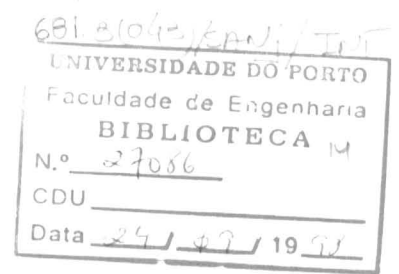


Jorge M. Fernandes dos Santos

A Interface Com o Utilizador
em Sistemas Domóticos

DEEC
FEUP
1997



FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Departamento de Engenharia Electrotécnica e de Computadores

A Interface com o Utilizador em Sistemas Domóticos

Jorge Manuel Fernandes dos Santos

Licenciado em Engenharia Informática Industrial pelo
Instituto Superior de Engenharia do Porto

Dissertação submetida para satisfação parcial dos requisitos do grau de mestre em
Engenharia Electrotécnica e de Computadores
(Área de especialização de Informática Industrial)

Dissertação realizada sob a supervisão de
Professor Doutor José Manuel Martins Ferreira,
do Departamento de Engenharia Electrotécnica e de Computadores
da Faculdade de Engenharia da Universidade do Porto

Porto, Setembro de 1997

Resumo

A domótica encontra-se numa fase de plena expansão.

Actualmente atingiu-se um estado em que os aspectos relacionados com a interacção com o utilizador têm vindo a receber maior atenção por parte dos projectistas e o tempo dispendido na implementação de novos sistemas é, em grande parte, dedicado à definição da interface com o utilizador e do modo como o sistema se deve adaptar a ele e não o contrário, o utilizador ter de se adaptar a um dado sistema com o qual não tem qualquer afinidade.

O utilizador deve, cada vez mais, ser capaz de instalar os dispositivos que pretende e, posteriormente, actualizar o seu sistema com novos dispositivos sem que para isso necessite da intervenção de um instalador especialista.

Este trabalho pretende constituir um estudo sobre as interfaces com o utilizador e a sua especificação e utilização nos sistemas domóticos. Ao abordar de uma forma geral a domótica e de uma forma mais pormenorizada as interfaces com o utilizador o autor pretendeu introduzir o primeiro tema e explorar o segundo de uma forma mais detalhada até que os dois se cruzem e formem um conjunto compreensível para o leitor.

A elaboração final de um *design* de interface com o utilizador para um sistema domótico foi o culminar de um estudo aprofundado nas duas áreas e um passo para, de futuro, se poder utilizar este documento como um contributo para todos os projectistas desta área.

Abstract

Home Automation are now in true expansion.

They reached a phase where aspects related to user interaction have been getting more attention from designers. Time spent in new systems implementation is largely dedicated to the definition of the user interface and to the way the system should adapt to the user and not the opposite.

The user should be more and more able to install the devices he wants and, later on, to update his system with new devices with no need for a specialist technician.

This work pretends to accomplish a study about user interfaces and its specification and effective use in home systems. Making a general approach to home automation and a more elaborated study of graphical user interfaces, the author pretended to introduce the first issue and to explore the second in a more detailed way to produce an understandable document to the reader.

The final execution of an user interface design to a home system has been the result of a deep research in both areas and pretends to be a contribution for future designing.

Prefácio

Este trabalho constitui um marco na vida do autor, procurando aglomerar os conhecimentos adquiridos nesta área, e servindo de patamar para novos investimentos na matéria da domótica e em particular na área das interfaces gráficas com o utilizador.

Desde sempre que o autor sentiu uma grande apetência para a domótica e num estágio mais recente, derivado da sua ligação à área do *design* e das aplicações para desenvolvimento de trabalhos gráficos, sentiu aumentar a preocupação com os aspectos relacionados com a interface gráfica com o utilizador e a forma como se efectuava a interacção entre ele e os dispositivos existentes.

Esta dissertação, apresentando-se como um culminar de várias etapas, consegue apresentar os elementos fundamentais das duas áreas abordadas e constitui um documento importante para todos aqueles que pretendam iniciar-se nestes dois domínios.

Sem o auxílio e apoio de várias pessoas este trabalho não teria sido possível.

Compete-me, por isso, agradecer em primeiro lugar ao meu orientador, Professor Doutor José Manuel Martins Ferreira por toda a atenção e tempo que me concedeu para orientação e revisão desta dissertação.

À minha esposa, Isabel, pelo incentivo inicial e por todo o apoio e carinho que sempre me dedicou, a toda a minha família e especialmente ao meu irmão Rolando que me acompanhou na mesma edição de mestrado.

Finalmente devo agradecer o apoio dado pelo Instituto Politécnico do Porto, através do seu Fundo de Apoio à Investigação, ao Instituto Superior de Engenharia do Porto pelo apoio logístico e aos meus colegas do Departamento de Matemática.

Índice

1 Introdução	18
1.1 Motivação para o trabalho apresentado	19
1.2 Organização da tese	20
2 Os Sistemas Domóticos.....	22
2.1 Terminologia e Envolvente Tecnológica	22
2.1.1 A Estrutura de Comunicações	24
2.1.2 Áreas Funcionais de um Sistema Domótico	25
2.2 Breve Resenha Histórica	29
2.2.1 Projectos Europeus	32
2.2.2 Projectos Americanos (EUA)	33
2.2.2 Projectos Asiáticos (Japão).....	35
2.3 Estado Actual da Tecnologia	36
2.3.1 Sistemas domóticos e produtos.....	39
3 A Interface com o Utilizador.....	42
3.1 A Evolução da Interface com o Utilizador.....	43
3.1.1 A interface WYSIWIG.....	45
3.1.2 Manipulação directa	45
3.2 Perspectivas Futuras	47
3.3 Apresentação da informação	50
3.3.1 Introdução	50
3.3.2 Apresentação da informação segundo as leis de Gestalt	51
3.3.2.1 Lei da condensação (terseness).....	52
3.3.2.2 Lei da proximidade (proximity)	52
3.3.2.3 Lei da similaridade (similarity).....	53
3.3.2.4 Lei da consistência (consistency).....	53
3.3.2.5 Lei da continuidade (continuity)	53
3.3.2.6 Lei da simetria (symmetry)	54
3.3.2.7 Lei das zonas de sobreposição (overlapping areas)	54
3.3.3 Apresentação da informação segundo técnicas visuais.....	54
3.3.3.1 Técnicas físicas.	55
3.3.3.2 Técnicas de composição.....	56
3.3.3.3 Técnicas de associação e dissociação.....	57

3.3.3.4	Técnicas de ordenação	58
3.3.3.5	Técnicas fotográficas	59
3.3.4	Apresentação da informação segundo técnicas estruturais	60
3.3.4.1	A estrutura das cenas visuais	61
3.3.5	Outros critérios	66
4	O Desenvolvimento de Interfaces com o Utilizador	68
4.1	O Processo de Desenvolvimento	68
4.1.1	Metodologias de Desenvolvimento	69
4.1.2	Fases do Processo de Desenvolvimento	72
4.1.2.1	Conhecimento do utilizador	73
4.1.2.2	Análise	74
4.1.2.3	Design da interface	74
4.1.2.4	Definição dos objectivos de usabilidade	75
4.1.2.5	Elaboração de Protótipos	76
4.1.2.6	Teste da Interface	76
4.2	Princípios de Concepção	80
4.2.1	Metáforas e Analogias	80
4.2.2	Regras Gerais de Concepção	83
4.2.2.1	Conformidade com as expectativas do utilizador	83
4.2.2.2	Tolerância ao erro e prevenção do erro	85
4.2.2.3	Controlabilidade	87
4.2.2.4	Possibilidade de individualização	89
4.2.2.5	Auto descrição / Aprendizagem	90
4.2.2.6	Documentação	93
4.2.3	Desenho de ícones	95
4.2.4	O uso das cores	97
4.2.4.1	Associação semântica e codificação	99
4.2.4.2	Codificação redundante com o uso da cor	99
4.2.4.3	Regras no uso das cores	100
4.2.4.4	Vantagens e desvantagens no uso da cor	102
4.2.5	O uso do som e voz	104
4.2.5.1	O uso de som	104
4.2.5.2	Interfaces com uso da voz	106
4.3	Ferramentas de apoio ao desenvolvimento de GUIs	109
4.3.1	Classificação das ferramentas	110
4.3.2	Vantagens e desvantagens no uso das ferramentas	112
4.3.3	A escolha de uma ferramenta	114
4.3.4	Ferramentas específicas de apoio ao desenvolvimento	116
5	Especificação de Requisitos da G.U.I. nos Sistemas Domóticos	124
5.1	Introdução	124

5.2 Output de texto.....	126
5.2.1 Regra geral.....	126
5.2.2 Legendas, abreviaturas, designações.....	126
5.2.3 Mensagens.....	127
5.2.3.1 Descrição.....	127
5.2.3.2 Aplicação.....	127
5.2.3.3 Apresentação.....	127
5.2.3.4 Interacção.....	127
5.2.4 Mensagem de informação.....	128
5.2.5 Mensagem de aviso.....	128
5.2.6 Mensagem de alarme.....	129
5.3 A Auto-Repetição.....	130
5.3.1 Introdução.....	130
5.3.2 Conceitos.....	130
5.3.2.1 Intervalo inicial e intervalo de repetição.....	130
5.3.2.2 Incremento / Decremento.....	131
5.3.2.3 Fases de repetição.....	131
5.3.3 Recomendações.....	132
5.3.3.1 Intervalo inicial.....	132
5.3.3.2 Incremento/Decremento.....	132
5.3.3.3 Intervalo de repetição e fases.....	132
5.3 Elementos de Controlo.....	133
5.4.1 Elementos de Controlo por Software.....	133
5.4.1.1 Lista de Opções (Drop Down List).....	133
5.4.1.2 Campo de Saída.....	135
5.4.1.3 Campo de Entrada.....	136
5.4.1.4 Caixas de Selecção (Check Box).....	139
5.4.1.5 Botão Rotação (Spin).....	140
5.5 Elementos básicos comuns.....	141
5.5.1 Os Botões.....	142
5.5.1.1 Botão OK.....	142
5.5.1.2 Botão CANCELAR ou CANCEL.....	142
5.5.1.3 Botão MAIS.....	143
5.5.1.4 Botão MENOS.....	143
5.5.1.5 Botão BAIXO (SEGUINTE).....	144
5.5.1.6 Botão CIMA (ANTERIOR).....	144
5.5.1.7 Botão DIREITA.....	145
5.5.1.8 Botão ESQUERDA.....	145
5.5.1.9 Botão ELIMINAR ou CLEAR.....	146
5.5.1.10 Botão APAGAR.....	146
5.5.1.11 Bloco Numérico.....	147
5.5.2 Combinações.....	148

5.5.2.1 Regras gerais de combinação.....	148
5.5.2.2 MAIS e MENOS	148
5.5.2.3 CIMA (ANTERIOR) e BAIXO (SEGUINTTE).....	149
5.5.2.4 MAIS, DIREITA e MAIS , DIREITA e ELIMINAR	149
5.5.2.5 MENOS, MAIS, DIREITA e ESQUERDA ou MENOS, MAIS, DIREITA, ESQUERDA e ELIMINAR.....	149
5.5.2.6 MENOS, MAIS, CIMA e BAIXO ou MENOS, MAIS, CIMA, BAIXO e ELIMINAR	150
6 Protótipo da Interface com o Utilizador da Aplicação WinDomo.....	151
6.1 O Projecto da Aplicação WinDomo	151
6.2 Estrutura da Aplicação WinDomo.....	152
6.3 A Interface com o Utilizador.....	154
6.3.1 O Menu.....	155
6.3.2 Exemplos de Interação.....	159
6.4 Considerações finais sobre a interface	172
7 Conclusão.....	174
Referências Bibliográficas	177
Anexos.....	185

Lista de Figuras

Figura 1 - Valores médios percentuais da redução de custos de energia. (fonte: Honeywell).	28
Figura 2 - A primeira revolução domótica [Rubinstein93].	30
Figura 3 - A segunda revolução domótica [Rubinstein93].	30
Figura 4 - Evolução dos sistemas domóticos.	31
Figura 5 - A estrutura do HES.	39
Figura 6 - Etapas temporais, aproximadas, do trabalho realizado nas principais tecnologias aplicadas à interface com o utilizador.	46
Figura 7- Lei da condensação	52
Figura 8 - Lei da proximidade.	52
Figura 9 - Lei da similaridade.	53
Figura 10 - Lei da consistência.	53
Figura 11 - Lei da continuidade.	53
Figura 12 - Lei da simetria.	54
Figura 13 - Lei das zonas de sobreposição	54
Figura 14 - Alinhamento e desalinhamento vertical.	56
Figura 15 - Layout simples <i>versus</i> layout complexo.	56
Figura 16 - Negatividade <i>versus</i> positividade.	57
Figura 17 - Unicidade <i>versus</i> fragmentação.	57
Figura 18 - Leveza <i>versus</i> ultra-densidade.	58
Figura 19 - Previsibilidade <i>versus</i> imprevisibilidade.	58
Figura 20 - Arredondamento <i>versus</i> angulosidade.	59
Figura 21 - Rectângulo estável; círculo, losango, rectângulo e triângulo instáveis.	59
Figura 22 - Eixos de equilíbrio do <i>layout</i> .	60
Figura 23 - Diagrama estrutural de uma cena.	61
Figura 24 - Diagrama de transições.	62
Figura 25 - Dois painéis de controlo que diferem na forma como os objectos estão agrupados.	62
Figura 26 - Diagrama de transições referente aos painéis de luzes (a) e aquecimento (b).	63
Figura 27 - Objectos com características diferentes sobressaem.	64
Figura 28 - Objectos maiores ou mais brilhantes tornam-se sujeitos pragmáticos.	64
Figura 29 - A estrutura afecta o agrupamento.	64
Figura 30 - O conhecimento pode afectar a estrutura.	65
Figura 31 - Processo de feedback entre o objecto e o seu significado.	65
Figura 32 - A cor é dominante sobre a forma.	66

Figura 33 - Continuidade temática da estrutura.....	66
Figura 34 - Categorias de Comunicação entre utilizador e projectista segundo Kensing.....	72
Figura 35 - Um Ciclo Iterativo no processo de desenvolvimento [Gulliksen95b].	73
Figura 36 - Exemplo de uma mensagem de erro.....	86
Figura 37 - A aparência das cores varia conforme a sua relação com outras cores. Cada "X" parece ser da cor do fundo contrário, no entanto os dois são da mesma cor. ...	98
Figura 38 - Os vários níveis de <i>Software</i> de Interface.....	111
Figura 39 - Estrutura de um UIMS.....	112
Figura 40- Nas mensagens de informação deve ser usado o símbolo "i".....	128
Figura 41 - Nas mensagens de aviso deve ser usado o símbolo "!".....	129
Figura 42 - Nas mensagens de alarme deve ser usado o símbolo "Stop"	129
Figura 43 - Exemplo da função de auto-repetição.....	131
Figura 44 - Um exemplo de lista de Opções (Drop Down List).....	134
Figura 45 - Um exemplo de campos de saída e de entrada.....	135
Figura 46 - Campos de entrada.....	136
Figura 47 - As legendas e os campos de entrada devem ser alinhados à esquerda.....	137
Figura 48 - Relação entre as legendas e os campos de entrada.....	137
Figura 49 - Organização espacial dos campos de entrada.....	137
Figura 50 - Caixas de selecção.....	139
Figura 51 - Botão de rotação (<i>Spin Button</i>).....	140
Figura 52 - A Estrutura do sistema domótico.....	152
Figura 53 - Estrutura da aplicação WinDomo	153
Figura 54 - A janela inicial da interface.....	154
Figura 55 - Os vários itens do submenu "Ficheiro".....	155
Figura 56 - Mensagem de aviso.....	155
Figura 57 - Os itens do submenu "Editar".....	156
Figura 58 - Os itens do submenu "Novo".....	157
Figura 59 - As áreas em que se divide o submenu "Objecto".....	157
Figura 60 - Janela de criação de um novo botão "Grupo" ou "Local".....	158
Figura 61 - A janela de criação de um novo botão "Objecto".....	158
Figura 62 - O submenu "Ajuda".....	159
Figura 63 - Painel de "Luz/Estores".....	160
Figura 64 - Painel de luzes e estores do "Quarto de casal".....	161
Figura 65 - Janela de diálogo relativa ao controlo da luminosidade.....	162
Figura 66 - Janela de diálogo relativa à regulação do estore.....	162
Figura 67 - O painel de "Climatização".....	163
Figura 68 - A janela de diálogo para definir as temperaturas ao longo do dia.....	163
Figura 69 - Janela de diálogo para climatização de cada local.....	164
Figura 70 - O painel das tomadas da cozinha.....	165
Figura 71 - O painel de "Lazer".....	165
Figura 72 - O painel "Electrodomésticos".....	166

Figura 73 - A janela de diálogo do Micro-Ondas.....	167
Figura 74 - O painel "Cenários".....	168
Figura 75 - A janela de diálogo para criar um novo cenário.....	168
Figura 76 - O painel "Cenários" depois da criação de um novo cenário (Boa Noite).....	169
Figura 77 - A janela de diálogo de alarme contra intrusos.....	170
Figura 78 - Parte do painel de comunicações.....	170
Figura 79 - A janela de diálogo do telefone.....	171
Figura 80 - A janela de diálogo de "Programação".....	172
Figura 81 - A estrutura do sistema domótico.....	186
Figura 82 - O formato da trama da comunicação PC-Módulo.....	187
Figura 83 - Formato da trama de resposta.....	188

Lista de Tabelas

Tabela 1 - Algumas das definições de "casa inteligente".....	23
Tabela 2 - Recomendações para instalação de cablagem.....	36
Tabela 3 - Relação entre as cores, a associação efectuada e o seu uso preferencial [FACE94].....	99
Tabela 4- Termos sobre a cor, definições e exemplos [Anthony96a].....	100
Tabela 5 - Regras fisiológicas no uso da cor.	101
Tabela 6 - Regras de percepção no uso da cor.	101
Tabela 7 - Regras cognitivas no uso da cor.....	102
Tabela 8 - Exemplo de auto-repetição com três fases.....	132
Tabela 9 - Recomendações para parâmetros da função de Auto-repetição.....	133

Lista de Acrónimos

ARPA	<i>Advanced Research Projects Agency</i>
AVAC	<i>Aquecimento, Ventilação e Ar Condicionado</i>
BCI	<i>BatiBus Club International</i>
CCITT	<i>Comité Consultatif International de Téléphonie et de Télégraphie</i>
CD	<i>Compact Disk</i>
CD-ROM	<i>Compact Disk Read Only Memory</i>
CIC	<i>CEBus Industry Council</i>
CSMA/CD	<i>Carrier Sense Multiple Access / Collision Detection</i>
CSTB	<i>Centre Scientifique et Technique du Bâtiment</i>
CUI	<i>Cultural User Interface</i>
EHS	<i>European Home System</i>
EHSA	<i>European Home System Association</i>
EIA	<i>Electronics Industries Association</i>
EIBA	<i>European Installation Bus Association</i>
EIBus	<i>European Installation Bus</i>
GUI	<i>Graphical User Interface</i>
HES	<i>Home Electronic System</i>
HSS	<i>Home Systems Specification</i>
ISO	<i>International Standards Organization</i>
LCD	<i>Liquid Crystal Display</i>
NAHB	<i>National Association of Home Builders</i>
NC	<i>Network Computers</i>
OEM	<i>Original Equipment Manufacturer</i>
OI	<i>Objectos de Interacção</i>
OSI	<i>Open Systems Interconnections</i>
PBX	<i>Private Branch Exchange</i>

PC	<i>Personal Computer</i>
RF	<i>Rádio Frequência</i>
TTY	<i>Teletype Terminal</i>
UI	<i>User Interface</i>
UIMS	<i>User Interface Management System</i>
VCR	<i>Video Cassette Recorder</i>
VHS	<i>Very High Speed</i>
WIMP	<i>Windows, Icons, Mouse, Pull-down menus</i>
WWW	<i>World Wide Web</i>
WYSIWIG	<i>What You See Is What You Get</i>

Glossário

- Camcorder* Câmara de vídeo.
- Chip* Dispositivo constituído por materiais semi-condutores (usualmente silício) que é produzido contendo uma grande quantidade de circuitos interligados. São bastante usados em processadores e memórias assim como em produtos industriais e domésticos.
- Click* Acção que consiste em pressionar o botão do rato (dispositivo de *input*) por forma a enviar um sinal ao sistema. Esta acção está relacionada com a posição do cursor (sinal que indica a zona onde se vai efectuar a interacção) e também com os objectos de interacção (OI).
- Clipboard* Zona de transferência de informação entre aplicações. Quando o utilizador efectua um “*copy*” a informação copiada é transferida para o *clipboard* e fica disponível para poder ser introduzida noutra aplicação.
- Display* Qualquer tipo de dispositivo de apresentação de informação. Monitor, ecrã de televisão, LCD, painel de plasma, painel de um electrodoméstico, etc.
- Drive* Periférico de um computador que serve para a leitura e escrita de dados. O tipo de meio usado pode ser o disco, disquete, fita magnética, CD-ROM, disco óptico, etc.
- Input* Sentido da transmissão da informação do utilizador para o sistema. A transmissão no sentido contrário é o *Output*.
- Internet* Rede de computadores a nível mundial. Deve o seu nome ao protocolo de comunicações utilizado (*Internet Protocol*).

