP ou o nome do servidor onde está localizado o Kernel
Server

Diretório dos arquivos de Log Local
C:\House\Log\

Diretório dos arquivos de Log do Kernel
C:\House\Log\

OK Cancelar

Setup do Viewer

Figura 4.9 : Tela de Setup do Viewer

5. Caso Base

Este capítulo apresentará um exemplo simples, ilustrando os passos necessários para compor um aplicativo. O primeiro passo é levantar todos os dispositivos que serão controlados pelo sistema, instalando-os e conectando-os às UTR's. Para este exemplo, utilizaremos apenas uma UTR, por haver um número reduzido de pontos monitorados.

Consideraremos um residência fictícia composta por três cômodos: cozinha, quarto e banheiro. Esta residência possuirá 41 dispositivos controlados pelo aplicativo, representando todos os equipamentos disponíveis no sistema.

O nome dos componentes com os seus respectivos valores para o endereço da UTR e a valor da porta são definidos a seguir.

Componente	Endereço	Porta
Porta 1	2	1
Porta 2	2	2
Janela 1	2	3
Janela 2	2	4
Janela 3	2	5
Janela 4	2	6
Lâmpada 1	2	7
Lâmpada 2	2	8
Lâmpada 3	2	9
Lâmpada 4	2	10
Infravermelho 1	2	11
Infravermelho 2	2	12
Infravermelho 3	2	13
Infravermelho 4	2	14
Cortina 1	2	15
Cortina 2	2	16
Cortina 3	2	17
Cortina 4	2	18
Interruptor 1	2	19

Interruptor 2	2	20
Interruptor 3	2	21
Interruptor 4	2	22
Presença 1	2	23
Presença 2	2	24
Presença 3	2	25
Presença 4	2	26
Torneira 1	2	27
Torneira 2	2	28
Incêndio 1	2	29
Incêndio 2	2	30
Incêndio 3	2	31
Tomada 1	2	32
Tomada 2	2	33
Tomada 3	2	34
Tomada 4	2	35
Tomada 5	2	36
Câmera 1	2	37
Câmera 2	2	38
Som 1	2	39
Som 2	2	40
Som 3	2	41

Com o levantamento de todos os pontos monitorados, com os seus respectivos endereços e portas, começa-se a representação lógica da residência, através da ferramenta Builder. Deve-se observar que os componentes são identificados pelas propriedades Endereço e Porta, razão pela qual não deve haver uma lâmpada com os mesmos valores que uma porta. Porém, uma mesma lâmpada representada em várias telas é permitido, já que representam a mesma lâmpada física.

5.1. Builder

Para armazenar a estrutura lógica da residência, é necessário criar um novo projeto, através do menu Arquivos, Novo Projeto. Neste exemplo, salvaremos o projeto como c:\House\MyHouse.db .

Com o projeto criado e aberto, iniciaremos a construção das telas. Construiremos quatro telas, uma para cada cômodo e mais uma representando a planta baixa da residência.

Para começar a construir as telas é necessário criar uma nova tela, através do menu Arquivos, Nova tela.

Através da tecla F12, chama-se a paleta de componentes, exibida na figura 4.2, com todos os componentes que o sistema gerencia e mais dois para a representação da residência, que são o componente THall, representando um cômodo, e o componente TPassage, representando uma passagem entre dois cômodos.

Os componentes são inseridos na tela da seguinte maneira : seleciona-se o componente desejado na paleta de componentes, e depois clique o botão esquerdo do mouse na posição onde deseja colocar o componente na tela. O componente que está atualmente ativo recebe uma grade de destaque, exibida na figura 4.5 .

Com a utilização da grade de destaque, o usuário pode mover o componente de um lado para o outro da tela e alterar o seu tamanho através do mouse ou utilizar o teclado, com as teclas Shift + Setas para controlar o tamanho do componente; e Ctrl + Setas para controlar a posição do mesmo. Pode-se também utilizar o editor de propriedades.