- Layout* A forma como a informação é disposta num determinado dispositivo de apresentação.
- On Screen Display* Apresentação da informação do sistema no ecrã de televisão.
- Output* Sentido da transmissão da informação do sistema para o utilizador. A transmissão no sentido contrário é o *Input*.
- Palette* Paleta com um determinado número de cores num dado sistema.
- Pixel* Abreviatura de *Picture Element*. O mais pequeno elemento discreto de um *display* visual constituído por uma grande quantidade de pontos, como o ecrã de uma televisão ou o monitor de um computador. Estes *displays* são constituídos por uma grelha de pontos que apresentam, cada um, a sua cor. O tamanho do pixel é um factor fundamental na resolução de um monitor.
- Play* Acção de inicio da leitura em estado normal de um dado dispositivo. A tecla de “*play*” de um VCR, de um leitor de CD’s ou de um leitor de cassetes são exemplos deste tipo de acção.
- Plug and Play* (Ligar e Usar) Tipo de dispositivos que são reconhecidos pelo sistema efectuando a sua auto-configuração sem a intervenção de um técnico.
- Prompt* Sinal que é mostrado ao utilizador indicando que o sistema está pronto a receber informação através dos *inputs* daquele. Nos antigos sistemas baseados no sistema operativo MS-DOS o *prompt* era geralmente constituído pela letra do disco actual seguido pelo sinal “>”. O *prompt* pode ser também uma indicação sonora ou visual noutra tipo de dispositivos.
- Standard* Especificação da forma como devem ser efectuadas certas operações. Na indústria da tecnologia de informação existem dois tipos de *standard*: o oficial, promulgado pela instituições oficiais de estandardização como a ISO ou o CCITT, e os de facto, definidos pelas empresas líderes de mercado, como a IBM.

- Time out* Como o próprio nome indica o significado desta expressão é: acabou o tempo. Quando o tempo predefinido para uma dada acção é ultrapassado, o sistema reage autonomamente por *time out*, ou seja, porque o tempo dado ao utilizador para agir terminou.
- Touch-screen* Um monitor que consegue detectar quando o utilizador toca uma área específica deste. Estas áreas contêm sensores (quer por infravermelhos, quer capacitivos) sobrepostos no ecrã de uma forma imperceptível e que recebem os *inputs* do utilizador.
- Usability* A maior ou menor facilidade com que um determinado dispositivo é manuseado pelo utilizador. Neste trabalho é usada a palavra “usabilidade” como tentativa de tradução de *usability*. O quanto um determinado produto é mais ou menos fácil de usar.
- User friendly* Designação que é atribuída às interfaces que são de uso bastante simples e que foram pensadas para que o utilizador as manejasse de forma expedita. Tradicionalmente usa-se a designação “amigável” para traduzir este conceito.
- Video on Demand* Sistema recente em que o utilizador indica, através de uma rede de transmissão de dados geralmente combinada com o uso da televisão qual o filme que deseja ver.
- Wizard* “Feiticeiro”. Utilitário que ajuda o utilizador na execução de uma determinada tarefa. Geralmente os *wizards* são usados para auxiliar o utilizador na instalação de novas aplicações.

Capítulo 1

Introdução

A domótica é, actualmente, uma área em plena expansão. Depois de, nos últimos anos, se ter presenciado uma certa relutância à introdução de sistemas domóticos nas novas habitações, existe, hoje em dia, a noção generalizada de que a interligação no lar dos dispositivos tecnologicamente avançados conduz a melhorias em áreas como o conforto, a gestão de energia, o entretenimento e a segurança.

A dificuldade em produzir dispositivos e sistemas domóticos reside, basicamente, na falta de um verdadeiro *standard*. O esforço de interligação dos grandes grupos envolvidos no desenvolvimento e aperfeiçoamento dos novos sistemas tem conduzido a resultados que fazem prever, para breve, um *standard* a nível europeu.

A par da ausência de *standards* está a reacção negativa do utilizador comum principalmente em relação à forma como se efectua a interacção com o sistema. As aplicações de controlo do sistema continuam a apresentar graves falhas no campo da interface com o utilizador. A incapacidade das aplicações se adaptarem ao utilizador e de ser ele quem possa instalar novos dispositivos, através da utilização de elementos PnP (*Plug and Play*), tem, também, contribuído para a falta de aceitação por parte dos potenciais utilizadores.

A interface com o utilizador continua a ser um dos aspectos principais da reacção positiva ou negativa por parte do utilizador. Uma interface fácil de usar e com grande apelo estético conduz, geralmente, a uma boa aceitação. Não se pode dizer que as características que fazem de uma interface gráfica com o utilizador uma “boa interface” estejam completamente definidas mas, no entanto, existem alguns princípios básicos que interessa seguir para que as hipóteses de atingirmos um resultado satisfatório sejam mais elevadas.

Com este trabalho pretendeu-se efectuar um estudo sobre os aspectos relacionados com as interfaces com o utilizador, o seu desenvolvimento e a sua utilização em sistemas domóticos.

1.1 Motivação para o trabalho apresentado

Quando o autor, no seu projecto final de curso, realizou uma aplicação para controlo de um sistema domótico, apresentado com o suporte de uma maqueta de uma habitação, desde logo surgiram certas dúvidas sobre a melhor forma de realizar a interface gráfica com o utilizador. Esta área tem sido em parte deixada para um segundo plano nos curricula dos cursos de engenharia e tem conduzido a algumas dificuldades no desenvolvimento das interfaces de novas aplicações.

O facto de, neste curso de mestrado, ter frequentado a disciplina de Infra-estruturas para Edifícios Inteligentes, veio relembrar dificuldades antigas para a área da interface com o utilizador. Os sistemas actuais e as diversas tentativas da constituição de *standards* apresentam algumas lacunas na área da interface gráfica.

O estudo do autor sobre este tema deve-se, em parte, a este facto, assim como ao de existir um grupo, onde este trabalho se inseriu, com conhecimentos e alguma massa crítica no domínio da domótica.

Constata-se, também, que uma das motivações foi a de tentar apresentar uma interface com o utilizador com alguma independência face ao tipo de tecnologia empregue, o tipo de barramento, o protocolo de comunicações utilizado, etc.

Convém, talvez, referir aqui que, desde a altura em que este projecto foi idealizado, se deram novas evoluções neste campo e a introdução, para breve, de um sistema desenvolvido por um grupo de trabalho da Bosch-Siemens, o HES (*Home Electronic System*, [***]), com características de interface gráfica com o utilizador bastante avançadas, conduzirá concerteza a resultados ao nível da aceitação pelo utilizador comum bastante interessantes.

1.2 Organização da tese

A tese aqui apresentada divide-se em sete capítulos encontrando-se cada um deles dividido nos respectivos subcapítulos.

Após este capítulo introdutório onde se explica sucintamente o objectivo deste trabalho, a motivação do autor para o mesmo e se apresenta a estrutura da tese, os capítulos seguintes abordam já os temas da domótica e das interfaces gráficas com o utilizador.

No segundo capítulo efectua-se a introdução à área da domótica, nos aspectos relacionados com a terminologia e tecnologia empregues, seguindo-se uma breve resenha histórica, abordando os principais projectos neste campo e terminando com uma apresentação sobre o estado actual da tecnologia.

O terceiro capítulo introduz a área da interface com o utilizador expondo os aspectos relacionados com a sua evolução e o tipo de interfaces mais usadas até aos dias de hoje. As perspectivas futuras são também apresentadas neste capítulo onde se procura identificar o rumo seguido no desenvolvimento de novas interfaces gráficas. O capítulo termina com o estudo dos principais aspectos relacionados com a apresentação da informação em interfaces com o utilizador, constituindo, uma das principais partes deste trabalho.

No quarto capítulo desenvolve-se o tema do desenvolvimento das interfaces gráficas com o utilizador, desde os diferentes processos de desenvolvimento, as principais metodologias, as suas fases, passando pelos princípios de concepção, outra das partes importantes desta área, onde se apresentam as regras gerais de concepção, as metáforas e analogias, o desenho de ícones, o uso das cores e do som e voz. Este capítulo termina expondo as principais ferramentas de apoio ao desenvolvimento de interfaces gráficas com o utilizador, desde a sua classificação, a vantagem e desvantagem do seu uso, a escolha e a apresentação de algumas ferramentas actuais.

O quinto capítulo pretende ser uma parte onde se apresentam algumas regras a seguir na definição de interfaces com o utilizador em sistemas domóticos. Numa primeira parte abordam-se os aspectos relacionados com a apresentação de texto, as legendas e as mensagens, passando pela função de auto-repetição, bastante utilizada em dispositivos comuns, e terminando com a apresentação dos principais elementos de controlo por *software* e dos elementos básicos comuns que se devem usar na interface por forma a torná-la mais familiar e fácil de usar.

No capítulo sexto, é apresentado o *design* de um protótipo de aplicação, com principal ênfase na interface gráfica, onde se tenta explicar o seu funcionamento e interacção com o utilizador. Este capítulo fará também as vezes de manual de utilização, efectuando uma explicação sucinta de cada uma das acções possíveis, as diversas funções disponíveis e a forma de as utilizar.

Esta tese termina com o capítulo de conclusão, onde se tenta efectuar uma auto-análise

crítica ao trabalho desenvolvido assim como as perspectivas de trabalho futuro no sentido de implementar, melhorar e aperfeiçoar a aplicação para controlo do sistema domótico.

Capítulo 2

Os Sistemas Domóticos

Este capítulo efectua uma breve apresentação da domótica, começando com a consideração das questões relacionadas com a terminologia e as características principais da tecnologia empregue, apresentando a seguir uma história sumária do tema, apesar de recente, e concluindo com uma descrição concisa do estado da tecnologia nesta área.

2.1 Terminologia e Envoltente Tecnológica

A domótica é uma área relativamente recente. O conceito de Domótica surgiu no fim da década de 70 e, numa primeira abordagem, era entendida como a aplicação à habitação das mais recentes tecnologias. Contudo, o uso isolado de produtos com uma tecnologia mais avançada não introduz nenhum tipo de inteligência na habitação. Este conceito de inteligência, que inicialmente suscitou em muitas pessoas expectativas que não correspondiam à realidade, tem a ver com a interligação que deve existir entre os vários produtos e sistemas e não com funções e capacidades normalmente associadas aos humanos.

Vários autores têm vindo a definir a domótica ou habitação inteligente desde diferentes

pontos de vista. Algumas dessas definições encontram-se na Tabela 1 citada em [Rubinstein93].

Tabela 1 - Algumas das definições de "casa inteligente".

Natureza	Autor	Definição
Definição técnica	Akio Orihara Nikken Sekkei Co. Ltd	Imóvel que realiza integração de conforto, segurança, produtividade e economia graças aos mais recentes recursos tecnológicos.
Definição técnica	Profissionais franceses	Habitação que assegura um conjunto de serviços graças a sistemas que realizam várias funções que podem ser interligados entre si e a estruturas de comunicação internas e externas.
Definição sócio-técnica	Ken Sakamura casa TRON	Habitação que responde às necessidades criadas pela utilização crescente de aparelhos informatizados no seu interior e pelos novos estilos de vida dos habitantes.
Definição sócio-técnica	CSTB	Edifício capaz de fornecer uma tecnologia avançada de informação e de responder às necessidades de serviços dos seus ocupantes actuais e futuros.
Definição cultural	M. B. Ullah Singapura	Concepção arquitectural neo-modernista que dá corpo às aspirações sociais, económicas, culturais e tecnológicas da era da informação e dotando o imóvel de um coração em analogia com os organismos vivos.
Definição de eficácia cultural	Lawrence Chin Singapura	Imóvel que promove efectivamente e eficazmente o grau de desempenho e de produtividade requerido pelos ocupantes através de uma flexibilidade que lhe permite aco-modar novas funções e novas necessidades no futuro.

Actualmente a domótica pode definir-se como a disciplina que procura fornecer várias funcionalidades em áreas como segurança, conforto, gestão energética e comunicação, através de uma interligação inteligente da informação entre os vários sistemas tecnológicos e entre estes e o mundo exterior.

Um dos grandes objectivos da domótica é o de tornar a habitação num espaço que se adapte às necessidades dos seus ocupantes e que forneça uma certa flexibilidade de utilização.

A utilização de meios informáticos e a integração de funções permite que cada produto se adapte perfeitamente ao local a que se destina, possibilitando uma utilização mais racional e eficaz dos recursos disponíveis. Por exemplo, um sensor de presença pode, num sistema domótico, fornecer dados para a gestão da climatização e da iluminação. Outra das grandes vantagens que deve ser assegurada pelos sistemas domóticos reside na modularidade e facilidade de expansão, permitindo assim ao utilizador acompanhar a evolução de um mercado em constante crescimento.

Termos como “*Smart Building*” e “*High Tech Building*” foram usados nos primeiros tempos para definir o que hoje é conhecido como “Edifício Inteligente”. Este tipo de conceito, que apresenta alguma analogia com a domótica, é integrado na disciplina de Imótica. Esta disciplina trata de edifícios de grande porte, geralmente escritórios, instituições, centros comerciais, hospitais, hotéis, universidades, etc. Um edifício inteligente é aquele que foi concebido e construído por forma a oferecer uma grande flexibilidade de utilização, dispondo da capacidade de evoluir, de se adaptar às necessidades das organizações e de oferecer, em cada momento, o suporte mais adequado à sua actividade [Nunes95]. Por outro lado, e ainda segundo Nunes, o edifício inteligente deve possuir sistemas de automação, de computação e de comunicações que possibilitem, de um modo integrado e coerente, gerir de forma eficaz os recursos disponíveis no edifício, potenciando aumentos de produtividade, permitindo poupanças energéticas e oferecendo elevados graus de conforto e de segurança aos indivíduos que nele trabalham.

2.1.1 A Estrutura de Comunicações

Um dos primeiros factores a considerar num sistema domótico é a estrutura de comunicações. Tal como sucedeu com os sistemas de controlo gerais, os sistemas domóticos têm evoluído de arquitecturas centralizadas, em que de um computador central se efectuava uma ligação ponto a ponto para cada um dos dispositivos de controlo, para arquitecturas do tipo distribuído. Esta evolução deveu-se fundamentalmente aos seguintes factores:

- o número de pontos a controlar cresceu consideravelmente, tornando impossível suportar os custos de instalação, manutenção e desempenho daí resultantes.
- a complexidade e variedade dos dispositivos do sistema tornou-se tão elevada que a arquitectura centralizada se mostrou ineficaz no controlo deste tipo de equipamentos.
- Tornou-se necessária a existência de um maior número de pontos de controlo, dispositivos de interface, no interior e exterior da habitação.

O controlo em arquitecturas distribuídas é conseguido através de uma repartição de tarefas ou funções pelos vários sistemas que interagem entre si. A estrutura de comunicações passou a ser assim um dos factores principais de desenvolvimento dos sistemas domóticos. Existem diferentes estruturas de comunicação dependendo da sua topologia, meio de comunicação e tipo de acesso ao meio:

- ***Topologia***

- Estrela
- Anel
- Barramento (*Bus*)

- **Meio de Comunicação**

- Par Entrançado
- Cabo Coaxial
- Fibra Óptica
- Rede Eléctrica
- Infra Vermelhos
- Rádio Frequência

- **Tipos de Acesso ao Meio**

- Mestre-Escravo (*Master-Slave*)
- CSMA/CD
- Passagem de Testemunho (*Token*)

As estruturas de comunicação mais habituais são a topologia em *Bus*, com acesso CSMA/CD. O meio de comunicação depende muito de quem desenvolve os sistemas, dos dispositivos a utilizar, das infraestruturas da habitação e das potencialidades requeridas.

2.1.2 Áreas Funcionais de um Sistema Domótico

As funcionalidades que o sistema domótico pretende fornecer, já abordadas anteriormente, podem ser divididas nas seguintes áreas:

- **Segurança (sistemas de alarme, emergência, porteiro automático, ...)**

Esta é uma das áreas mais importantes, e a que atrai mais pessoas para a domótica. A insegurança actual sobre as pessoas e bens é um dos aspectos que mais preocupa o cidadão comum, pelo que se observa uma procura de sistemas que, de alguma forma, consigam transmitir alguma segurança aos utilizadores de uma habitação. A segurança pode reflectir-se sobre vários tipos de riscos, tanto internos (incêndio, fuga de gás, inundação, etc.) como externos (assalto). Para fazer a detecção de cada um destes agentes são usados detectores. No caso de incêndio, por exemplo, usam-se detectores de fumo ou calor.

A segurança é aumentada com a interligação dos vários sistemas. Quando o sensor de presença detecta um intruso, o sistema de iluminação pode reagir de forma a iluminar a área onde se efectuou a detecção, o sistema de comunicação pode enviar uma

mensagem telefónica para as forças policiais e outras acções podem ser tomadas por forma a evitar o assalto. A segurança começa pela prevenção. O sistema de iluminação pode simular a presença de ocupantes na habitação acendendo algumas luzes a horas pré-estabelecidas pelo utilizador, ligando o sistema de som e efectuando outras acções que procuram evitar que a habitação pareça desocupada.

- **Conforto e Entretenimento (iluminação, climatização, audio, vídeo, televisão, ...)**

O conforto começa a ser um dos aspectos relevantes para o utilizador. Este facto é constatado pelo crescente interesse em actividades de lazer.

O conforto pode ser entendido de várias formas [Santos94]:

- Conforto físico
- Condições de pureza e de higiene do ar ambiente
- Conforto termo-higrométrico

As condições de conforto de Verão, que se podem traduzir por uma temperatura de 26,5°C e humidade relativa de 70%, variam das de Inverno, temperatura de 20°C e humidade relativa de 60%.

- Conforto acústico
- Conforto visual

Os níveis de iluminação variam com o local. Apresentam-se, de seguida, alguns exemplos de níveis de luz necessários:

- Acessos e circulação - 30 lux
- Iluminação geral de ambiente (ao nível dos planos de trabalho) - 200 lux
- Leitura intermitente - 300 lux
- Leitura prolongada e estudo - 500 lux

As aplicações na área do conforto devem permitir que o utilizador possa controlar estes parâmetros de acordo com a sua sensibilidade e preferência.

O conforto é também o podermos ligar o nosso televisor, sentados no sofá, utilizando o controlo remoto, ou ainda acender e apagar as luzes, controlar a temperatura de um quarto ligando ou desligando os sistemas de climatização, ou deixando que o sistema domótico faça esse trabalho por nós. Podemos até realizar algumas das acções por comandos dados na forma oral.

A automatização de certas tarefas traz consigo uma diminuição da quantidade de tempo que perdemos a efectuar acções repetitivas como o ligar ou desligar a luz a certas horas

do dia, o abrir e fechar da garagem quando entramos ou saímos com o nosso automóvel, etc. Todos estes factores se juntam para a melhoria da qualidade de vida do utilizador de um sistema domótico.

Relativamente ao entretenimento, que é um dos factores que leva à aquisição de sistemas domóticos, podem imaginar-se diferentes situações. É possível distribuir o entretenimento pelas várias divisões da habitação. Uma cassette no vídeo ou um CD-ROM no computador podem ser vistos em qualquer televisão ou computador da casa. O controlo destes dispositivos, sendo remoto, permite que o utilizador “manipule” os aparelhos mesmo sem estar fisicamente próximo deles.

- ***Gestão Energética (consumos de água, electricidade, gás, ...)***

A energia necessária para fazer funcionar todos os equipamentos de uma habitação é um dos principais encargos na economia familiar. Um sistema de gestão de energia permite efectuar poupanças no consumo através, por um lado, de uma maior racionalização da sua utilização e, por outro, do recurso a outras fontes de energia alternativa. No primeiro caso, da racionalização, o objectivo é evitar o consumo desnecessário de energia. O facto de nos esquecermos de uma lâmpada ligada ou de o sistema de climatização funcionar ininterruptamente, mesmo quando não se torna necessário durante largos períodos, leva a consumos desnecessários de energia. Há também a possibilidade de restringirmos o horário de funcionamento de certos equipamentos às horas de tarifa mais baixa, tarifa bi-horária (p.e. máquinas de lavar roupa e louça, aquecimento de água, etc.). No segundo caso, devem, sempre que possível, ser implementados sistemas de energia alternativa, por exemplo, os vulgares painéis solares, e passar a usá-los alternativamente com os sistemas tradicionais de gás e electricidade para o aquecimento de água ou mesmo para o fornecimento de energia (memorizade) do sistema de gestão energética é tentar controlar cada uma destas situações com o objectivo de baixar os custos relacionados com o consumo de energia.

A redução dos custos de energia varia conforme o tipo de casa, a localização, os materiais de construção e os hábitos do utilizador. No entanto, reduções de cerca de 40% dos custos iniciais são possíveis de atingir (Figura 1), e o investimento inicial nos dispositivos de controlo é recuperado ao fim de algum tempo de uso.

4.0 %	Aquecimento de água	41.5 %
6.0 %	Controlo de iluminação	
31.5 %		
AVAC (Aquecimento, ventilação e ar condicionado)		

Figura 1 - Valores médios percentuais da redução de custos de energia. (fonte: Honeywell).

A HOME TEAM fornece um “guia para planeamento de recursos” [<http://www.hometeam.com/homenet/home.htm>] que permite verificar as potenciais reduções de custos em cada habitação pela introdução de sistemas domóticos.

- **Comunicações (telecontrolo, intercomunicadores, teletrabalho, ...)**

A comunicação, em termos de domótica, é uma área bastante abrangente. A comunicação pode significar a actividade existente entre os vários dispositivos do sistema, passando pela comunicação entre o utilizador e o próprio sistema, pela comunicação do sistema com o exterior e vice-versa ou ainda pela comunicação entre utilizadores.

O sistema de comunicações é o responsável por efectuar as trocas de informação entre os vários dispositivos de controlo e entre os vários sistemas. A activação de um sensor pode significar o envio de informação para um dispositivo, para o sistema de segurança, de climatização, de iluminação. O controlo dos vários actuadores por cada um dos sistemas é efectuado, também, pelo envio de informação através do sistema de comunicação.

O sistema de comunicação permite a troca de informação do interior para o exterior e vice-versa. O utilizador pode trabalhar em casa, simulando o ambiente de trabalho no seu computador pessoal (teletrabalho), pode controlar o sistema a partir de um telefone situado no exterior da habitação, pode contactar com sistemas de difusão de informação como a televisão interactiva, integrar as redes mundiais de computadores, enfim, efectuar um sem número de acções baseadas no sistema de comunicação.

Provavelmente, os maiores beneficiados deste sistema são a população com necessidades especiais, já que ele permite o controlo de qualquer dispositivo ou sistema a partir de qualquer ponto da habitação, inclusivamente da sua cadeira de rodas, se for essa a sua situação, ou o contacto com qualquer instituição exterior, como um centro de saúde ou hospital, ou ainda usufruir de actividades de entretenimento como a televisão interactiva ou o *Video On Demand*.

Outros equipamentos como o circuito interno de audio e vídeo, o correio electrónico, o

PBX, o telefone, o fax, etc., podem fazer parte do sistema de comunicação.

Esta área pode ser considerada a mais importante num sistema domótico, já que é sobre ela que assentam todas as outras.

Como já referimos anteriormente o objectivo final de um sistema domótico é efectuar a interligação destas quatro áreas.

Podemos ainda referir que a opção pela instalação de um sistema domótico, para além de introduzir melhoria no conforto e entretenimento, na gestão energética e na segurança, traz também algumas vantagens no aspecto ecológico, menos poluição e menos agressões ambientais, e introduz um valor acrescentado na habitação.

2.2 Breve Resenha Histórica

Segundo [Rubinstein93] a domótica não é um conceito novo. Para este autor a domótica iniciou-se pela tentativa de introdução na habitação de estruturas de apoio e de satisfação das necessidades básicas de conforto. Esta seria a primeira revolução domótica efectuada ainda no século XIX com a introdução de sistemas de distribuição de água, de energia e de gás, de sistemas de saneamento básico e de sistemas de aquecimento central.

Já no século XX, por volta dos anos 30, com a invenção e difusão da televisão, assiste-se à segunda revolução domótica, que visa as necessidades de comunicação, de cultura e lazer.

Pode, portanto, afirmar-se que a domótica tem mais de um século e que nos últimos anos atravessa uma nova revolução com o surgimento da electrónica, da informática e das telecomunicações.

Estes vários estádios da domótica estão representados na Figura 2 e Figura 3.

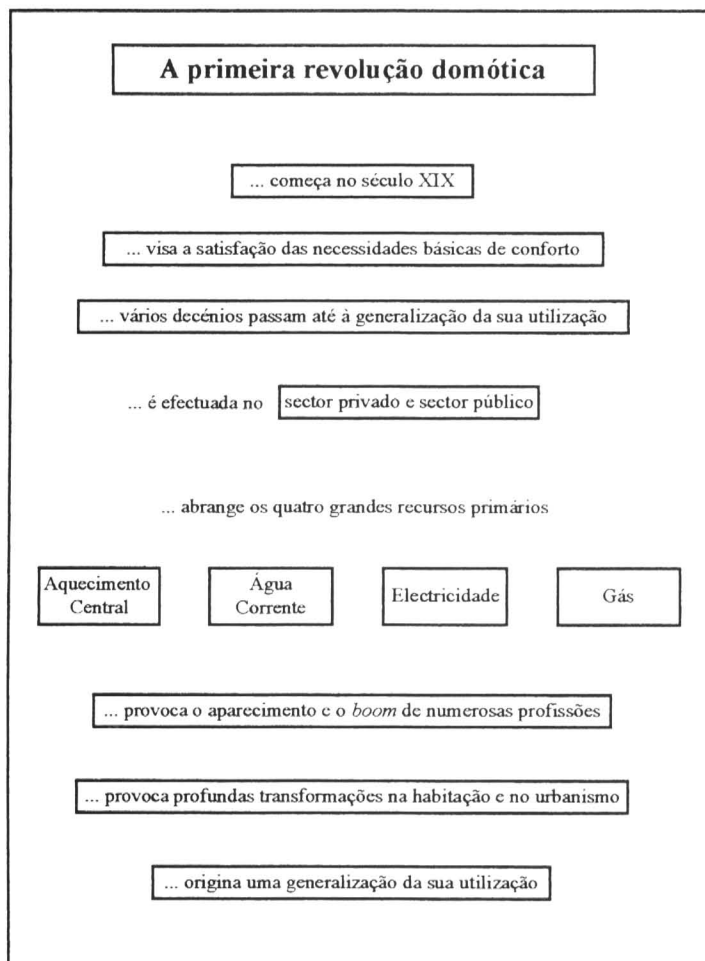


Figura 2 - A primeira revolução domótica [Rubinstein93].

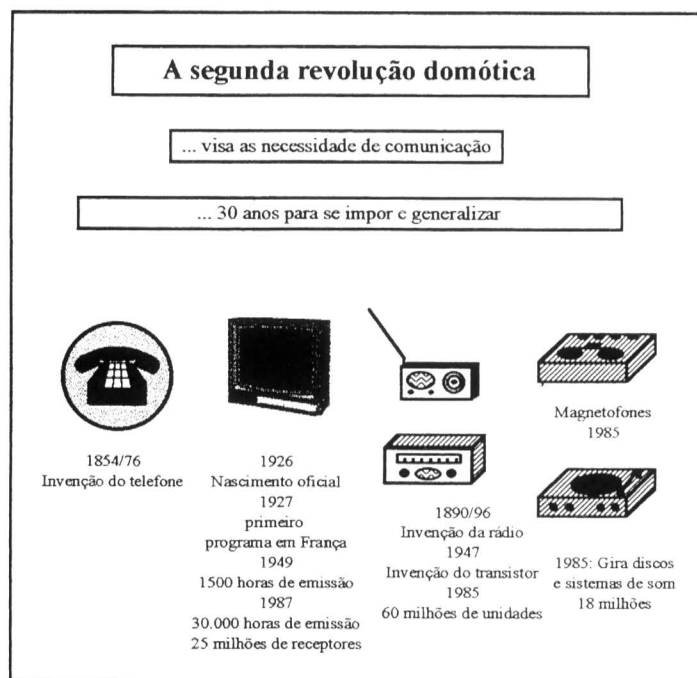


Figura 3 - A segunda revolução domótica [Rubinstein93].

Com o início da terceira revolução, em meados dos anos oitenta, verificou-se um aumento dos projectos de investigação direccionados para a área da domótica.

Com o aparecimento destes projectos pretendeu-se, também, caminhar no sentido da elaboração de *standards*. As empresas potenciais produtoras de sistemas domóticos têm bastante relutância em efectuar um elevado investimento em novos produtos sem a certeza de que eles vão ter aceitação no mercado, derivada da ausência de um *standard*. Basta recordarmos o que aconteceu com os sistemas de vídeo dos anos oitenta. Não existia um *standard* e os principais sistemas, Beta e VHS, concorriam para a liderança do mercado. Apesar do sistema Beta ser superior tecnologicamente, a verdade é que os consumidores passaram a preferir o sistema VHS, pelo que, certas empresas, tiveram prejuízos que, com certeza, não querem repetir.

Os grupos envolvidos na tentativa de especificação dos próprios sistemas têm vindo a juntar-se para tentar definir um *standard*. Com certeza não se chegará a um *standard* único e universal porque os grupos europeus, norte-americanos e asiáticos estão a trabalhar em sentidos distintos.

A Figura 4 mostra o desenvolvimento dos vários projectos ao longo dos últimos anos, começando pelo X-10, pioneiro em “*Home Automation*”, termo frequente nos E.U.A. para a domótica, passando pelo EIB, pelos projectos ESPRIT iniciais e continuando com os projectos mais recentes, continuação dos anteriores ou novos projectos. No subcapítulo seguinte, abordaremos o estado actual dos vários projectos limitando-nos, nas secções seguintes, a efectuar uma análise sucinta dos principais candidatos a *standard*.

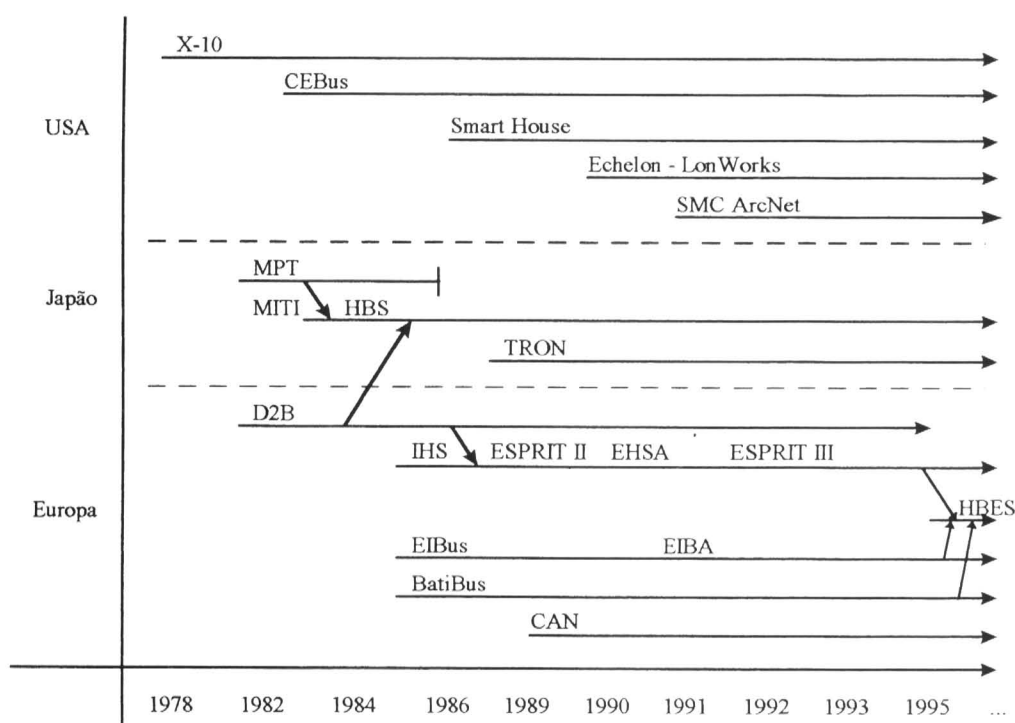


Figura 4 - Evolução dos sistemas domóticos.

2.2.1 Projectos Europeus

- **ESPRIT II (EHSA - European Home Systems Association)**

O projecto ESPRIT II, Home (EP2431), surgiu como tentativa da Comunidade Europeia para unir a indústria na criação de um verdadeiro *standard* europeu, capaz de competir com os *standards* americanos. Deste esforço resultou a especificação de um *standard*, o HSS - *Home Systems Specification*, também conhecido como EHS - *European Home System*, actualmente mantido pela EHSA, situado ao nível dos projectos de normalização americanos mais importantes como o CEBus e o LonWorks.

Esta especificação pode ser implementada em diferentes meios: rede eléctrica, par entrançado, cabo coaxial, infra-vermelhos e RF.

Neste projecto, e nos que se lhe seguiram, estiveram envolvidas mais de 50 organizações, entre elas parceiros industriais como a ABB, a AEG, a Philips, a Siemens, a SGS-Thompson, a Thorn, etc.

- **EIBus**

O EIBus - *European Installation Bus*, surgiu pela necessidade de flexibilidade, facilidade de instalação e baixo consumo de energia dos sistemas de gestão de edifícios. Actualmente é gerido pela EIBA (*European Installation Bus Association*) constituída por cerca de 80% dos fabricantes europeus de material eléctrico. Este *bus* está numa fase de implantação no mercado e tem sido bastante utilizado e fomentado pela Siemens e a ABB.

O EIB define a utilização de cablagem própria, par entrançado, para a transmissão de dados e para a alimentação dos dispositivos.

A principal desvantagem deste *bus* reside no facto de necessitar de cablagem própria para a sua instalação, por não permitir a utilização da rede eléctrica (recentes informações indicam a que vai ser possível essa utilização) como via de comunicação, e por os seus produtos apresentarem, ainda, um custo muito elevado.

- **BatiBus**

O BatiBus é um protocolo especialmente desenvolvido para edifícios. A sua implementação baseia-se, fundamentalmente, na utilização de módulos produzidos pelo grupo *Shneider*, pelo que a sua generalização se encontra limitada. No entanto, como abordaremos mais à frente, o BCI (*BatiBus Club International*), responsável pelo BatiBus, colabora no sentido do desenvolvimento de um *standard* europeu.

Uma das desvantagens do BatiBus é a de não permitir a comunicação através da rede eléctrica. Outras são a de ser um sistema centralizado e a de possuir pouca aplicação no

mercado habitacional.

- **CAN**

O CAN -*Controller Area Network*, surgiu originalmente para o mercado automóvel, para efectuar a comunicação entre os vários componentes de controlo do veículo. As suas características, principalmente a simplicidade do protocolo de comunicações, o baixo custo e a elevada fiabilidade, mostraram-se adequadas a mercados como o dos sistemas domóticos.

As principais vantagens deste sistema residem na fiabilidade de transmissão de dados de alto débito, baixo custo e variedade de *chips* disponíveis. As principais desvantagens deste sistema são a pouca utilização na área da domótica que advém da ausência de produtos no mercado vocacionados para este domínio e também o de não suportar a comunicação sobre a rede eléctrica.

2.2.2 Projectos Americanos (EUA)

- **X-10**

O X-10, tal como os seus sucessores, desenvolvido pela *Pico Electronics Ltd*, uma empresa escocesa, nasce da ideia de desenvolver um sistema que permitisse ao utilizador acender as luzes e ligar outros equipamentos remotamente sem haver a necessidade de instalar uma nova cablagem na habitação [TEAM97].

A transmissão de dados é efectuada através da rede eléctrica de 60Hz/50Hz usando a introdução de um sinal modulado de 120KHz aquando da passagem por zero da tensão da rede. A presença deste sinal equivale a 1 e a sua ausência a 0.

Apesar de ser algo limitado, o X-10 apresenta algumas vantagens. Talvez a mais significativa seja o baixo custo dos componentes. Interruptores de luzes ou controlo remoto podem custar algo como 10 ou 15 dólares EUA, respectivamente. Outra, é a facilidade de implementação. Qualquer utilizador compra um dispositivo novo e efectua a sua instalação sem grande dificuldade devido às suas características bastante próximas do *plug and play*. O utilizador pode, por isso, ir efectuando a evolução do seu sistema, introduzindo novos e mais complexos dispositivos, evitando assim um grande investimento inicial.

Aconselha-se, para mais informações sobre o X-10, consultar a sempre útil página das perguntas mais frequentes sobre o assunto (também válido para os outros projectos): <http://www.homation.com/x10faq/>

- **O CEBus**

Em 1984, membros da EIA (*Electronics Industries Association*) identificam a necessidade de desenvolver um *standard* que incluía mais que simples dispositivos. Engenheiros de empresas internacionais reuniram regularmente nos seis anos que se seguiram e desenvolveram a proposta de *standard* CEBus - *Consumer Electronic Bus*, que viria a tornar-se definitiva em 1992.

A especificação CEBus define o protocolo de comunicação entre vários dispositivos sobre rede eléctrica, par entrançado, cabo coaxial, infra-vermelhos, RF e fibra óptica. Por possuir uma arquitectura aberta, qualquer pessoa pode ter uma cópia da respectiva documentação e desenvolver produtos que funcionem de acordo com esta norma.

Futuros desenvolvimentos do *standard* CEBus são agora conduzidos pela CIC (*CEBus Industry Council*), uma organização com fins não lucrativos que reúne empresas como a Microsoft, IBM, Compaq, AT&T, Honeywell, Panasonic, Sony, Thomson, etc. Qualquer empresa pode fazer parte da CIC através do pagamento de uma cota anual entre 300 e 10,000 dólares EUA, além de outros requisitos.

- **SmartHouse**

Em 1987, a NAHB, Associação dos Construtores Cívicos dos EUA, desenvolveu um conceito para controlo de habitação que evoluiu para o sistema comercial conhecido por SmartHouse. Os direitos tecnológicos da SmartHouse foram vendidos e são agora pertença da *Amp Incorporated* e da *Molex Incorporated* [TEAM97].

O SmartHouse foi originalmente proposto como *standard* que incluía inteligência centralizada e novas cablagens e *hardware* para construções de raiz. Mais recentemente, começaram a trabalhar em conjunto com a CEBus e a LonWorks no sentido de desenvolver componentes de *hardware* para as casas existentes.

O SmartHouse apresenta uma tecnologia proprietária e orientada para o distribuidor. A instalação e o serviço de manutenção requer treino, certificação, ferramentas e *hardware* especiais.

- **Echelon - LonWorks**

LonWorks é o nome do *standard* proposto pela Echelon Corporation, empresa fundada em 1990 [TEAM97].

O principal componente do LonWorks é o circuito integrado “Neuron”, produzido sob licença pela Motorola e Toshiba, que inclui todo o *hardware* e protocolo de comunicações necessários à comunicação entre os vários dispositivos de uma rede. O *chip* Neuron pode ser instalado em qualquer produto e comunica através de rede eléctrica, par entrançado, cabo coaxial, infra-vermelhos, RF e fibra óptica.

A Echelon tem uma estratégia de mercado agressiva, com um elevado grau de disciplina no desenvolvimento dos seus produtos. A sua estratégia inicial consistia em analisar quais as áreas de negócio em que o Neuron podia ser implementado, por forma a estabelecer uma rede de comunicações entre as várias máquinas numa indústria ou edifícios comerciais. O LonWorks é usado pela na automação, gestão de energia, controlo de cargas e leitura remota de dados.

- **SMC ArcNet**

A SMC propõe uma arquitectura baseada na disponibilização do protocolo ArcNet em circuitos integrados [Baptista97].

Sendo o controlo da rede efectuado pelo microprocessador, facilmente integrado como periférico, existe uma elevada independência dos produtores de sistemas domóticos.

A transmissão de dados pode ser efectuada sobre a rede eléctrica, par entrançado, cabo coaxial, infra-vermelhos, RF e fibra óptica.

A principal vantagem desta abordagem consiste no uso de um protocolo de rede já *standard* (o ArcNet), com custos de desenvolvimento extremamente baixos e sem custos adicionais de licenças.

2.2.2 Projectos Asiáticos (Japão)

- **TRON**

O projecto TRON - *The Real-Time Operating System Nucleus*, foi iniciado em 1984, na Universidade de Tóquio.

O objectivo a longo prazo do TRON é o de desenvolver um sistema distribuído altamente funcional em que cada objecto computacional, objecto inteligente, possa cooperar com outros objectos. O projecto inicial foi dividido em vários subprojectos, cada um tratando de definir uma área específica. O CHIP, que pretende especificar a arquitectura do microprocessador, o ITRON, que define as especificações do sistema operativo multi-tarefa em tempo-real, o BTRON, que propõe as especificações do sistema operativo para uso em PCs e *Workstations*, o CTRON, que pretende definir as especificações da interface do sistema operativo para a comunicação e o processamento da informação e o MTRON, que vai definir a arquitectura do sistema operativo.

Uma das opções chave do projecto TRON é a de permitir que os resultados do projecto, as especificações publicadas, sejam do domínio público, permitindo assim, que fabricantes de todo o mundo possam desenvolver produtos baseados nessas especificações.

Podemos concluir, da análise histórica feita aos vários projectos, que os EUA possuíram inicialmente um largo avanço no desenvolvimento de sistemas domóticos já que, ao contrário dos europeus e asiáticos, tentaram encontrar e fornecer propostas que vão ao encontro do mercado e do utilizador comum, mesmo não apresentando os seus produtos todas as potencialidades que se poderiam implementar num sistema domótico.

Os europeus e asiáticos, tentaram criar soluções “perfeitas” e depararam-se com as dificuldades inerentes a uma normalização. Só mais recentemente começaram a sair desse impasse e a desenvolver produtos para utilização no dia a dia.

2.3 Estado Actual da Tecnologia

A instalação de novas estruturas de comunicação baseadas na fibra óptica tem contribuído para a qualidade dos produtos destinados, principalmente, ao teletrabalho e ao entretenimento (TV interactiva, *Video On Demand*, Internet, etc.). Os sistemas domóticos implantados de raiz em habitações novas usam, quase sempre, um meio de transmissão em par entrançado ou cabo coaxial, optando-se pela transmissão na rede eléctrica ou pela rádio frequência, menos habitual, quando pretendemos instalar um sistema numa habitação já construída.

[Charm96] aconselha a instalação de diversos tipos de cablagens conforme a necessidade dos vários sistemas. A Tabela 2 apresenta essas recomendações.

Tabela 2 - Recomendações para instalação de cablagem.