O editor de propriedades visualizado através do menu ou da tecla F11, exibido na figura 4.3, possibilita ao usuário configurar cada componente da melhor maneira possível, buscando-se deixar a residência lógica uma representação realística da residência física. Diversas propriedades estão disponíveis para o usuário, sendo as mais importantes encontradas nos componentes :

- Ordem : indica qual é a hierarquia da tela, informando ao sistema a ordem de exibição das telas. Cada tela do projeto deve possuir um valor de ordem única ;
- Address : contém o endereço da UTR onde está localizado o componente;
- Port : contém a porta da UTR onde está conectado o equipamento;
- Name : armazena o nome do componente;
- Plain : indica se o componente deve ser desenhado na vertical ou na horizontal;

- **Align** : indica se o componente deve ser desenhado em cima ou embaixo, caso a propriedade `plain` seja vertical; e se deve ser desenhado na direita ou na esquerda caso a propriedade `plain` seja horizontal;
- **ColorActivated** : qual a cor que o componente assumirá quando estiver no estado de ativado;
- **Regras** : armazena as regras do componente.

As regras são editadas num editor separado, chamado pelo editor de propriedades através da propriedade `Regras`. O editor de regras, exibido na figura 4.4, permite ao usuário programar as ações que cada componente tomará em cada ocasião. Cada linha do editor representa uma regra, que deve estar em conformidade com a estrutura apresentada anteriormente.

O usuário pode salvar as alterações feitas para o disco rígido através do menu `Arquivos, Salvar Tela`. Através dos comandos localizados no menu `Editar`, pode-se copiar, colar, recortar e apagar um componente da tela, ou mesmo copiar a tela para a área de transferência ou imprimi-la, onde poderá dar um zoom na imagem para visualização.

Telas feitas para outros projetos, ou feitas isoladamente podem ser agregadas ao sistema, através do menu `Arquivos, Adicionar ao Projeto`. Para excluir-se uma tela, executa-se o comando `Arquivos, Remover do projeto`. As telas que compõem o sistema podem ser visualizadas através da tecla `F10` ou do menu.

Concluída a construção das telas, passa-se para a fase de configuração do sistema, de modo que o mesmo possa ser executado.

5.2. Kernel

Por tratar-se do núcleo do sistema, deve ser configurado e executado antes do `Viewer`, caso contrário o `Viewer` não terá com quem se conectar. Para configura-lo executa-se o comando do menu `Principal, Setup`, que exibirá a tela de setup do `Kernel`, onde as configurações deverão ser realizadas, conforme a figura 3.17.

5.3. Viewer

Com o Kernel em execução, pode-se configurar o Viewer de modo que se integre ao sistema. Para configura-lo é utilizado o comando Setup do menu Principal, que exibirá a tela de Setup do Viewer, conforme a figura 4.9.

Com o Viewer conectado ao sistema, o usuário pode começar a alterar e a supervisionar a residência através do sistema. Através do comando do menu Principal, Logs o usuário visualiza os registros de logs, exibidos na tela 4.7.

Com o botão esquerdo do mouse o usuário consegue ativar e desativar qualquer equipamento assistido pelo sistema, e com o botão direito, consegue-se visualizar os outros estados do componente e as regras armazenadas para aquele componente, através da utilização do editor de propriedades exibido na figura 4.3 e do editor de regras exibido na figura 4.4.