	Tipo de Cablagem	Quantidade	Utilização
Mínimo	Cabo <i>standard</i> para telefone	3 ou 4 pares	Telefone, Fax, Comunicações de baixa-velocidade.
“	Cabo par entrançado de alta velocidade	4 pares	Computadores, Dados, Controlo.
“	Cabo Coaxial	RG/6	Vídeo, Comunicações de alta velocidade.
Opcional	Cabo par entrançado de alta velocidade	4 pares	Dados e controlo adicionais.
“	Cabo Coaxial	RG/6	Distribuição Local <i>Duplex</i> de Vídeo
“	Cabo de Fibra Óptica	1	Serviços futuros (TV interactiva)

Nos Estados Unidos, o sistema X-10 é considerado actualmente um *standard*. O X-10 define as especificações da transmissão na rede eléctrica e é usado especialmente para

habitações já existentes, pelo que se tornou um sistema bastante utilizado. O sistema X-10 encontra-se instalado em, aproximadamente, 4 milhões de lares e é comercializado por empresas de elevada reputação como a Radio Shack, a Stanley, a Leviton e a Honeywell. Existe um sem número de produtos que usa a tecnologia X-10. Desde interruptores, sistemas de segurança, interface com televisão ou computador, atendedores de chamadas, controlo remoto, etc. Uma das limitações do X-10, que se mantém desde o início, é o número de endereços suportados, de 256, o que é limitado para os vários produtos. Existe a possibilidade do aumento do número de endereços somente para um reduzido número de produtos especiais. Outra é o facto de quase todos os dispositivos X-10 comunicarem só num sentido. Se uma luz for acendida com o interruptor de parede tal facto não pode ser comunicado ao sistema.

O CEBus é um *standard* que utiliza endereços que são definidos de fábrica no *hardware* e incluem 4 biliões de possibilidades. O *standard* oferece, também, uma linguagem *object oriented* com muitos objectos de controlo [TEAM97].

Actualmente, todo o *hardware* de comunicações, linguagem e protocolo está disponível num *chip* da Intellon Corporation. A Intellon vende o *chip* a outros fabricantes para o instalarem nos seus produtos e fornece, também, rotulagem própria e produtos OEM (*Original Equipment Manufacturer*) que usam o *standard* CEBus. A Intellon vende ainda “Kits de desenvolvimento” a quem pretenda desenvolver produtos compatíveis CEBus.

A CIC, organização que gere o CEBus, tem, em fase de desenvolvimento, um “laboratório de testes” que permite efectuar a certificação dos produtos. Os fabricantes são assim encorajados a usar este laboratório para verificar a conformidade dos seus produtos no ambiente de rede da habitação. A certificação permite que o fabricante possa passar a usar o logotipo da CEBus nos seus produtos.

A CIC apresentou no início do ano de 1997, na CES-Habitech, a sua nova especificação *Home Plug and Play* (HPnP). Esta especificação usa o *standard* CEBus. Os produtos *Plug and Play*, como o nome indica, só necessitam de ser ligados à rede da habitação para ficarem operacionais, não precisando de configuração por parte do utilizador. Os próprios dispositivos *PnP* efectuem a comunicação com os outros dispositivos na rede para se apresentarem e auto-configurarem de acordo com os critérios do sistema.

A LonWorks que, como referimos antes, usa o *chip* Neuron, tornou possível a implementação do seu protocolo, sob licença, em qualquer tipo de processador, desde Junho de 1997. A Adept Systems Inc. vai efectuar a implementação do protocolo para o Motorola 68360, um processador de 32 *bits*, em C, e o resultado desse trabalho poderá ser consultado brevemente através da página da Echelon na WWW. A Echelon comercializa os produtos *LonWorks*, como *routers*, módulos, *transceivers* e ferramentas de rede e de desenvolvimento que usam o *chip* Neuron.

A Echelon criou, em 1994, um “grupo do consumidor” para analisar as perspectivas de consumo dos diferentes produtos e sistemas para a habitação. Em Agosto de 1997 a EIA,

através do seu comité - *Integrated Home Systems Technical Committee* - formou um sub-comité para considerar a hipótese de desenvolvimento de um *standard* EIA adicional para redes habitacionais baseado no LonWorks (EIA/IS-709).

Cada um dos grupos referidos no capítulo anterior tem vindo a desenvolver actividades no sentido de melhorar a especificação dos seus *standards*. Tem-se assistido a algumas alianças por partes dos grupos europeus no sentido do desenvolvimento de um *standard* único.

A *Home Systems Specification*, originada pelo projecto ESPRIT Home (EP2431), é uma especificação muito complexa, baseada no modelo OSI, e destinada a especificar uma rede a usar em habitações unifamiliares. Por apresentar uma arquitectura muito pesada tem sido de difícil implementação pelos fabricantes e projectistas que, geralmente, optam por modelos mais simples.

Dos sistemas europeus actuais, o EIBus é o que potencialmente mostra capacidade para se transformar num verdadeiro *standard*. A Siemens tem vindo a investir fortemente no seu desenvolvimento e a sua implantação no mercado tem vindo a aumentar. Podemos afirmar que este sistema se destina, principalmente, a edifícios de escritórios ou similares, sendo implantado de raiz, já que necessita de cablagem própria. Ultimamente tem vindo a ser também utilizado para edifícios de habitação e alguns projectos deste tipo já foram implementados em alguns países. Em Portugal a Siemens tem feito várias instalações em edifícios de escritórios.

O *BatiBus Club International*, a *European Home Systems Association* e a *European Installation Bus Association* juntaram-se, em 1996, para criar um protocolo único para as redes de comunicação das habitações. Este *standard*, baseado numa série de soluções consistentes de comunicações para diferentes meios, permitirá que qualquer fabricante de produtos possa utilizar um microprocessador ou microcontrolador *standards* disponíveis no mercado.

Recentemente desenvolvido pela Bosch-Siemens, o HES - *Home Electronic System*, apresentado em 1996 na CEBIT, surge como uma plataforma de *software* aberta e modular para sistemas de gestão de habitações. É baseado no sistema EIB e permite ao utilizador operar, controlar e monitorizar, de um ponto central, todos os componentes interligados no sistema. Os componentes ficam, desta forma, visíveis e acessíveis ao utilizador. O núcleo do sistema é o *Home Assistant*, um *software* de gestão com uma interface gráfica com o utilizador bastante bem desenvolvida, a correr sobre um PC multimédia *standard*. Na interface bastante amigável e de simples utilização, os componentes são representados por símbolos gráficos. A actuação é efectuada por toque directo no monitor ou pela acção do rato e teclado. Graças à sua capacidade multimédia, o sistema actua também como uma ponte entre as funções do interior da casa e a rede exterior [Nonnen97].

O HES, sendo um sistema aberto, fomenta a cooperação com todos os fabricantes de produtos e sistemas para a habitação. Fabricantes de componentes *standard* EIB têm a hipótese de integrar os seus próprios produtos e aplicações no *software* HES e assim

comunicar, controlar e administrar os seus produtos via plataforma HES.

A modularidade da estrutura do HES, representada na Figura 5, torna-o mais adaptável às necessidades do consumidor. O utilizador não tem que adquirir o pacote completo de *software*, mas apenas os módulos apropriados às suas necessidades específicas.

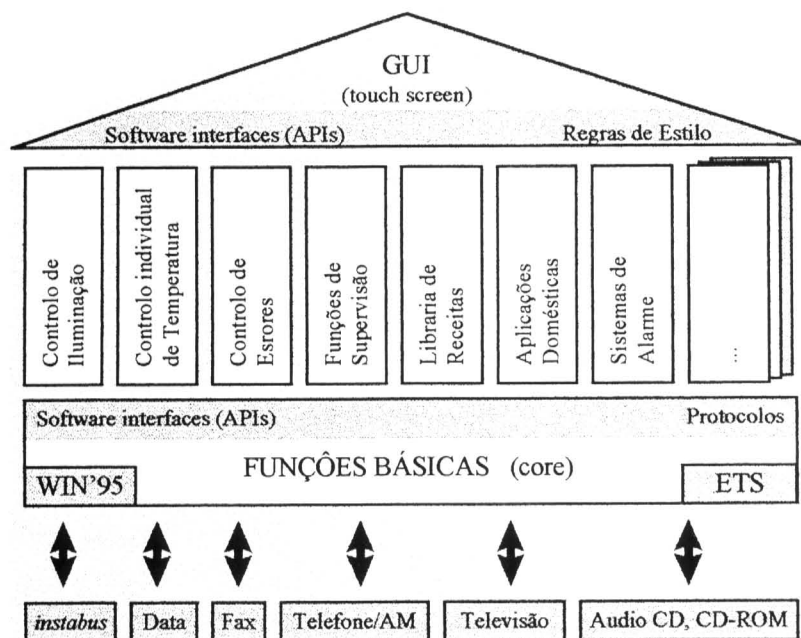


Figura 5 - A estrutura do HES.

O HES torna-se um elemento importante para a disseminação e sucesso dos sistemas domóticos.

O grupo responsável pelo HES, a BSHG (*Bosch-Siemens Hausgeräte GmbH*), anunciou para 1997, a introdução no mercado de produtos da linha branca (*white goods*).

2.3.1 Sistemas domóticos e produtos

Mais do que citar todos os produtos existentes no mercado ou listar os inúmeros sistemas domóticos interessa, neste subcapítulo, efectuar algumas referências a locais e publicações onde se pode obter informação sobre a área da domótica e dos edificios inteligentes.

Relativamente a exposições e conferências são de referir as realizadas nos Estados Unidos, a HABITECH, integrada no CES - *Consumers Electronics Show*, as exposições e conferências da IHA - *International Home Automation* ou da CABA (*Canadian Automated Buildings Association*), os seminários realizados pelos responsáveis de cada um dos projectos e os realizados na Europa, a Euro Domotique, em França e a Domotechnica, na Alemanha ou conferências internacionais como a última realizada em Israel.

As revistas especializadas são muitas. Podemos citar a *Domotique Magazine* e a *Domotique News*, francesas, a *Home Systems Installer*, a *Home Automation Pro*, a *Electronic House*, a *Home Electronics*, a *Popular Home Automation* e a recente *Home Automator*, americanas, e a *InfoDomotique*, canadiana. Existem também outras publicações sobre domótica, principalmente em França e nos Estados Unidos, actas das conferências realizadas e ainda documentos electrónicos.

Quanto aos produtos, além de os podermos encontrar nas revistas especializadas e nas empresas que os comercializam, uma pesquisa pela Internet pode fornecer-nos uma infinidade de elementos sobre os vários sistemas e dispositivos disponíveis no mercado nacional e estrangeiro. O mercado português tem ainda poucas empresas a trabalhar nesta área, mas o seu número aumenta de ano para ano. Existiu também uma tentativa de publicar uma revista portuguesa sobre a domótica (Casa do Futuro), chegando mesmo a sair o primeiro número em Abril de 1994, mas que não teve continuidade.

A título de exemplo podemos citar alguns sistemas, interfaces e casas de demonstração existentes nos vários países:

- ***Sistemas Domóticos***

- Synforic (Francês)
- Domogyr (Suíço)
- Domobox (Francês)
- Isis (Francês)
- Samantha (EUA)
- Honeywell TotalHome (EUA)

- ***Interfaces com o utilizador***

- Computador pessoal
- Telecomandos
- Terminal de videotexto (Minitel)
- Quadros sinópticos de parede

- ***Casas de demonstração***

- La Casa Domotica (Espanha)
- Maison du Futur (França)

- Maison Performante (Canadá)
- Huis Van de Toekomst (Holanda)
- Xanadu (EUA)
- SmartHouse (EUA)
- Honeywell House (EUA)
- Tron -Concept Intelligent House (Japão)

Como referimos atrás, uma pesquisa na Internet pode orientar-nos na escolha dos equipamentos ou dos fabricantes e vendedores de um sistema domótico. Existem, também, páginas com apontadores para os vários pontos de interesse sobre o tema, não só produtos, equipamentos e empresas, mas também explicações sobre os vários projectos, as próximas conferências, a situação actual, etc. Existe um *newsgroup* dedicado ao tema da Domótica com endereço `comp.home.automation`.

Existe ainda a possibilidade de usarmos um agente electrónico de pesquisa de informação para coleccionar e seleccionar páginas com interesse nesta ou outra área. Um dos agentes, situado num servidor de uma instituição universitária americana, The Informant, permite que nos inscrevamos gratuitamente nos seus serviços. Este agente efectua a pesquisa da informação requisitada pelo utilizador evitando, assim, a necessidade de ligação contínua à Internet para obter informação sobre um determinado assunto. O endereço do The Informant é (<http://informant.dartmouth.edu>).

Alguns locais de interesse na Internet são apresentados no anexo B, resultado da compilação efectuada pelo The Informant para as palavras chave indicadas.

Uma vez apresentada uma caracterização sumária dos sistemas domóticos, serão introduzidas no próximo capítulo as questões principais relacionadas com a interface com o utilizador.

Capítulo 3

A Interface com o Utilizador

A interface homem-máquina existe desde há muito tempo. Desde que existem ferramentas de apoio ao trabalho que se pode dizer que existem interfaces para efectuar a comunicação entre essas ferramentas e o homem. Brenda Laurel define a interface como a zona onde se estabelece o contacto entre duas entidades. Uma interface é uma superfície de contacto, as funções que podem ser realizadas e o balanço entre o poder e o controlo [Laurel90]. Podemos mesmo considerar, por exemplo, como interface, a maçaneta de uma porta ou o acelerador e volante de um automóvel. A maçaneta estabelece a ligação entre a possibilidade da porta estar aberta ou fechada e o desejo do utilizador de efectuar qualquer uma das operações possíveis. Convém também referir que, na maior parte das vezes, a própria interface reflecte quem é o agente e quem é o paciente. No caso da porta, a forma da maçaneta indica-nos que o agente somos nós. No entanto, nos casos das portas automáticas, a impressão que temos quando nos aproximamos delas é que o controlo da interface está do outro lado e que nós passamos a ser o paciente.

A “interface com o utilizador”, aplicada ao mundo informático, é uma expressão que surgiu por volta dos anos setenta na área da engenharia. Na altura, quem interagia com os computadores eram, sobretudo, programadores e engenheiros. Desde que o computador passou a ser um instrumento de massas, sendo usado em qualquer instituição,

departamento, universidade, escritório e, mais recentemente, em casa, o conceito de utilizador tipo alterou-se substancialmente. Agora, os utilizadores tipo passam a ser os comuns cidadãos que usam os computadores nos seus empregos, em casa e até nas horas livres. Devido a esta alteração, a evolução da interface com o utilizador fez-se de uma forma bastante rápida, por forma a satisfazer as novas necessidades impostas pelos novos utilizadores. Podemos ser levados a pensar que a massificação do uso do computador se deveu especialmente às novas interfaces com o utilizador, mas não devemos esquecer que a primeira “pequena massificação” se deu nos serviços, na altura em que ainda se usavam terminais e deveu-se, basicamente, ao facto de se ter constatado que o computador simplificava um determinado número de tarefas, poupando um tempo considerável. E tempo é dinheiro, logo, apesar de ser necessário um investimento inicial elevado, a amortização fazia-se facilmente porque os novos métodos de trabalho eram mais produtivos.

O principal objectivo, a partir de então, passou a ser o de adaptar o computador ao homem e não o que acontecia anteriormente, o homem ter de se adaptar ao computador. Nesse sentido, o caminho seguido foi o de procurar desenvolver interfaces com o utilizador que fossem usadas com muita facilidade e com aprendizagem e memorização rápidas. Assim surgiu o conceito de *usability* (que traduzo como usabilidade; o quanto um sistema é mais ou menos usável) que tem sido um dos factores que tem levado à massificação antes abordada. A usabilidade está relacionada com a eficácia e eficiência da interface e com a reacção dos utilizadores à mesma. Os custos de uma baixa usabilidade podem ser medidos em termos de custos de treino e perda de produtividade.

Neste capítulo são abordados os aspectos relacionados com a evolução das interfaces com o utilizador, a situação actual e os desenvolvimentos nesta área e, por fim, efectua-se uma explanação sobre as técnicas de apresentação da informação em interfaces gráficas, que constitui uma das partes principais deste trabalho.

3.1 A Evolução da Interface com o Utilizador

Um dos primeiros passos realizados desde a construção dos primeiros sistemas computacionais foi o de tentar reduzir o seu tamanho a um volume razoável. Os primeiros sistemas, com um poder de cálculo reduzidíssimo, ocupavam uma sala de grandes dimensões. Depois dos primeiros avanços no sentido de maior capacidade de cálculo e menor volume, o passo seguinte, derivado da constatação que uma máquina deste tipo podia ser empregue para um grande número de tarefas repetitivas, foi o de tentar melhorar a interface com o utilizador. Os primeiros sistemas a serem usados por um grande número de pessoas foram os terminais TTY (*Teletype*), inicialmente com impressão em papel, mas que rapidamente evoluíram para dispositivos Glass TTY, nos quais a informação era mostrada em *displays* visuais. Este tipo de dispositivo apresenta uma interface baseada em *prompts* e comandos.

Um tipo de interface baseado em *prompts*, em que a aplicação faz perguntas que devem ser respondidas pelo utilizador, é fácil de desenvolver, de usar, e pode ser usado para pequenas aplicações, apresentando, no entanto, sérias limitações. Além disso verifica-se que o controlo da acção se encontra do lado do sistema, sendo o utilizador relegado para segundo plano. Este tipo de interface torna-se repetitivo e rapidamente provoca no utilizador uma sensação de rotina.

No sentido de inverter esta situação avança-se para as interfaces baseadas em comandos. Este tipo de interfaces é, também, fácil de desenvolver, bastante flexível e poderoso, mas apresenta alguns problemas inerentes às suas limitações.

- Requer entrada pelo teclado dos vários comandos, o que proporciona a existência de erros na introdução.
- Torna-se bastante fastidiosa. Mesmo para realizar uma tarefa simples o utilizador pode ter que introduzir comandos longos.
- Necessita de memorização. O utilizador tem que ter sempre presente um grande número de comandos válidos o que, para um utilizador inexperiente, pode ser extremamente desagradável.
- Estimula a acção orientada ao comando “verbo-objecto”. Este tipo de comandos funciona da seguinte forma: primeiro escolhe-se o tipo de acção e em seguida o objecto sobre o qual ela deve incidir (apagar - palavra).

Quando se introduziram os terminais CRTs (*Cathode Ray Tube*) nos sistemas computacionais verificou-se uma revolução extremamente importante nas interfaces com o utilizador. A tecnologia deste tipo de *display* forneceu aos investigadores os meios necessários para a substituição das interfaces baseadas em comandos pelas interfaces baseadas em menus, estas já mais familiares.

As interfaces baseadas em menus resolvem alguns dos problemas anteriores, como a entrada pelo teclado e a necessidade de memorização, já que os comandos disponíveis se encontram organizados em menus. Assim o utilizador quando precisa de fazer um determinado comando só tem que abrir o menu correspondente ao tipo de comando necessário e seleccioná-lo da lista de comandos apresentados. No entanto, as interfaces baseadas em menus, por si só, não resolvem os problemas da necessidade de memorização do estado do sistema e da interface baseada em comandos. O problema de recordar o estado do sistema é resolvido apresentando ao utilizador o estado corrente através da técnica muito comum nos dias de hoje, conhecida por WYSIWIG (*what you see is what you get*). Os problemas derivados da interface baseada em comandos resolvem-se através do uso de **manipulação directa** combinada com a interface WIMP (*Windows, Icons, Mouse, Pull-down menus*).

3.1.1 A interface WYSIWIG

A interface do tipo WYSIWIG, como o próprio nome indica, caracteriza-se por apresentar ao utilizador uma imagem real do que pode obter num determinado estado do sistema, isto é, não existe informação não visível ou escondida. Este tipo de interface é bastante utilizado em aplicações gráficas interactivas em que existe a necessidade do utilizador controlar visualmente o estado final da apresentação. A representação no ecrã é muito semelhante ao resultado final. Se a interface não for do tipo WYSIWIG, o utilizador tem que efectuar a relação entre a imagem mental do resultado pretendido e os códigos necessários para a sua obtenção. Um bom exemplo destas interfaces são os antigos processadores de texto em que a formatação dos caracteres era obtida através de teclas de controlo e a informação visual da formatação era dada através de mudança de cor.

Exemplo

- Interface WYSIWIG - Isto seria um **texto** visto neste tipo de *interface*
- Interface não WYSIWIG - Isto seria um texto visto neste tipo de interface

3.1.2 Manipulação directa

Uma interface gráfica de manipulação directa apresenta um estilo de interacção em que os objectos, atributos ou relações que podem ser manipulados são representados de uma forma visual; as acções são realizadas sobre representações visuais das operações pretendidas, tipicamente através da utilização do rato, em vez de se efectuar explicitamente a descrição da acção a realizar. Os comandos, neste tipo de interface, estão implícitos na representação visual, que pode ser texto, como o nome de um objecto, ou uma imagem gráfica, como um ícone. Podem ainda existir comandos derivados de acções que simulam a realidade, como o arrastar um ficheiro para “dentro” de uma pasta com intenção de o mudar de sítio ou o copiar um ficheiro de um disco para outro.

A interface de manipulação directa é largamente utilizada e é reconhecidamente o melhor tipo de interface para o utilizador comum. É muito poderosa e particularmente fácil de aprender. O número de erros cometidos pelo utilizador baixa consideravelmente quando comparado com outro tipo de interfaces.

No entanto, para um utilizador experiente, este tipo de interface pode tornar-se bastante lento quando comparado com a interface de comandos. A situação ideal será conseguir combinar os vários tipos de interface de forma a providenciar vários tipos de interacção de acordo com as necessidades dos utilizadores e a sua experiência. Será aconselhável fornecer a possibilidade de o utilizador experiente escolher, por exemplo, teclas de acesso rápido, para os seus comandos mais frequentes, diminuindo assim o tempo gasto na selecção do

comando desejado. A interface WYSIWIG pode e deve, também, ser combinada com a manipulação directa formando assim, muito provavelmente, um estilo de interacção poderoso, fácil de aprender e fácil de usar. Exemplos desta junção de tipos de interfaces encontram-se em qualquer aplicação actual.

Na Figura 6 mostram-se as várias etapas do desenvolvimento das principais tecnologias empregues nas diferentes componentes da interface com o utilizador, desde os aspectos ligados à investigação até à sua comercialização [Myers96].

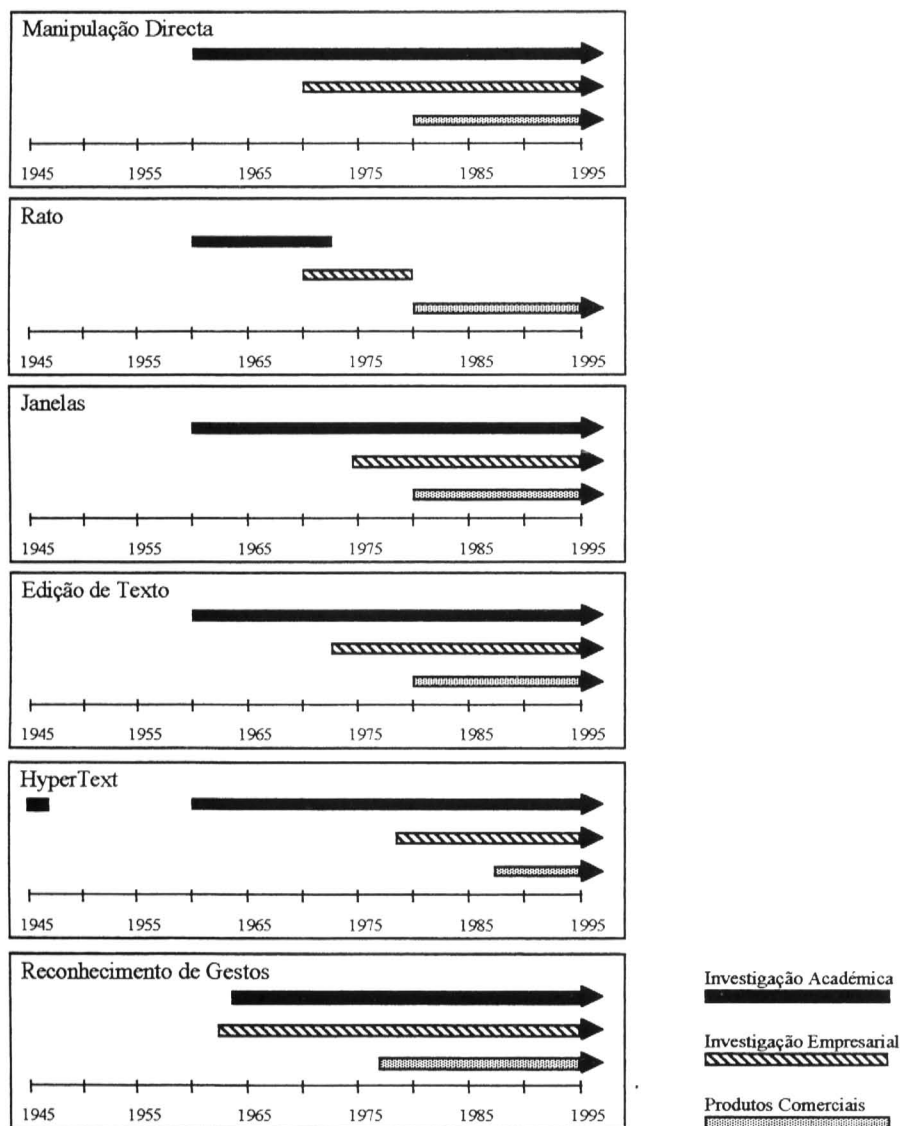


Figura 6 - Etapas temporais, aproximadas, do trabalho realizado nas principais tecnologias aplicadas à interface com o utilizador.

Convém ainda referir que um dos factores quase sempre esquecidos quando se aborda o tema das interfaces gráficas com o utilizador são as características e o tamanho dos monitores usados nos computadores. Se as características mais aconselhadas para os

monitores são aquelas que determinam um melhor conforto, uma melhor visão, cores mais brilhantes, melhor definição, um ponto mais pequeno, o ecrã plano, já quanto ao tamanho, alguns cuidados devem ser prestados na escolha do monitor. O dizer “quanto maior melhor” apresenta alguns pontos de reflexão. O tamanho do monitor deve ser o necessário para as funções que vamos realizar. De pouco nos serve um grande monitor para, por exemplo, o utilizar no controlo de um sistema domótico. Um sistema deste tipo não apresenta elementos que necessitem de um grande espaço de apresentação, como sucede com os projectos de arquitectura ou as construções mecânicas que se situam na área do CAD. O investimento num monitor de tamanho superior ao necessário pode revelar-se inútil. A própria aplicação pode apresentar um aspecto pouco cuidado quando utilizamos um monitor de grandes dimensões. Os objectos podem aparecer bastante afastados uns dos outros o que contraria algumas das regras básicas da apresentação da informação, descritas num subcapítulo posterior.

Quando são usados dispositivos de apresentação da informação bastante grandes, com dimensões da ordem do metro, convém ter em atenção regras próprias de apresentação, já que elas são diferentes, em alguns pontos, das utilizadas nos monitores. [Swaminathan97] apresenta um estudo sobre a interacção em grandes *displays*.

Actualmente tem-se vindo a assistir a grandes avanços em técnicas de interface com o utilizador para áreas específicas, desde o reconhecimento de gestos, sistemas multimédia, sistemas 3-D, realidade virtual e “realidade aumentada”, trabalho cooperativo com suporte computacional, linguagem natural e reconhecimento de voz [Myers96].

3.2 Perspectivas Futuras

Há já alguns anos que as interfaces de manipulação directa combinadas com interfaces do tipo WYSIWIG e WIMP são usadas nas aplicações mais variadas e de uma forma generalizada. Os sistemas mais recentes passaram de uma filosofia orientada para a aplicação para uma filosofia orientada para o documento. O documento passou a ser o centro da atenção e os componentes que lhe poderiam ser ligados seriam fornecidos pelas várias aplicações que passaram a ser periféricas ao documento. Esta abordagem acompanhou a necessidade sentida pelo utilizador quando elaborava um determinado documento e verificava que tinha que saltar de aplicação para aplicação para ir acrescentando novos elementos ao seu trabalho. A tecnologia OLE veio facilitar a possibilidade de utilizar elementos de várias aplicações num mesmo documento e de ligar informações entre documentos.

Actualmente a sociedade tecnológica encontra-se numa nova fase de evolução como consequência da massificação dos recursos computacionais e da globalização da tecnologia. Estamos na era da informação. Todos os sistemas actuais avançam no sentido de permitir ao utilizador a obtenção da informação que necessita no mais curto espaço de tempo

possível. Já não faz sentido trabalhar isoladamente e investir tempo e recursos económicos para desenvolver um trabalho que já foi feito por alguém, noutra parte do mundo. O precioso tempo e dinheiro podem ser utilizados para a investigação e para o aperfeiçoamento dos sistemas existentes. O acesso à informação tornou-se, assim, o ponto fulcral da sociedade actual. As novas interfaces com o utilizador não podem deixar de acompanhar este avanço e já se conhecem novos desenvolvimentos por parte de empresas conceituadas, como a Apple, Microsoft, Netscape, Oracle, Sun, IBM, Lotus e outras menos conhecidas, por forma a realizar interfaces orientadas à informação.

As novas interfaces por vezes *designadas* como NCGUI (*Network-Centric Graphical User Interface*), WUI (*Web User Interface*), virtual desktops, active desktops, Webtops ou NUI (*Network User Interface*), estão já a ser desenvolvidas e apresentam um aspecto de *browser* permitindo navegação local e remota de sistemas de ficheiros. Podem apresentar *applets* Java e outros componentes dinâmicos da Web sem a necessidade de um *browser*. Farão a actualização dinâmica dos componentes através de carregamento e descarregamento usando tecnologia Web. Desvanecerão, assim, a crescente irrelevância entre plataformas nativas/distribuídas e aplicações locais/remotas [Halfhill97].

Sendo maior o empenho das empresas em construir as novas interfaces para computadores para a rede (NCs - *Network Computers*), a vaga estende-se também aos computadores pessoais. A Microsoft está a introduzir no Memphis (Windows 97) e Windows NT a navegação de ficheiros tipo *browser* e a ligar o Internet Explorer 4.0 a fundo no sistema Windows. A IBM já adicionou características como *Java Virtual Machine* ao OS/2 Warp 4 e encontra-se a desenvolver uma NUI com nome de código "*Bluebird*" que oferecerá um compromisso entre clientes de redes de computadores e clientes remotos. A Apple está a introduzir o acesso Internet no seu sistema Mac OS/8, assim como publicação pessoal Web e outras características. A Netscape desenvolveu o *Netscape Constellation* que se sobrepõe à interface gráfica existente no computador pessoal.

É fácil perceber porque razão as GUI estão a perder terreno face às novas interfaces. A sua concepção inicial, como referido anteriormente, era tentar substituir as interfaces baseadas em comandos por interfaces visuais do sistema operativo. O seu objectivo era facilitar a gestão de alguns *megabytes* de recursos locais. Actualmente, o utilizador enfrenta um sistema de ficheiro virtual que se pode dizer global em termos regionais, e a quantidade de informação é da ordem dos *terabytes*. Torna-se necessário introduzir características de rede nos computadores pessoais.

Existem seis razões principais pelas quais se torna evidente a necessidade de algo novo [Halfhill97]:

- Conteúdo executável - O *software* de hoje já não é somente nativo e local. Pelo contrário, pode chegar a qualquer momento de um servidor situado a milhares de quilómetros de distância. Mais, a aplicação pode ser um *stock* de notícias ou outra informação que o sistema actualiza automaticamente pelo acesso a um servidor Web.

- Código - As NUI providenciam *gateways* para programas ou bases de dados que podem ser utilizados pelos programadores de forma a facilitar o desenvolvimento ou alteração de aplicações.
- Novos utilizadores - Muitos utilizadores recentes concentram o seu trabalho num número bastante reduzido de tarefas e não necessitam de uma GUI com todas as potencialidades, assim como o vulgar fotógrafo não necessita de uma Hasselblad (o Rolls Royce das máquinas fotográficas).
- Repensar a GUI - Os engenheiros de *software* estão a reconsiderar pressupostos antigos e a aplicar novos conhecimentos. Algumas das novas NUI abandonam funcionalidades que causam problemas aos novos utilizadores como o duplo *click*, as janelas sobrepostas, os menus hierárquicos e as barras de botões hieroglíficas.
- Acesso remoto - As NUI facilitarão a troca de correio electrónico aos trabalhadores não fixos (tele-trabalho) e a duplicação do ambiente de trabalho na sua localização remota.
- Alterar as capacidades de trabalho - As NUI permitem uma troca mais fácil de computadores entre utilizadores através do armazenamento do estado local num determinado servidor, o que possibilita ao utilizador o acesso ao seu espaço de trabalho personalizado onde quer que faça o *Logon*.

Qualquer das NUI tem o objectivo de tentar mostrar os recursos da rede de uma forma tão familiar como os recursos locais. Claro que, para já, não poderão esconder as limitações físicas impostas pela largura de banda da rede. Um servidor Web levará sempre mais tempo a aceder que um disco interno. Mas o que as NUI podem fazer é atenuar essas limitações tornando menos abruptas as diferenças. Aliás, já [Gray96] aponta os requisitos para as novas linguagens de especificação de interface com especial atenção para os atrasos das comunicações na *World Wide Web*.

Uma das formas de conseguir esse objectivo é utilizar uma apresentação gráfica universal para todos os recursos. Outra, é integrar as características de rede nas operações comuns no *desktop* em vez de obrigar o utilizador a usar aplicações específicas para efectuar tarefas que passam a ser de rotina. Num futuro próximo as NUI absorverão as funções dos vários programas necessários para uso das redes de computadores: *browsers*, *newsreaders*, FTP, *e-mail*, editores HTML, etc.

Na altura da elaboração deste trabalho, as NUI não são, ainda, do domínio público, esperando-se para breve a sua introdução no mercado informático. Informações adicionais sobre o estado das NUI pode ser encontrada nos endereços Web das empresas dedicadas ao seu desenvolvimento:

Apple - <http://www.apple.com>

IBM - <http://www.internet.ibm.com/computers/networkstation/>

Lotus - <http://kona.lotus.com>

Microsoft - http://www.microsoft.com/backoffice/sbc/sbc_summary.htm#top

Netscape - http://www.netscape.com/comprod/tech_preview/index.html

Oracle - <http://www.nc.com/>

The Santa Cruz Operation - <http://tarantella.sco.com/>

Sun/JavaSoft - <http://www.javasoft.com/>

TriTeal - <http://www.softnc.triteal.com/>

Ulysses Telemedia - <http://www.ulysses.net/>.

Investigações recentes têm introduzido um novo conceito de interfaces baseadas em agentes. Estes agentes serão responsáveis por efectuar um trabalho autónomo de acordo com as instruções predefinidas pelo utilizador e também consoante o tipo de utilização que a pessoa faz do seu sistema.

A interface baseada em agentes parte do pressuposto que, tanto o utilizador como o sistema, através do agente, podem iniciar a comunicação e tomar a iniciativa [Kay90]. Em [Cesta96] é apresentado um caso de estudo em que o objectivo é conseguir obter interfaces activas e cooperativas capazes de se auto-adaptar nas suas formas de interacção de acordo com a sua interpretação e percepção das necessidades do utilizador. A metáfora utilizada é a de um ou mais “assistentes pessoais” que apoiam o utilizador e colaboram com ele no mesmo ambiente de trabalho.

Estes agentes serão utilizados preferencialmente em tarefas repetitivas. Podem ser implementados agentes com vários fins: agentes de interface, agentes de correio electrónico, agentes de reuniões, de informação, de armazenagem, etc.

Devido ao desenvolvimento rápido e à generalização das redes de comunicações esta abordagem terá cada vez maior difusão pela sua capacidade de lidar com a sobrecarga de informação e com a complexidade crescente dos sistemas e recursos.

3.3 Apresentação da informação

3.3.1 Introdução

Esta secção é dedicada à análise das regras fundamentais que devem ser usadas para a apresentação da informação no monitor e nos restantes dispositivos de interface com o utilizador.

A primeira questão que se nos coloca é: o que é a informação? Segundo [Dubrau96] a informação é composta por informação que por sua vez é também composta por informação e assim sucessivamente. A situação é semelhante à matéria. Tentamos chegar à unidade de informação sem nunca conseguir. Para Dubrau a informação comporta-se de forma análoga à energia e tem as seguintes propriedades:

- Não tem leis de conservação
- Está em todo o sítio
- Pode ser criada
- Pode ser destruída
- Pode ser copiada
- Pode ser alterada: espaço - tempo, paralela - série.

Cada uma das leis não existe separadamente, elas estão todas interligadas; se não existem leis de conservação a informação pode ser destruída; copiando a informação altera-se a informação e alterando a informação destruímos e criamos nova informação porque a informação está em todo o lado.

Podem ser considerados três tipos de informação:

- Informação acerca do sistema técnico e sua funcionalidade.
- Informação acerca do estado do sistema.
- Informação acerca das possibilidades de interacção e indicação das acções a realizar.

Muitos autores, desde o movimento gestaltico, tentaram definir regras para a disposição da informação de forma a obter uma apresentação harmoniosa.

Nos subcapítulos que se seguem faz-se referência a três tipos de abordagem:

- A abordagem segundo as leis de Gestalt,
- A abordagem segundo técnicas visuais,
- A abordagem segundo técnicas estruturais.

3.3.2 Apresentação da informação segundo as leis de Gestalt

A disposição dos vários elementos no monitor de um sistema de controlo, no ecrã de uma televisão ou em qualquer dispositivo de interface segue regras que podem ser consideradas como regras fundamentais de percepção e cognição do sistema visual humano

[Wertheimer58]. Estas regras foram inicialmente descritas com bastante profundidade por vários autores no início do século XX e são conhecidas como as leis de Gestalt (forma, figura). A disposição da informação deve ser efectuada de acordo com este conjunto de leis que se resumem nas secções seguintes.

3.3.2.1 Lei da condensação (*terseness*)

O cérebro humano organiza os elementos de forma a minimizar a densidade de informação do padrão memorizado, pelo que, se existirem diferentes interpretações possíveis é adoptada aquela que apresenta menor complexidade. No exemplo da Figura 7 são apresentados alguns elementos simples (linhas, áreas) que são percebidos pelo observador como um quadrado branco sobre círculos, pretos nos seus cantos, e sobre um rectângulo branco com contorno preto.

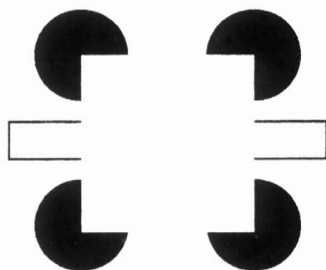


Figura 7- Lei da condensação

3.3.2.2 Lei da proximidade (*proximity*)

Os elementos colocados numa posição próxima entre si serão percebidos como pertencentes a um mesmo grupo. Na Figura 8 a) a percepção é orientada para as colunas enquanto que na Figura 8 b), pelo contrário, ela é orientada para as linhas. Os elementos que possuam alguma relação semântica devem ser colocados próximos uns dos outros, devendo ser colocados separadamente aqueles que não a possuem.

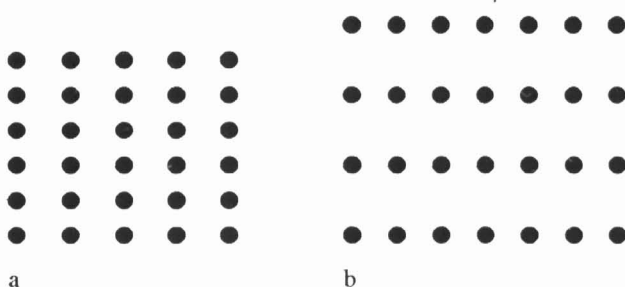


Figura 8 - Lei da proximidade

3.3.2.3 Lei da similaridade (similarity)

Os elementos com a mesma forma ou cor serão percebidos como pertencentes a um mesmo grupo. Na Figura 9, apesar da distância dos círculos aos rectângulos ser menor que a distância entre rectângulos, os círculos são percebidos como fazendo parte de um grupo e os rectângulos de outro. Os elementos que tenham alguma relação semântica devem ter a mesma forma ou cor.

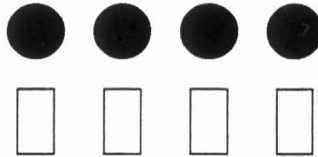


Figura 9 - Lei da similaridade

3.3.2.4 Lei da consistência (consistency)

A informação gráfica incompleta é completada no cérebro do observador através do seu sistema visual. Na Figura 10, a série de rectângulos de dimensão crescente é percebida como um triângulo. Se a figura representasse uma escala ela seria percebida de forma contínua.

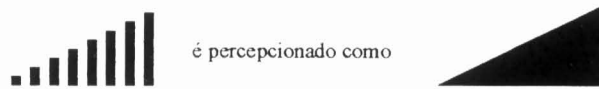


Figura 10 - Lei da consistência.

3.3.2.5 Lei da continuidade (continuity)

O sistema visual percebe os contornos de tal forma que o contorno original seja continuado da forma mais suave. Na Figura 11, as linhas são percebidas como (AC)(BD) e nunca como (AB)(CD) [Wertheimer58].

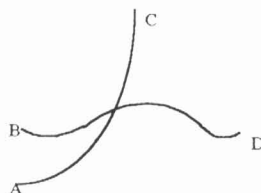


Figura 11 - Lei da continuidade.

3.3.2.6 Lei da simetria (*symmetry*)

Uma formação simétrica é facilmente detectada como característica dominante e constitui um contributo especial para a qualidade de percepção de um padrão. A Figura 12 mostra o esboço de um *layout* simétrico.

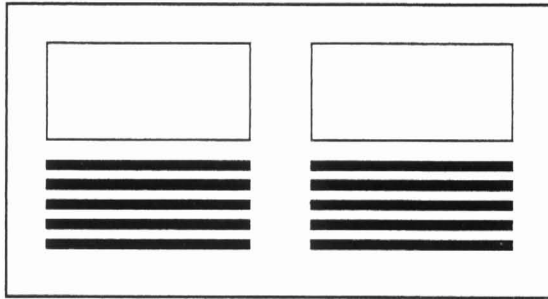


Figura 12 - Lei da simetria.

3.3.2.7 Lei das zonas de sobreposição (*overlapping areas*)

Se dois elementos estiverem colocados sobrepostos no monitor, é assumido pelo observador que o mais pequeno está colocado por cima do maior. Na Figura 13, o quadrado parece estar situado num plano acima do círculo.