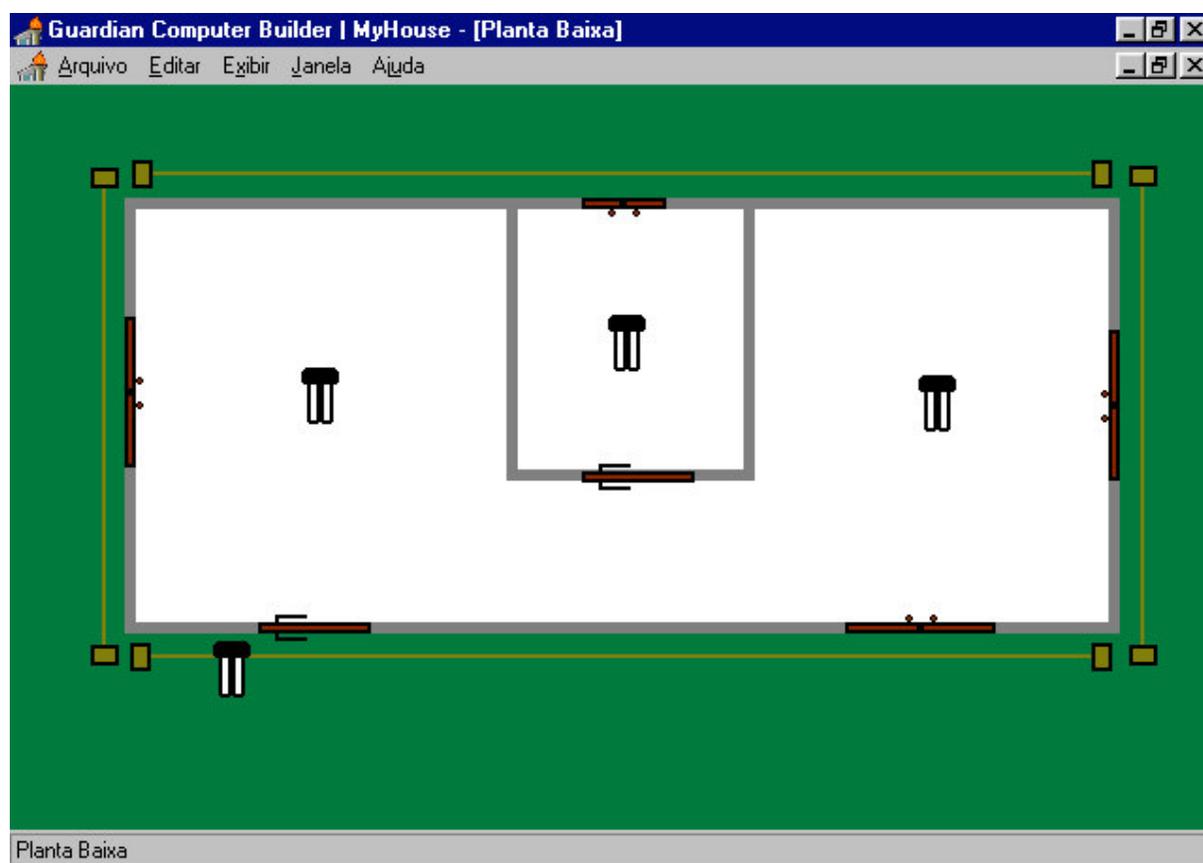


Figura 5.1 : Planta Baixa

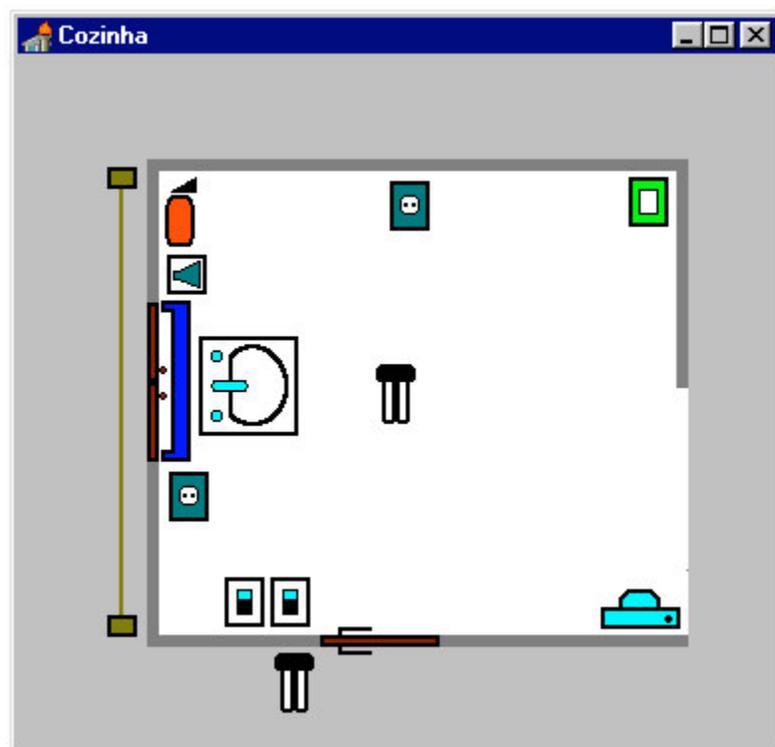


Figura 5.2 : Cozinha

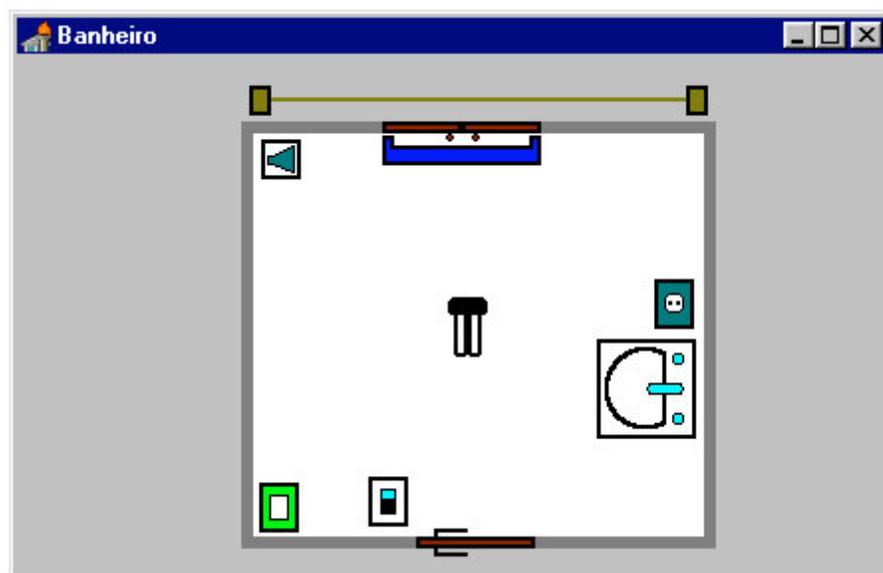


Figura 5.3 : Banheiro

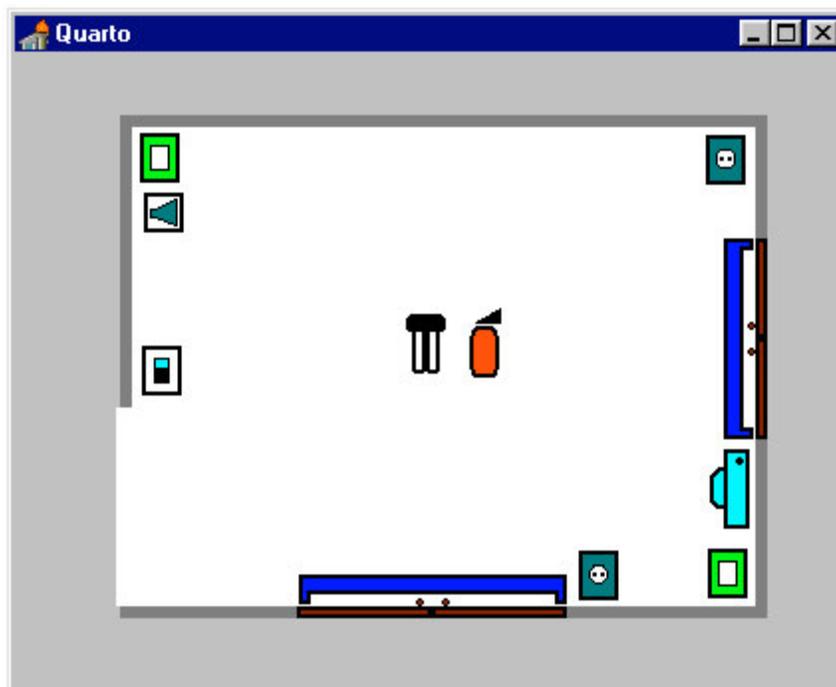


Figura 5.4 : Quarto

6. Conclusão

As residências tradicionais estão dando lugar a uma nova categoria de residências, as chamadas residências inteligentes, que conseguem obter informações do ambiente onde estão inseridas e atuar no ambiente com base nestas informações.