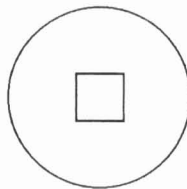


Figura 13 - Lei das zonas de sobreposição

As leis de Gestalt podem interagir umas com as outras e nesse caso dominará a que se mostrar mais efectiva. Em certos casos, a lei de consistência pode dominar a lei de proximidade, a de proximidade dominar a de similaridade, etc.

3.3.3 Apresentação da informação segundo técnicas visuais

[Vanderdonck94] define técnicas visuais para a apresentação da informação que se baseiam nos critérios definidos pelos *designers* gráficos. Estas técnicas dividem-se em cinco grandes áreas:

- **técnicas físicas**

equilíbrio, simetria, regularidade, alinhamento, proporcionalidade e horizontalidade.

- **técnicas de composição**

simplicidade, economia, neutralidade, singularidade, negatividade e transparência.

- **técnicas de associação e dissociação**

unicidade, repartição, agrupamento e leveza.

- **técnicas de ordenação**

consistência, previsibilidade, sequencialização e continuidade.

- **técnicas fotográficas**

diferenciação, arredondamento, estabilidade, nivelamento, acção, subtiliza, representação e realismo.

Nas secções seguintes será apresentada uma breve explicação das várias técnicas e descritos alguns exemplos característicos.

3.3.3.1 *Técnicas físicas.*

- ***Equilíbrio***

Deve-se procurar um equilíbrio entre a quantidade e “peso” dos objectos de interacção (OI) que se colocam no monitor, tanto no eixo vertical como no eixo horizontal. Esta característica é diferente da simetria, pois podem fazer-se *layouts* equilibrados sem serem simétricos.

- ***Simetria***

Consiste, basicamente, em duplicar os OI ao longo do eixo vertical ou horizontal.

- ***Regularidade***

Por exemplo, um *layout* uniformemente espaçado em colunas e linhas.

- ***Alinhamento***

O alinhamento é conseguido se reduzirmos os pontos de alinhamento vertical das linhas e horizontal das colunas. A Figura 14 demonstra este princípio: no grupo da esquerda existem dois pontos de alinhamento vertical enquanto no da direita existem seis pontos de alinhamento.

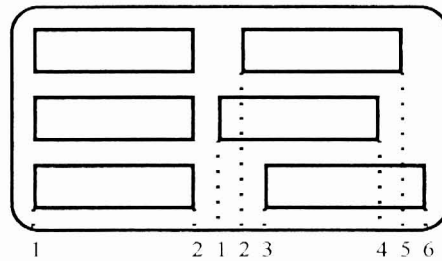


Figura 14 - Alinhamento e desalinhamento vertical.

- **Proporcionalidade**

Reside na tentativa de realizar OI com uma relação apelativa entre as suas dimensões. Geralmente é usada a chamada regra de ouro $1:\frac{1+\sqrt{5}}{2}$, usada pelos gregos, ou formas mais simplificadas ($1:\sqrt{2}$, $1:2$, $1:1,29$, ...), que representa a relação entre a altura e largura de um dado OI.

- **Horizontalidade**

Está demonstrado que os OI com maior largura que altura são predominantes.

3.3.3.2 Técnicas de composição.

- **Simplicidade**

Deve-se procurar efectuar *layouts* o menos complexos possíveis. A simplicidade é garantida se colocarmos os OI de uma forma lógica e natural. O contrário de simplicidade é a complexidade. A Figura 15 mostra um exemplo de um layout simples e de um layout complexo.

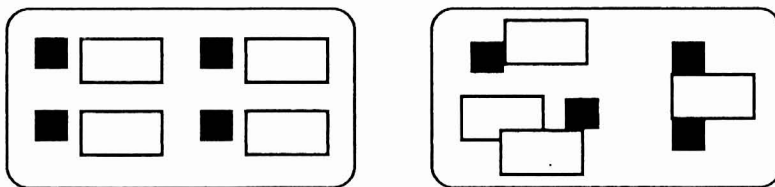


Figura 15 - Layout simples versus layout complexo.

- **Economia**

Deve-se colocar no *layout* somente o número suficiente e necessário de OI, não usando elementos decorativos.

- **Neutralidade**

É conseguida uma atmosfera neutral se colocarmos os OI todos ao mesmo nível e com os mesmos atributos de apresentação e, se possível, sem ênfase.

- **Singularidade**

Consiste em usar um só OI de forma a atrair a atenção para ele.

- **Negatividade**

Dispor os OI de cores escuras sobre fundo claro. A negatividade reduz os erros e o tempo de pesquisa. O contrário, a positividade, menos aconselhada, consiste em colocar OI claros em fundo escuro. A Figura 16 mostra um exemplo de negatividade e positividade.



Figura 16 - Negatividade *versus* positividade.

- **Transparência**

Um OI que seja sobreposto por outro deve, em certos casos, continuar visível. Esta técnica é válida quando, por exemplo, se quer colocar texto por cima de uma imagem.

3.3.3.3 Técnicas de associação e dissociação.

- **Unicidade**

Os OI parecem pertencer uns aos outros, fazer parte de um mesmo conjunto. Ver um dos elementos é ver todos. A unicidade pode ser implementada com cores e tamanhos similares ou outra forma de interligação. A Figura 17 mostra a diferença entre a unicidade e o seu contrário, a fragmentação.

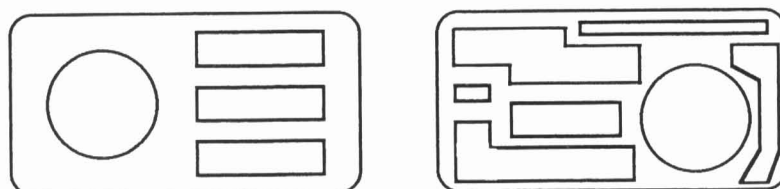


Figura 17 - Unicidade *versus* fragmentação.

- **Repartição**

Os OI devem estar repartidos pelos quatro quadrantes de forma sistemática. Algumas experiências mostraram que a utilização repartida dos quadrantes nem sempre era a mais vantajosa. Assim propõem 40%, 20%, 15% e 25% para o quadrante superior esquerdo, superior direito, inferior direito e inferior esquerdo respectivamente. No entanto convém referir que o olho humano prefere o quadrante inferior esquerdo.

- **Agrupamento**

É conseguido através de técnicas de similaridade e atracção entre os OI. O contrário é a separação.

- **Leveza**

O *layout* não deve estar sobrecarregado de OI, mas sim situar-se dentro de níveis aceitáveis. O contrário é geralmente *designado* por ultra-densidade (Figura 18).

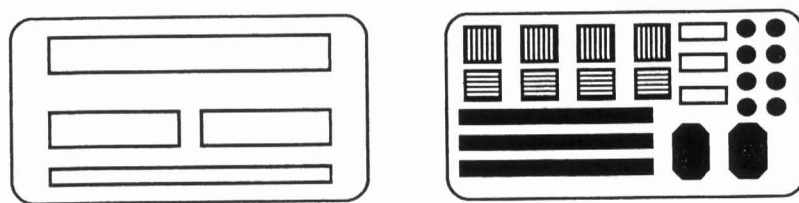


Figura 18 - Leveza versus ultra-densidade.

3.3.3.4 Técnicas de ordenação.

- **Consistência**

Deve existir alguma compatibilidade entre os OI utilizados. As suas variações de cor, tamanho, forma ou som, devem ser pequenas.

- **Previsibilidade**

Sugere ao utilizador a totalidade do *layout* observando somente uma parte deste (Figura 19).

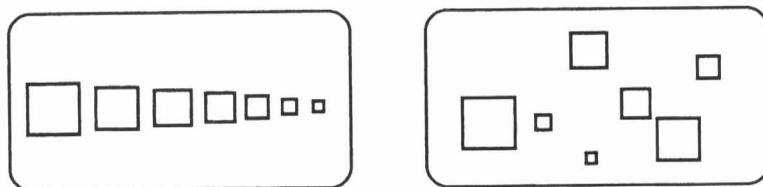


Figura 19 - Previsibilidade versus imprevisibilidade.

- **Sequencialização**

Quando o *layout* é arranjado de forma lógica, rítmica e com uma ordem esperada. Esta ordem pode ser numérica, alfabética, cronológica, física, sequencial, funcional, lógica, de frequência, importância, consenso, *designação*, etc.

- **Continuidade**

A continuidade assegura a conexão visual existente entre os vários OI.

3.3.3.5 Técnicas fotográficas.

- **Diferenciação**

Os OI devem ser perfeitamente distinguíveis entre si. As suas fronteiras devem ser precisas, bem delineadas e com margens distintas. O contrário desta técnica é a difusão, em que os elementos parecem misturar-se e não possuem uma clara separação.

- **Arredondamento**

Usado para atribuir ao *layout* uma atmosfera suave. A angulosidade é o oposto de arredondamento. A Figura 20 mostra a diferença entre um layout baseado na técnica de arredondamento e um layout baseado na técnica de angulosidade.

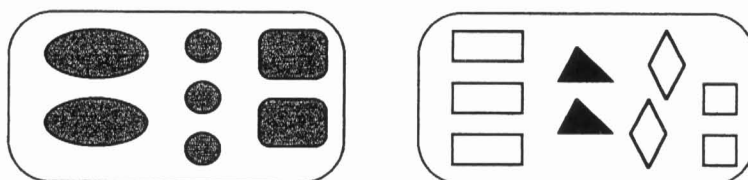


Figura 20 - Arredondamento *versus* angulosidade.

- **Estabilidade**

Devem ser usados OI na sua posição natural de “descanso”. Um rectângulo pousado sob a sua base transmite uma sensação de estabilidade. Pelo contrário um círculo não parece estável e provoca sim uma sensação de *stress*. Esta diferença encontra-se representada na Figura 21.

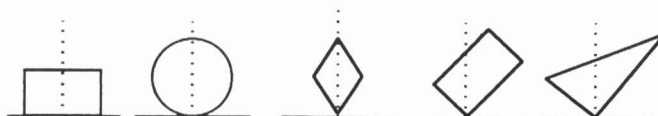


Figura 21 - Rectângulo estável; círculo, losango, rectângulo e triângulo instáveis.

- **Nivelamento**

Técnica visual que permite estabelecer automaticamente o equilíbrio. Quando os OI são colocados no *layout* são automaticamente representados os eixos de equilíbrio deste (Figura 22). Esta técnica permite orientar a colocação de novos OI.

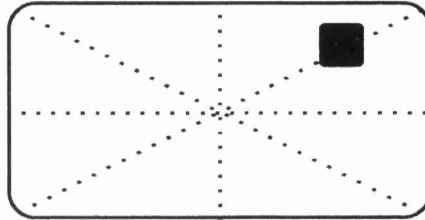


Figura 22 - Eixos de equilíbrio do *layout*.

- **Acção**

Reflecte movimento através de representação explícita ou sugestão implícita.

- **Subtileza**

Os OI devem ser dispostos de uma forma “limpa” e organizada.

- **Representação**

Pressupõe o uso de OI que representem concretamente o mundo real. O contrário é a abstracção.

- **Realismo**

É a técnica natural da câmara de vídeo. São usados truques que permitem simular os estímulos que chegam ao cérebro quando recebe uma imagem do mundo. Um dos métodos usados é o da perspectiva.

3.3.4 Apresentação da informação segundo técnicas estruturais

Alguns autores optaram por não estabelecer regras visuais sobre a disposição dos vários elementos no monitor do computador ou em qualquer dispositivo de apresentação.

Em alternativa definiram a organização dos objectos segundo a sua estrutura e inter-relação, sem a ajuda de critérios visuais, mostrando que se deve combinar a informação visual com o conhecimento do mundo, adquirido através da longa experiência de interacção com ele.

Em resultado do projecto AMODEUS, [May95] definiu um guia baseado nessas características estruturais, que se apresentam de forma resumida nos pontos seguintes.

3.3.4.1 A estrutura das cenas visuais

Quando olhamos para uma cena, as várias características, texturas, cores, juntam-se para formar os vários objectos. A cena no seu todo é um conjunto de objectos.

Percepcionamos o mundo a várias escalas diferentes, a partir de um nível mais global e progredindo sucessivamente vários níveis de detalhe. Esta estrutura pode ser representada por uma árvore, como a da Figura 23, que representa a estrutura de uma cena de um escritório e que mostra as relações hierárquicas entre os vários objectos.

O objecto que constitui o ponto de interesse numa determinada altura é *designado* por "sujeito psicológico" e os objectos do mesmo nível estrutural são os seus predicados.

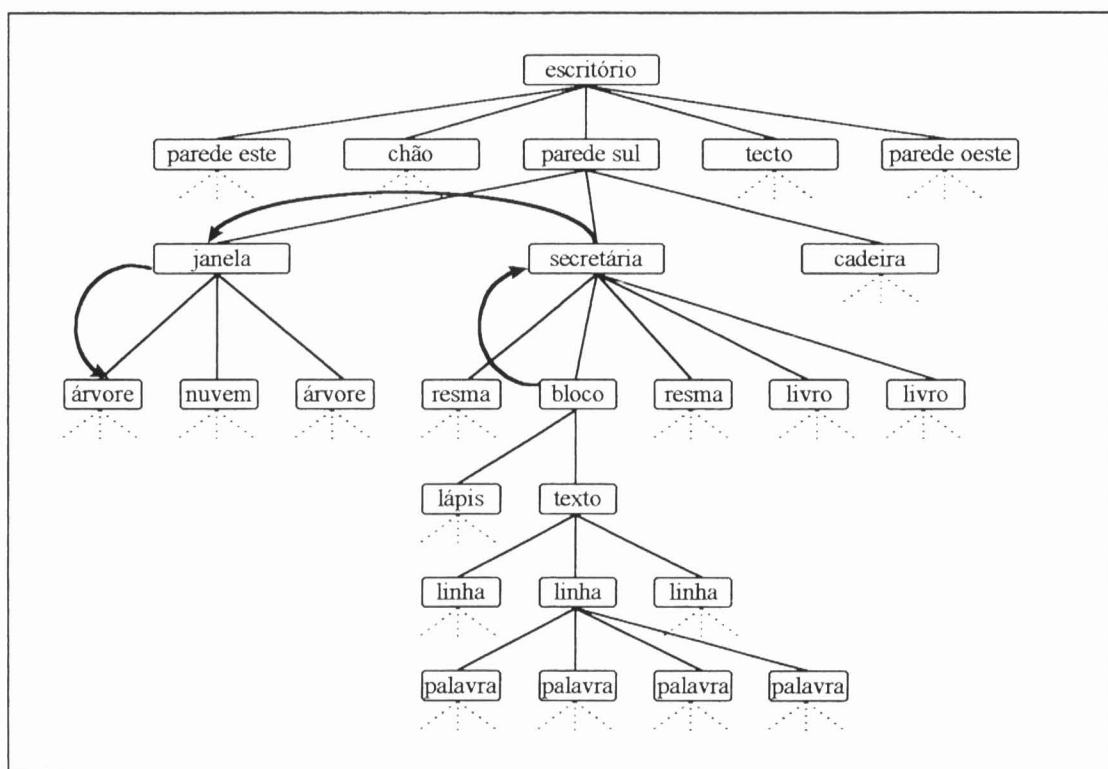


Figura 23 - Diagrama estrutural de uma cena.

A Figura 23 mostra também um conjunto de setas que representam a transição da atenção do observador do bloco para a secretária, daí para a janela e depois para a árvore. Por forma a simplificar a representação das transições de pontos de interesse são utilizados diagramas de transição como o representado na Figura 24 que corresponde à parte da Figura 23 com as setas de transição.

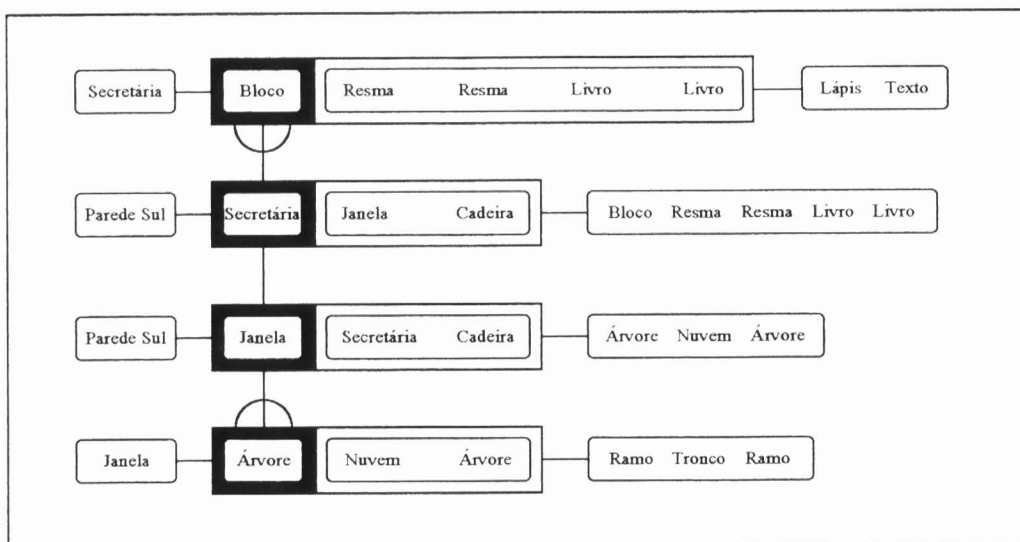


Figura 24 - Diagrama de transições.

Cada linha representa um dado momento no tempo. O objecto sobre fundo negro representa o sujeito psicológico, sendo os outros da mesma linha (rectângulo) os seus predicados. Os blocos à esquerda representam o grupo a que pertence o objecto e os blocos à direita representam os constituintes estruturais do objecto psicológico. As forquilha representam a subida e descida de nível; para o lado da forquilha tem lugar o “*zoom in*” e para o lado do cabo da forquilha o “*zoom out*”.

Como podemos constatar, com a ajuda deste tipo de diagramas, é possível com alguma facilidade verificar se é mais fácil ou mais complicado para o observador desviar a sua atenção de um objecto para outro. O “trabalho” necessário para mudar a atenção de um objecto para outro torna-se maior ou menor quanto mais ou menos níveis, respectivamente, tivermos que percorrer.

Analisemos um exemplo simples, como o da Figura 25. Os dois painéis diferem apenas na forma como os vários objectos estão agrupados. A tarefa pretendida seria a de ligar a luz do quarto 133 no primeiro painel e o aquecimento do quarto 133 no segundo painel.

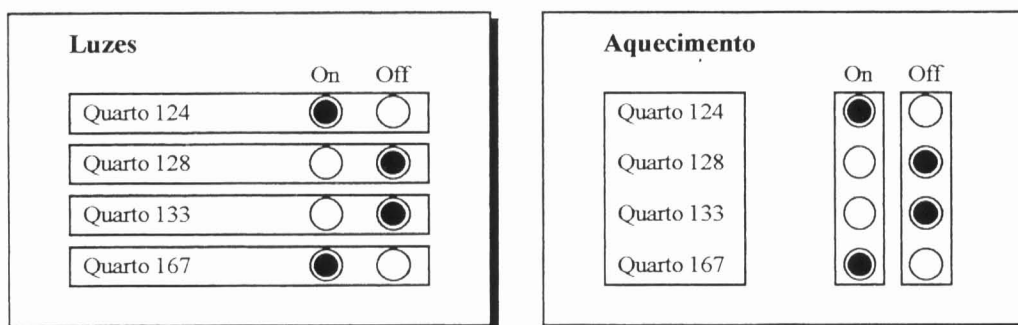


Figura 25 - Dois painéis de controlo que diferem na forma como os objectos estão agrupados.

Elaboramos os diagramas de transições representados na Figura 26 a) e Figura 26 b).

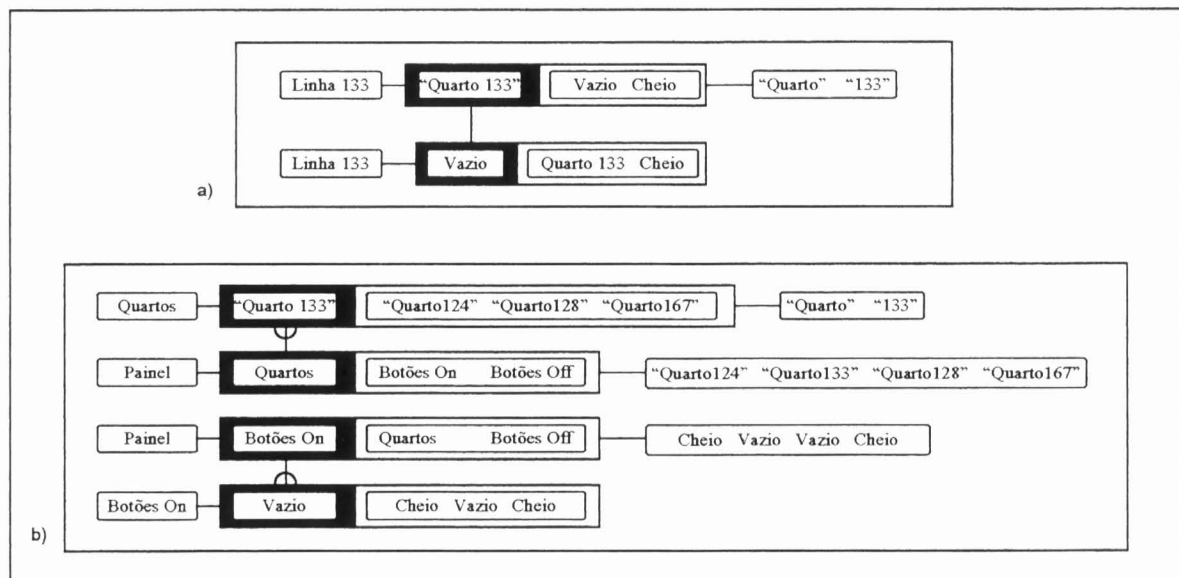


Figura 26 - Diagrama de transições referente aos painéis de luzes (a) e aquecimento (b).