Há um enfoque para a segurança e o conforto dos seus moradores, considerando em primeiro lugar o morador, e permitindo uma economia dos recursos, tais como água e energia, além de valorizarem o imóvel.

As tentativas de construir residências totalmente automatizada, esbarravam na indisponibilidade de equipamentos que realizassem tarefas de controle e sistemas que interligassem estes equipamentos. Muitos dispositivos foram desenvolvidos com o passar dos anos, porém operando independentes uns dos outros.

A integração dos equipamentos foi a segunda fase da evolução, surgindo no mercado uma série de arquiteturas para interliga-los, como as arquiteturas americanas X-10, CEBus e Smart House , e as arquiteturas européias, EIB, BatiBus e European Home Systems, entre outras.

Estas arquiteturas interligam uma série de dispositivos, como lâmpadas e sensores, feitos especialmente para determinada arquitetura, mas excluía os eletrodomésticos, que com o decorrer do tempo evoluíram a ponto de serem considerados inteligentes, como por exemplo, os fornos de microondas que calculam automaticamente o período de cozimento dos alimentos.

A Sun Microsystems, Inc. está atualmente desenvolvendo uma arquitetura de comunicação capaz de interligar os eletrodomésticos, utilizando-se processadores nos equipamentos, que gerenciariam a comunicação e a execução das tarefas. Através do protocolo, computadores poderiam enviar e receber informações dos eletrodomésticos, integrando-os num grande sistema. Esta arquitetura chama-se Jini, e deverá estar disponível no mercado por volta do próximo ano.

O sistema descrito no presente trabalho, se propõem a ser o gerenciador central dos equipamentos, ou seja, o cérebro da residência, interagindo com todos os equipamentos através de uma arquitetura e possibilitando ao usuário o controle centralizado dos mesmos.

Não preocupou-se em desenvolver uma arquitetura de interligação com os equipamentos, apenas especificou-se uma arquitetura básica para dar suporte ao sistema. Os equipamentos conectados ao sistema devem poder realizar suas tarefas independentes do

sistema, pois o mesmo apenas envia o pedido da ação, e não os passos necessários para executá-la. Por exemplo, o equipamento responsável pela abertura da porta, deve ser capaz de abri-la sozinho quando receber do sistema a ordem, e não esperar do sistema todos os comandos para executar a operação de abrir a porta, como acionar os motores, verificar sua posição entre outras tarefas necessárias.

A inserção de novos equipamentos foi uma preocupação no desenvolvimento, já que todo dia surgem novos equipamentos que auxiliam a automação. Através do paradigma da orientação a objetos, conseguiu-se definir um conjunto de características básicas que são utilizadas por todos os equipamentos, de forma que os novos precisam apenas colocar as suas peculiaridades no sistema.

A meta na construção do sistema foi deixá-lo o mais independente possível da arquitetura de interligação dos equipamentos, de forma a facilitar a utilização de qualquer uma das arquiteturas, como X-10, Jini, EIB, entre outras. Isso é atingido através do componente básico para todos os equipamentos, que encapsula as características da arquitetura a respeito dos equipamentos.

A integração do sistema com a Internet é uma característica muito interessante, que poderá ser realizada num futuro próximo. Com o desenvolvimento do Viewer para ser executado dentro de um browse, o usuário conseguiria conectar-se facilmente ao sistema de qualquer parte do mundo, interagindo remotamente com o sistema.