Podemos confirmar que o painel de luzes se torna mais fácil de usar que o painel de aquecimento. No painel de luzes necessitamos somente de uma transição para realizar a tarefa pretendida enquanto que no painel de aquecimento precisamos de três transições.

É óbvio que a tarefa a executar influencia enormemente a escolha do agrupamento ideal de objectos com vista a um mais fácil uso por parte do utilizador. Se, no exemplo que mostramos anteriormente, a tarefa fosse colocar todos os botões a On, então o painel de aquecimento mostrava-se mais apropriado. Isto mostra, mais uma vez, que o desenvolvimento de interfaces com o utilizador deve, acima de tudo, entender e analisar com bastante rigor as tarefas a executar, para que a interface se adeque às necessidades do utilizador.

Através destes diagramas de transições podemos também definir um determinado número de regras aplicáveis às estruturas de objectos. As regras que se seguem fazem parte deste estudo:

- Os objectos que, dentro de um grupo, possuem características diferentes, sobressaem relativamente aos outros elementos do grupo (Figura 27). Transformam-se em "sujeitos pragmáticos". Os sujeitos pragmáticos são aqueles que, de alguma forma, parecem atrair a atenção sobre si.

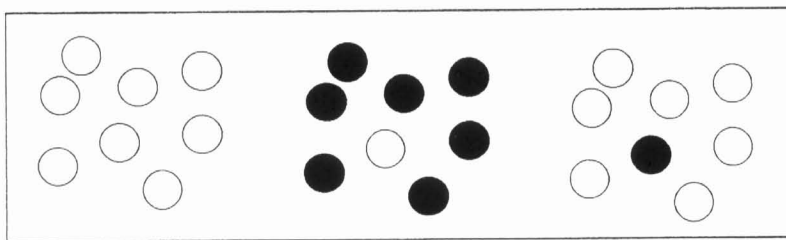


Figura 27 - Objectos com características diferentes sobressaem.

- Objectos que sejam maiores ou mais brilhantes parecem mais perto do observador e podem transformar-se em sujeitos pragmáticos (Figura 28). No primeiro agrupamento o círculo maior torna-se no sujeito pragmático, enquanto que no segundo, ao contrário do que seria de esperar (o círculo menor tornar-se o sujeito pragmático), tal não sucede. Os círculos grandes formam também um grupo pragmático.

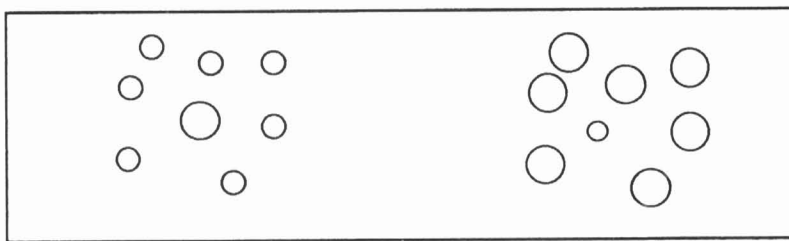


Figura 28 - Objectos maiores ou mais brilhantes tornam-se sujeitos pragmáticos.

- Objectos com estrutura mais complexa sobressaem sobre aqueles mais simples (Figura 29). No primeiro conjunto, quando colocamos um círculo incompleto no meio de outros círculos completos, ele é facilmente reconhecível (torna-se no sujeito pragmático). No segundo conjunto, o objecto com estrutura mais simples parece esconder-se no meio dos outros.

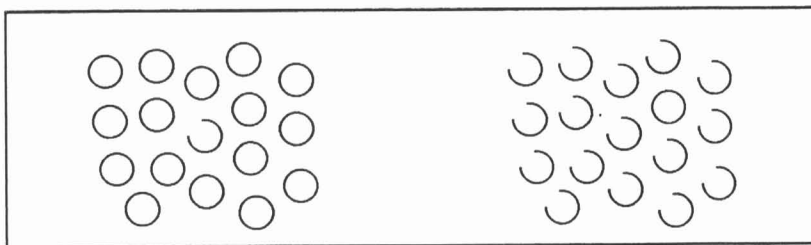


Figura 29 - A estrutura afecta o agrupamento.

- O conhecimento pode afectar a estrutura. Na Figura 30, o cubo que aponta para o lado contrário é facilmente identificável e transforma-se no sujeito pragmático. Este facto

não se deve a qualquer outro factor já conhecido como o tamanho, forma, cor, etc., o que podemos constatar no outro conjunto da mesma figura. Aqui os objectos têm todos o mesmo número de linhas e ângulos e apesar de existir um que não está orientado para o mesmo lado, ele não sobressai.

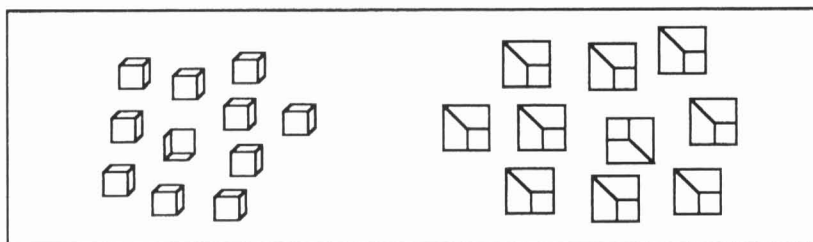


Figura 30 - O conhecimento pode afectar a estrutura.

Se, entretanto, dermos uma indicação do significado dos objectos do segundo conjunto, por exemplo uma caixa que desce para um plano mais afastado, depois de formarmos uma imagem mental desse objecto, torna-se bastante mais fácil proceder à identificação do objecto que aponta no outro sentido. No entanto o único facto que se alterou neste procedimento foi o nosso conhecimento do objecto, um dos seus possíveis significados. Isto mostra que o conhecimento do significado da estrutura de um dado objecto pode afectar a estrutura do próprio objecto. Chama-se a este processo reconhecimento com *feedback*, e encontra-se esquematizado na Figura 31.

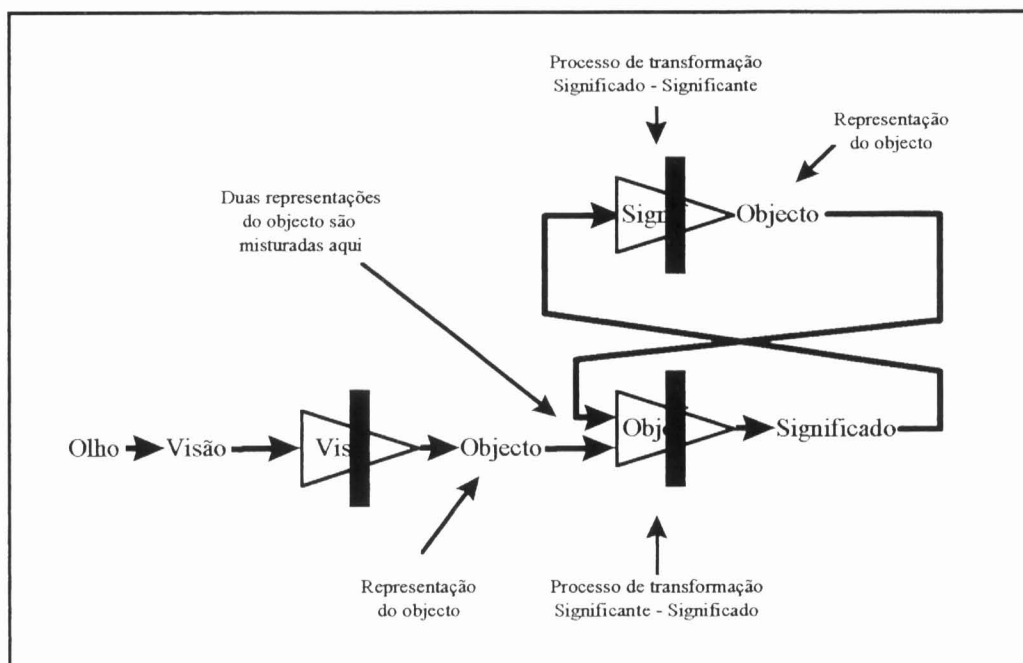


Figura 31 - Processo de feedback entre o objecto e o seu significado.

- No agrupamento de objectos a cor é dominante sobre a forma (Figura 32).

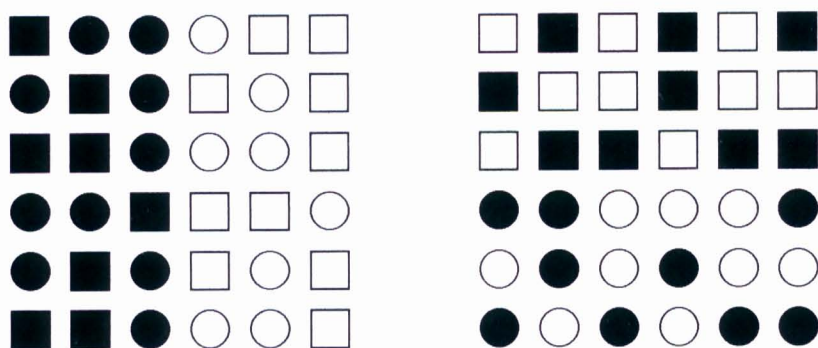


Figura 32 - A cor é dominante sobre a forma.

- O movimento pode ser dominante sobre a cor e a forma. Objectos que se alteram ao longo do tempo podem ser percebidos como sendo o mesmo objecto, desde que a mudança não seja tão brusca que altere o significado do objecto. Isto introduz um novo conceito de “continuidade temática” usado com frequência em cenas de filmes como a representada na Figura 33.

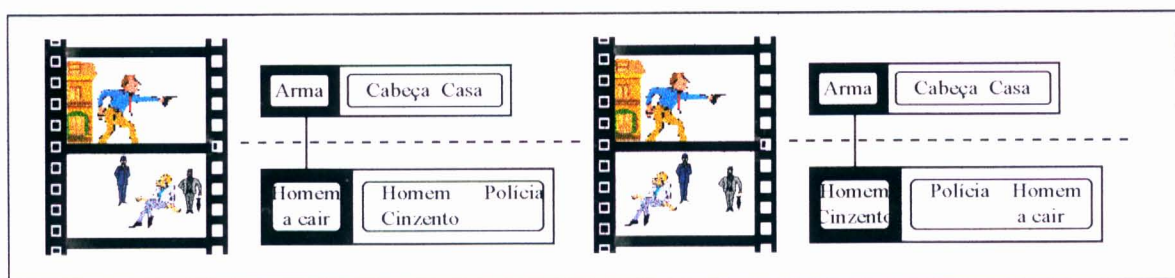


Figura 33 - Continuidade temática da estrutura.

3.3.5 Outros critérios

Por vezes, a informação deve ser apresentada tendo em conta critérios que não são de natureza visual, mas sim dependentes de conceitos ou situações específicas.

A apresentação da informação deve ter em consideração, por exemplo, as preferências culturais de leitura dos utilizadores. Na cultura oriental a leitura não se processa da esquerda para a direita e de cima para baixo, como acontece na cultura ocidental.

Podemos dizer que a apresentação da informação e dos elementos de controlo deve ter em consideração pelo menos um destes critérios:

- ***Importância***

Se existirem elementos mais importantes que outros estes devem ser colocados numa zona mais proeminente do monitor.

- ***Sequência de uso***

Os elementos de diálogo a serem usados numa determinada sequência devem ser apresentados nessa mesma ordem.

- ***Frequência de uso***

Os comandos mais frequentes devem ter um acesso mais fácil por parte do utilizador. Se, por exemplo, o utilizador deve escolher entre vários comandos possíveis de uma lista de menus, os mais frequentes devem ser colocados no início da lista.

- ***Convenção de uso***

Devem ser seguidas as convenções de uso preestabelecidas.

- ***Cronológico***

Obedecer à ordem cronológica dos elementos que a possuam.

- ***Ordem alfabética***

Pode-se seguir o critério da ordem alfabética, se não existir outra relação semântica mais apropriada.

Neste capítulo foram apresentadas as questões relacionadas com a interface com o utilizador e com a apresentação da informação. No próximo capítulo abordaremos os aspectos sobre o processo de desenvolvimento de interfaces gráficas com o utilizador.

Capítulo 4

O Desenvolvimento de Interfaces com o Utilizador

O desenvolvimento de interfaces com o utilizador abrange áreas que vão desde métodos de desenvolvimento, regras de concepção de interfaces, ferramentas de desenvolvimento, passando por aspectos relativos ao uso da cor e som. Numa primeira parte são focados os vários métodos de desenvolvimento de interfaces com o utilizador, apresentando com algum detalhe alguns dos aspectos mais importantes das várias fases. Seguidamente apresentamos os princípios de concepção e por fim algumas ferramentas usadas no apoio ao desenvolvimento de interfaces e os principais factores relacionados com a sua escolha e uso.

4.1 O Processo de Desenvolvimento

Existem actualmente várias abordagens sobre o desenvolvimento de interfaces com o utilizador. Estas abordagens vão desde os tradicionais métodos de desenvolvimento baseados na engenharia de *software* até métodos de desenvolvimento participados em que os utilizadores finais colaboram em todo o processo.

Algumas regras básicas são, no entanto, evidentes. O *design* de uma interface deve ser consolidado com um núcleo, estrutura, sólidos, como num edifício, para que se obtenham bons resultados [Ford97]. Qualquer edifício “aceitável” providencia abrigo e as funções básicas aos seus ocupantes, no entanto um edifício bem concebido, bem projectado, vai mais longe; ele fornece aos seus utilizadores os meios para se focarem mais nos seus objectivos e não na forma como o edifício está construído. Obviamente que isto depende da particularidade dos objectivos. Num edifício público deve providenciar-se, aos utilizadores que não estejam familiarizados com os procedimentos, formas de navegação intuitivas.

Tanto no interface com o utilizador como num edifício, um bom *design* baseia-se em princípios como usabilidade e funcionalidade, que satisfaçam as necessidades reais. Apesar do foco ser a facilidade de uso, não se devem descurar os aspectos estéticos.

4.1.1 Metodologias de Desenvolvimento

Quanto às metodologias propriamente ditas podemos analisar aquelas em uso pela engenharia de *software* e as metodologias mais usadas na comunidade HCI (*Human Computer Interaction*).

As metodologias seguidas pela engenharia de *software* para o desenvolvimento de aplicações podem ser assim divididas [Brown97]:

- ***Ciclo de Vida ou Queda de Água***

Esta metodologia envolve seis grandes etapas: Engenharia, Análise, *Design*, Código, Teste e Manutenção.

Nenhuma destas etapas aborda especificamente o desenvolvimento da interface.

- ***Prototipagem***

Neste tipo de metodologia uma série de protótipos são implementados até se atingir uma boa compreensão dos requisitos do *software*.

- ***O Modelo Espiral***

Esta metodologia é desenvolvida num contexto empresarial. Aqui os gestores avaliam a exequibilidade de um projecto, os recursos necessários e os riscos inerentes. Cada ciclo da espiral avalia se se passa ou não para o ciclo seguinte. São definidos quatro passos: Planeamento, Análise de risco, Engenharia e Avaliação de consumo.

- ***Ciclo de Desenvolvimento com Ênfase no Teste***

Semelhante à metodologia de queda de água mas com ênfase no teste. Nesta abordagem todos os passos estão interligado permitindo o desenvolvimento por prototipagem.

- ***Ciclo de Vida com conhecimento HCI.***

Esta metodologia é proposta para os projectistas e engenheiros de *software* com uma grande experiência técnica na qual o desenvolvimento da interface é acompanhado com o conhecimento sobre modelos cognitivos e perceptivos, regras de diálogo e técnicas de avaliação informais.

- ***Ciclo de Vida com Participantes e Especialistas em HCI***

Semelhante à anterior mas onde existe uma intervenção no desenvolvimento do sistema de especialistas em HCI. Todavia mantém-se o esquema do ciclo de vida.

Na comunidade HCI existe a convicção que é necessária uma técnica de *design* iterativo para o desenvolvimento de interfaces com o utilizador. A metodologia da queda de água não é indicada para o desenvolvimento de interfaces visto que a interface com o utilizador não pode ser especificada sem se efectuar uma série de testes com o utilizador. É necessária um abordagem experimental já que não existem teorias suficientes sobre o comportamento e o processo cognitivo humano para produzir um *design* de interfaces com base teórica.

As metodologias mais usadas na comunidade HCI são assim divididas:

- ***Centradas no Utilizador***

- Metodologia de Uso: elaboração rápida de protótipos não acompanhada pelo desenvolvimento dos aspectos funcionais da aplicação. Isto permite efectuar com alguma facilidade os protótipos que serão testados com o utilizador e iterativamente melhorados.
- Engenharia de “Usabilidade” - apresenta dez passos:
 - Conhecimento do utilizador
 - Análise competitiva
 - Definição dos objectivos de usabilidade
 - *Design* paralelo
 - *Design* participado

- *Design* coordenado da interface total
 - Aplicação de regras e análise heurística
 - Elaboração de protótipos
 - Teste empírico
 - Recolher *feedback* dos utilizadores
-
- **Modelação Cognitiva**
 - Metodologias não enfatizadas: O objectivo da modelação cognitiva é compreender e modelar uma actividade como é entendida pelo utilizador. Os especialistas em modelação cognitiva defendem que se conseguirmos realizar estes modelos então seremos capazes de criar interfaces que serão intuitivas para o utilizador. Não são usados métodos especiais ou metodologias particulares mas, no entanto, existem alguns métodos seguidos por estes especialistas como o GOMS (Goals, Operators, Methods and Selection Rules).
-
- **Design Participado**
 - Sem metodologia particular: Os defensores do *design* participado advogam que não vêm o uso de metodologias como a forma de produzir uma boa interface, no entanto, existem alguns métodos nesta abordagem.
 - Ênfase na Comunicação: Para os especialistas em *design* participado o aspecto mais importante é a comunicação entre utilizadores e projectistas. Este factor é ainda mais importante no *design* participado pois ele assenta a sua actuação na participação do utilizador no processo. O factor importante no *design* participado é a comunicação que ocorre entre os projectistas e os utilizadores, mais que qualquer método em particular. Segundo Kensing, citado em [Brown97], as técnicas ou metodologias não são a resposta. Ele propõe a criação de modelos que facilitem e melhorem a comunicação entre utilizadores e projectistas. Ele classifica o *design* participado e as técnicas de engenharia de *software* em seis diferentes categorias de comunicação apresentadas na Figura 34:

Trabalho Actual do Utilizador	Novo Sistema	Opções Tecnológicas
Conhecimento Abstracto		
Estruturas relevantes no trabalho actual do utilizador.	Propostas de <i>design</i>	Análise das opções tecnológicas
Experiência Concreta		
Experiência concreta com o trabalho actual do utilizador	Experiência concreta com o sistema novo	Experiência concreta com as opções tecnológicas

Figura 34 - Categorias de Comunicação entre utilizador e projectista segundo Kensing.

Pode concluir-se que os projectos que envolvem uma interface com o utilizador são normalmente mais complicados de desenvolver porque exigem um maior esforço, tanto financeiro como de tempo despendido, havendo a necessidade da realização de etapas próprias para especificação e teste das interfaces assim como, em alguns casos, a hipótese da existência da participação dos utilizadores em todas as etapas do desenvolvimento.

Vimos também que existe uma grande diferença entre a abordagem realizada pelos engenheiros de *software* e os especialistas em interacção homem-máquina. Os especialistas em HCI são aqui que podemos chamar de “Centrados no Utilizador” enquanto que os engenheiros de *software* são “Centrados no Sistema”. Os especialistas HCI testam ideias de *design* em utilizadores reais e usam técnicas de avaliação formal como substituição da intuição no desenvolvimento da interface. É este contacto com a realidade que melhora constantemente o produto final.

4.1.2 Fases do Processo de Desenvolvimento

As metodologias de engenharia de *software* preocupam-se mais com certos aspectos do domínio do problema. Desenvolveram vários métodos para representar os aspectos da arquitectura, informação e procedimentos de um sistema de *software*. Os especialistas em HCI testam as ideias de *design* nos utilizadores reais e usam técnicas formais de avaliação, por forma a substituir a intuição no *design* guiado. De referir a propósito que alguns autores defendem a criação de interfaces baseadas na intuição e imaginação [Nielsen93]. Outros aplicam o processo de *design* por crítica para melhorar interfaces já existentes [Vertelney90].

O processo de desenvolvimento tem que ser iterativo, ou seja, as várias etapas podem ser repetidas conforme a evolução do projecto. A Figura 35 representa um ciclo iterativo e apresenta os vários processos de trabalho e modelos mentais.