O grande desafio da integração com a Internet é a questão da segurança. Barreiras de segurança devem ser implementadas e testadas de forma a garantir que estranhos não invadam o sistema e nem recebam as informações dos estados dos equipamentos, podendo-se por exemplo, utilizar a criptografia para a transmissão dos dados.

O custo necessário para automatizar uma residência é atualmente muito alto, devido ao preço dos equipamentos e a reforma necessária na residência para se instalar o sistema. Arquitetos já projetam residências que suportam a automação, através da instalação das tubulações especiais para a passagem dos cabos que irão interligar os equipamentos.

Com a popularização da automação residencial, os preços deverão cair e novas tecnologias irão aparecer, tanto na arquitetura, nos equipamentos e principalmente nos softwares de gerenciamento.

Técnicas de inteligência artificial deverão ser aplicadas aos sistemas de automação residencial, incorporando sistemas especialistas no gerenciamento de um residência.

Ferramentas de reconhecimento de imagem e voz serão agregadas aos sistemas, de forma que o mesmo reconheça os moradores da residência e aceite comandos falados.

A possibilidade de auto-aprendizagem também deverá ser um diferencial nos futuros sistemas, possibilitando ao mesmo adaptar-se da melhor forma possível ao ambiente onde está inserido, interagindo mais intimamente com o ambiente.

Logo será comum residências com alto grau de automação, utilizando-se sistemas muito mais inteligentes do que os atuais, e quem sabe num futuro não muito distante, alcançaremos a automação usufruída pelos Jetsons.

7. Referências Bibliográficas

CALDAS, Fausto Manuel Coutinho Pereira Caldas , PERDIDÃO, João Pedro Rebelo , PEREIRA, Pedro José Guerra Pessoa . **Sistema de Segurança para Edifícios**. Disponível da Internet. http://alumni.dee.uc.pt/~fausto/p0_proj.html . 04 jul 1998.

CAMPBELL, Joe. **C Programmer's Guide to Serial Communications**. 2. Ed. Sams Publishing, 1993.

CANTU, Marco. **Dominando o Delphi**. São Paulo : Makron Books, 1996.

COMER, Douglas. **Internetworking with TCP/IP**. 3 Ed. , Volume 1. Prentice-Hall, 1995.

HOME AUTOMATION ASSOCIATION. **The Automation Association**. Disponível na Internet. <http://www.homeautomation.org/> . 19 set 1998.

INPRISE. **Borland Delphi Developer Version 2.0** . Copyright 1983-96. Ambiente de Programação.

MARTE, Claudio Luiz. **Automação predial** : a inteligência distribuída nas edificações. São Paulo : Carthago & Forte, 1995.

MUELLER, John Paul. **Peter Norton – Guia para o Delphi 2**. São Paulo : Makron Books, 1997.

SCHULTZ, Thomas W.. **C and the 8051 Hardware, modular programming, and multitasking**. 2. Ed. Volume 1. Prentice Hall, 1998.

STEVENS, W. Richard. **TCP/IP Illustrated** : The protocols. Addison Wesley, 1994.

SUN MICROSYSTEMS, Inc. **Jini(tm)** . Disponível na Internet. <http://java.sun.com/products/jini/index.html> . 12 ago 1998.

TANENBAUM, Andrew S. **Redes de Computadores**. 3 Ed. Rio de Janeiro : Campus, 1997.

TRINDADE, Alessandro Bezerra. **Microcontroladores 32 bits e a Internet** : Uma proposta de Aplicação para Automação Industrial, Predial e Residencial. Disponível na Internet. <http://members.tripod.com/~albt/sumary.htm> . 19 set 1998.

X-10, Inc. **X-10 Product Manuals**. Disponível na Internet. http://www.x10.com/support_manuals.htm . 23 jun 1998.

WACKS, Kenneth P.. **Introduction to the CEBUS® Communications Protocol**. Disponível na Internet. <http://www.hometoys.com/htinews/aug97/articles/kwacks/kwacks.htm> . 22 set 1998.