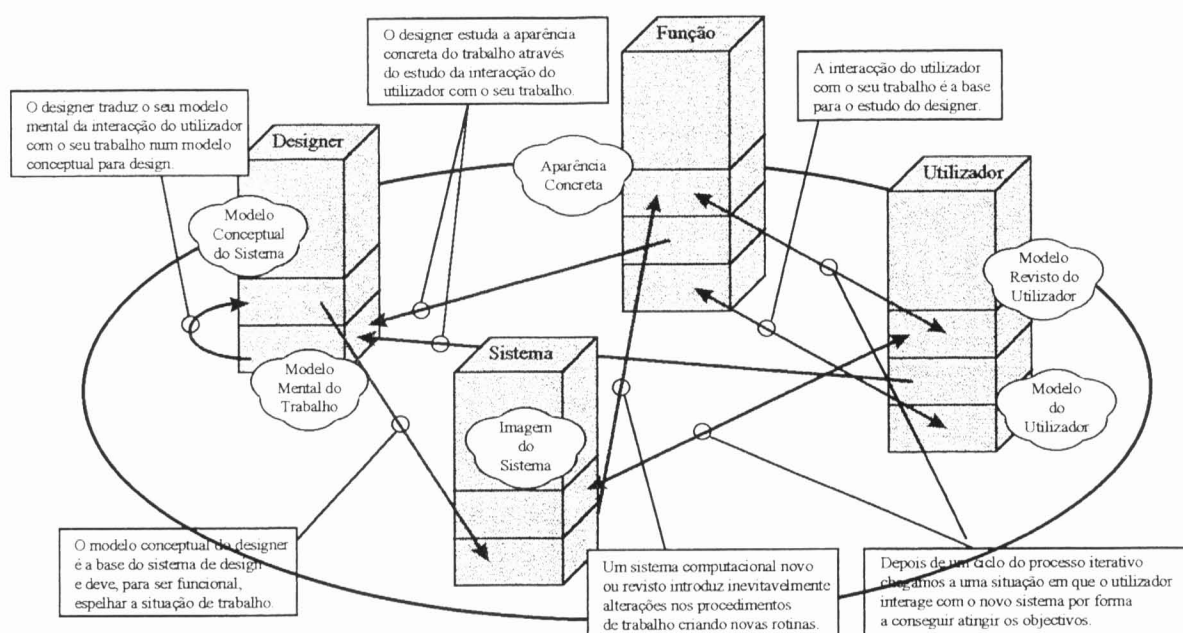


Figura 35 - Um Ciclo Iterativo no processo de desenvolvimento [Gulliksen95b].

O método iterativo mostrou melhorar a usabilidade do produto final.

O método “ideal” deve também ser desenvolvido com a participação do utilizador final, de preferência em todas as fases do projecto [Wilson95]. Esta participação produz economia em termos de tempo, especialmente na fase de elaboração de protótipos e teste.

Nos pontos seguintes fazemos uma análise pormenorizada sobre as várias etapas deste método “ideal”, que é uma junção do método de engenharia de usabilidade com o *design* participado com ênfase na comunicação.

4.1.2.1 Conhecimento do utilizador

Esta etapa do projecto trata do conhecimento do utilizador que irá utilizar o sistema. É necessário definir classes representativas dos utilizadores em termos das tarefas a serem executadas e dos conhecimentos e competências que possam influenciar essas tarefas. Aspectos como o grau de instrução, a experiência com computadores, o tipo de *hardware* e *software* com que está mais familiarizado, se tem experiência com aplicações semelhantes, podem ser considerados para a estruturação das classes. O conhecimento do utilizador e das suas necessidades é crítico para a qualidade final do produto. Para além de fazer o que se espera do sistema, ele deve fundir-se no mundo do utilizador e no seu trabalho. Uma das formas de iniciar o trabalho de conhecimento do utilizador é começar pela observação do seu ambiente e métodos de trabalho.

São conhecidas as diferentes sensibilidades em termos culturais dos vários utilizadores, já abordadas no capítulo anterior quando falámos da apresentação da informação. [Yeo96] introduz o conceito de CUI - *Cultural User Interface* e propõe estratégias para o seu desenvolvimento.

Deve fazer-se um esforço para juntar especialistas na área em que o sistema vai ser usado e nas áreas dos computadores. É sabido que cada área tem os seus termos próprios e a sua gíria.

4.1.2.2 *Análise*

É necessária uma descrição completa das tarefas, subtarefas e métodos envolvidos na utilização do novo sistema, identificando os recursos necessários para que os utilizadores e o sistema cooperem na realização dessas tarefas. Identifica-se um conjunto hierárquico de tarefas representativas que definam o que a interface deve permitir realizar.

É também necessário efectuar uma análise funcional, isto é, uma descrição das funções internas do sistema que devem ser projectadas e implementadas na parte do *software* da aplicação, para realizar as tarefas identificadas, de forma a satisfazer as directivas iniciais.

4.1.2.3 *Design da interface*

Esta fase não é ainda o protótipo da interface mas sim uma fase anterior em que se faz uma descrição inicial, que pode ser feita no papel, dos vários contextos de interacção com o utilizador. Nesta fase é importante que se possam realizar os primeiros testes através de uma análise detalhada por parte dos *designers* e até utilizadores finais para eliminar alguns dos problemas que possam surgir. Isto evita que a fase de criação e teste de protótipos seja, por força da quantidade de problemas ser provavelmente menor, bastante mais rápida. Irão surgir, concerteza, menos problemas e assim o melhoramento da interface será menos demorado. Aspectos a serem considerados nesta fase são, por exemplo, o número de teclas e operações mentais (decisões) para realizar uma dada tarefa que a interface deve suportar. Isto permite identificar tarefas que necessitem um número demasiado elevado de passos para serem realizados.

No sentido de permitir uma melhor comunicação entre *designers* de interfaces, programadores e utilizadores, usa-se uma especificação. Geralmente os *designers* usam técnicas informais, *ad hoc*, para definir as suas interfaces. As consequências do uso destas técnicas incompletas e ambíguas originam diferentes interpretações por parte dos restantes elementos.

Sem uma especificação técnica completa, credível e compreensível, as interfaces serão, muito provavelmente, de difícil desenvolvimento. Uma boa especificação da interface

permite aos *designers* verificar a existência de inconsistências ou problemas de usabilidade antes que ela seja desenvolvida. Neste sentido tem-se vindo a estudar algumas técnicas para efectuar a especificação formal de interfaces apoiadas, por exemplo, em redes de *petri* ou modelos matemáticos. Podemos encontrar uma síntese das várias abordagens em [Rouff96].

4.1.2.4 Definição dos *objectivos de usabilidade*

Deve tentar-se estabelecer as condições de usabilidade o mais cedo possível no processo de desenvolvimento.

A especificação da usabilidade pretende estabelecer *objectivos* quantitativos de usabilidade que permitem determinar quando é que o desenvolvimento está a convergir na direcção de uma interface de sucesso. Deste modo torna-se evidente que a especificação da usabilidade é parte integrante dos dispositivos de controlo e gestão do processo de refinamento iterativo. A iteração final é atingida quando a versão actual da interface assegurar os valores mínimos de especificação da usabilidade.

É importante estabelecer especificações mensuráveis e monitorá-las em cada iteração para determinar se os *objectivos* iniciais foram, de facto, atingidos. As especificações de usabilidade envolvem considerações sobre atributos, instrumentos de medida, valores a serem medidos e níveis para esses valores.

Os atributos de usabilidade que são medidos têm de ser claramente quantificáveis e vão desde o desempenho na primeira utilização, o desempenho a longo prazo, a facilidade de aprendizagem e memorização dos aspectos aprendidos, a satisfação depois de usar o sistema por muito tempo, etc.

Os instrumentos de medida, que darão sempre resultados quantitativos, usados para atribuir valores aos vários atributos de usabilidade, podem ser *objectivos* ou *subjectivos*. Os instrumentos de medida *objectivos* são medidas do desempenho do utilizador perfeitamente mensuráveis e são obtidos quando o utilizador realiza tarefas predeterminadas, por exemplo, o tempo que um utilizador gasta na tarefa de adicionar um elemento a uma base de dados. Os instrumentos de medida *subjectivos* dão-nos valores quantitativos baseados na opinião do utilizador sobre a interface.

Os níveis utilizados na quantificação da usabilidade de uma interface podem ser classificados em [Marcos95]:

- *Nível Corrente*

É o nível actual do valor a ser medido para um determinado atributo de usabilidade, na versão actual do sistema.

- ***Pior Nível Aceitável***

É o mais baixo nível aceitável, e não o pior que pode ocorrer, para o desempenho do utilizador para cada atributo de usabilidade.

- ***Nível Planeado como Objectivo***

É o objectivo nominal de usabilidade planeado para cada atributo específico.

- ***Melhor Nível Possível***

Exprime o nível máximo de usabilidade que pode ser atingido para cada atributo específico.

Existem outros autores, como Rauch, que estabelecem um conjunto de técnicas de usabilidade a implementar em cada uma das fases do desenvolvimento de um projecto [Rauch96].

4.1.2.5 Elaboração de Protótipos

Nesta fase, depois de ter recolhido toda a informação sobre os utilizadores, as suas tarefas, ter-se efectuado o *design* da interface e definido os aspectos de usabilidade, avança-se para a criação de protótipos. A elaboração de protótipos, prototipagem, é o processo pelo qual se desenvolvem versões preliminares do nosso *design* para aferir da sua usabilidade.

Para a elaboração destes protótipos podem ser usadas várias técnicas. Uma delas consiste em efectuar um *storyboard* que não é mais que a elaboração em papel dos vários *displays* possíveis na execução de uma dada tarefa que o utilizador deve atingir [Wagner90]. Outra técnica possível consiste em executar uma interface sem a funcionalidade exterior (aplicação). Neste tipo de abordagem não é necessário implementar toda a funcionalidade da interface devendo prestar-se especial atenção na elaboração das partes da interface necessárias à avaliação das tarefas representativas.

4.1.2.6 Teste da Interface

Apesar de numa situação ideal o utilizador acompanhar o desenvolvimento de uma nova interface, serão sempre de esperar alguns problemas que só aparecerão quando se efectuarem teste mais rigorosos. Assim será sempre necessário efectuar uma série de testes para eliminar, idealmente, a totalidade das situações menos correctas.

Estes testes podem ser efectuados com ou sem a intervenção do utilizador. No entanto aconselha-se que sejam implementados os dois processos.

Os testes sem intervenção do utilizador são necessários porque a intervenção dos

utilizadores é geralmente demorada e nem sempre se pode dispor da sua presença prolongada. O tempo disponível dos utilizadores não é um recurso ilimitado, e assim devem efectuar-se testes sem eles para que numa fase posterior não sejam deparados com erros “triviais” que poderiam muito bem ter sido detectados por uma análise mais cuidada. Estes testes mostram também que por vezes se conseguem detectar erros que provavelmente não seriam detectados por utilizadores especialmente se o seu número for limitado. Não esquecer que um sistema que é avaliado por uma dúzia de pessoas pode vir a ser usado por milhões.

Os testes sem o auxílio do utilizador dividem-se em três abordagens distintas:

- ***Exercício Cognitivo***

O exercício cognitivo é um modo formal de imaginar os pensamentos e acções das pessoas, quando utilizam uma dada interface pela primeira vez. Sucintamente, o exercício desenvolve-se do seguinte modo: possuindo um protótipo ou descrição detalhada da interface, e conhecendo o utilizador tipo, selecciona-se uma das tarefas suportadas pela interface. Depois é necessário encontrar a motivação para cada uma das acções que o utilizador terá que executar de modo a realizar a tarefa.

- ***Análise de Acções***

A análise de acções é um processo de avaliação que força a olhar detalhadamente para a sequência de acções que o utilizador tem de realizar para completar uma tarefa.

A análise de acções apresenta duas fases fundamentais. A primeira consiste em decidir quais os passos físicos e mentais que o utilizador vai executar para completar uma ou mais tarefas na interface. A segunda fase consiste em analisar esses passos à procura de problemas.

- ***Avaliação heurística***

A avaliação heurística consiste em avaliar a interface, verificando se obedece às linhas mestras pré-estabelecidas. Esta avaliação faz-se analisando item a item a aplicação das regras estabelecidas. Esta avaliação deve ser efectuada por pessoas com um conhecimento considerável no âmbito da construção de interfaces. Preferencialmente, esta análise deve ser realizada por mais de um analista, que avalia a interface individualmente, sendo os resultados daí decorrentes combinados numa lista única que reflecte os problemas encontrados.

O outro tipo de testes possível, já referido anteriormente, é o **teste com utilizadores**. Este tipo de teste deve ser efectuado, idealmente, à medida que a construção do interface avança. Isto significa que é necessário escolher pessoas, utilizadores, para testar a interface, tarefas para estes realizarem e uma versão do sistema para ser testada. Abordaremos, seguidamente, cada uma destas necessidades.

- *Escolha dos Utilizadores*

Para recrutar os utilizadores, devem procurar-se pessoas com o mesmo nível de experiência que o futuro utilizador típico do nosso produto. Este facto evita, por exemplo, que o vocabulário usado na interface não seja adequado aos utilizadores futuros. Deve também evitar-se o recrutamento de pessoas familiarizadas com o nosso produto ou com as nossas opiniões acerca dele [Gomoll90].

- *Definição das Tarefas*

Um dos passos para a observação dos utilizadores consiste em seleccionar algumas tarefas destinadas a ser executadas pelos mesmos. Da mesma forma que o tipo de utilizadores deve ser representativo dos utilizadores reais, também as tarefas devem reflectir tarefas reais. Devem ser escolhidas tarefas com particular incidência no aspecto do produto que pretendemos avaliar. Por exemplo, se queremos saber se os nossos menus são úteis, devemos desenhar uma tarefa que requeira por parte do utilizador um acesso frequente aos mesmos.

Depois de definir as tarefas que vão ser usadas, estas devem ser escritas através de instruções curtas e simples.

- *Realização do Teste*

Após a definição das tarefas e a escolha dos utilizadores, procede-se à realização do teste e a uma cuidada observação dos procedimentos levados a cabo por estes. A informação recolhida da observação será usada para efectuar uma revisão do protótipo, após a qual se deve realizar, preferencialmente, uma segunda observação para validar as alterações introduzidas na interface. Este processo iterativo da criação de protótipos e observação de utilizadores pode ser continuado até que os objectivos inicialmente definidos sejam atingidos.

A Apple em [Apple92] define dez passos para a realização do teste e observação dos utilizadores:

1. **Descrição, em termos gerais, da finalidade da observação. Deve-se evitar mencionar que estaremos a observá-los.**

Algumas das apresentações possíveis:

- Vão-nos ajudar a testar este produto.
- Se tiverem dificuldade em alguma das tarefas é problema do produto e não vosso. É exactamente esses casos que pretendemos encontrar.

2. **Dizer aos participantes que podem desistir em qualquer altura.**

Este é um ponto importante. Devemos assegurar-nos que informamos os participantes que podem sair em qualquer altura se, por qualquer motivo, não se sentirem à vontade.

3. **Explique a finalidade do equipamento da sala.**

Deve explicar-se qual a finalidade de cada equipamento instalado na sala de observação como *hardware*, *software*, câmara de vídeo, gravador, microfones, etc.

4. **Explique aos utilizadores que devem pensar em voz alta.**

Deve pedir-se aos utilizadores que pensem em voz alta nos passos que vêm à ideia enquanto trabalham dando uma explicação do facto pelo qual pretendemos que assim ajam. Ao ouvir os utilizadores pensar e planear podemos inferir das suas expectativas, das suas intenções e também das suas estratégias para resolução dos problemas. Este processo é bastante frutuoso em informação para uso posterior.

5. **Explique que não dará qualquer tipo de ajuda.**

Este é o melhor processo para verificar as dificuldades que cada utilizador enfrenta na resolução dos problemas que lhe são apresentados, pelo que se deve, a todo o custo, evitar fornecer qualquer tipo de ajuda. É evidente que haverá situações em que se tornará indispensável a intervenção do observador, mas elas devem previamente ser definidas.

6. **Descreva em termos gerais o que o participante vai efectuar.**

Explicar quais são os materiais que vai usar e a sua sequência de uso. Fornecer ao participante instruções escritas para as tarefas. Se tiver que demonstrar o produto antes da observação deve-se ter o cuidado para não mostrar o que estamos a tentar testar.

7. **Pergunte se os participantes têm alguma questão a colocar antes de iniciar o teste.**

8. **Durante a observação recordar certos aspectos.**

Ficar alerta. É natural começar a perder a atenção depois de algumas horas a analisar utilizadores. Uma grande parte da informação que se pode obter é bastante subtil.

Colocar questões aos participantes. Isto deve ser anteriormente equacionado. As interrupções não devem ser frequentes e sempre que o utilizador hesita ou faz “Humm”, pergunte-lhe o que está a pensar.

Seja paciente. Os utilizadores estão a fazer um favor, provavelmente com algum esforço por parte destes, em situações que provocam um certo nervosismo.

9. **Conclua a observação.**

Fazer o seguinte quando terminar o teste:

- Explicar o que se procurava saber durante o teste.
- Responder a alguma pergunta suplementar.
- Discutir algum comportamento interessante.
- Pedir sugestões aos participantes para melhorar o produto.

10. Use os resultados.

Ao observar os utilizadores com certeza verá comportamentos inesperados, de que não suspeitaria. Quando observa um erro a tentação é atribuí-lo à inexperiência ou falta de inteligência do utilizador. Geralmente não é esse o caso. Quando observa os utilizadores o objectivo é verificar quais as partes do produto que se mostraram difíceis de usar ou ineficazes.

Para tirar o máximo de informação possível dos testes deve-se rever todos os dados obtidos através das notas, vídeo, gravação de som, tarefas, etc., com um certo cuidado. Deve-se tentar encontrar soluções nas partes em que os utilizadores revelaram maiores dificuldades através de novas abordagens da interface.

4.2 Princípios de Concepção

A concepção de uma interface com o utilizador deve seguir um conjunto de princípios que vão desde a escolha da metáfora apropriada para a tarefa em questão até à observância de um conjunto de regras preestabelecidas, largamente estudadas e experimentadas. Neste subcapítulo abordamos estes aspectos.

4.2.1 Metáforas e Analogias

Os modelos conceptuais das interfaces gráficas existentes actualmente baseiam-se, na maior parte dos casos, em analogias com o meio físico que envolve o utilizador, de forma a induzir neste algum conhecimento *à priori* do sistema. Classificadas como metáforas estas analogias não são mais do que tentativas de representar ideias abstractas sob a forma de conceitos concretos.

As metáforas não se restringem à poesia ou aos “romances de cordel”, existindo na linguagem do dia a dia e inclusive no pensamento.

[Erickson90] diz, inclusivamente, que a metáfora é parte tão integrante do nosso discurso e pensamento que chega a ser invisível. Uma palavra que é usada como metáfora é somente a ponta do *iceberg*. A metáfora funciona como um modelo natural, permitindo que façamos uso do nosso conhecimento das experiências e dos objectos concretos e familiares, para com ele construir a estrutura de conceitos abstractos.

Devem ser implementadas metáforas e analogias concretas que correspondam ao ambiente especial em causa, para que os utilizadores possam desenvolver uma série de expectativas acerca da funcionalidade do sistema.

Como um dos ambiente mais vulgares no uso dos computadores foi, durante muito tempo, o de escritório, tornava-se natural que a primeira tentativa de uso de metáforas assentasse

neste ambiente. O projecto LISA e o projecto Xerox PARC foram pioneiros no uso da metáfora *desktop*, que reflectia o ambiente de secretária. A Apple Macintosh aproveitou este conceito e desenvolveu a interface gráfica historicamente mais importante, que fazia uso de elementos como janelas, ícones, pastas, ficheiros, caixote do lixo, de tal forma popular que viria a ser também imitada pela Microsoft (Windows), NEXT (NeXT Step), Commodore (Amiga) e Digital Research (GEM).

Nem todos os utilizadores no entanto estão familiarizados com o ambiente de secretária, facto que se pronunciou nos últimos anos com a globalização do uso dos sistemas informáticos à população em geral. Nestes casos seria melhor optar por desenvolver uma metáfora baseada em conhecimentos mais vastos do utilizador comum, como *cozinha* ou *casa* [Santos95], ou *imprensa* ou *televisão* [Erickson90] ou *teatro* com os seus actores, cada um com o seu papel (agentes?) [Negroponte90]. Jan Gulliksen, em [Gulliksen95c], introduz-nos na ideia do uso da metáfora baseada em quartos, que consiste em fornecer ao utilizador zonas de trabalho específicas (quartos), cada um com portas para os outros quartos disponíveis. Cada quarto terá também a porta das traseiras, que nos leva ao último quarto (zona de trabalho) visitado. O uso desta metáfora, segundo Gulliksen, permite compensar problemas como: o tamanho limitado dos monitores, o problema da navegação, saltar entre tarefas e problemas de acesso à informação.

Num sistema domótico é aconselhável fazer algum esforço para encontrar as metáforas apropriadas a esse ambiente. Metáforas do tipo da introduzida por Gulliksen seriam mais apropriadas a este tipo de sistemas.

Num capítulo posterior faremos referência a uma nova abordagem na concepção de Interface Gráficas com Utilizador, em desenvolvimento pelas grandes empresas informáticas e por algumas pequenas empresas pioneiras, centrada na informação. Como veremos aí, o conceito de desktop é de certa forma esbatido.

No sentido de avaliar a utilidade das metáforas de interface gráfico, [Erickson90] define cinco questões fundamentais:

- *A metáfora possui estrutura suficiente?* a utilidade de uma metáfora é proporcional à estrutura da mesma.
- *A estrutura tem aplicabilidade?* quanto da estrutura é relevante para o problema e quanto é irrelevante conduzindo o utilizador para um caminho errado.
- *A metáfora é facilmente representável?* metáforas ideais possuem representações visuais e auditivas distinguíveis assim como palavras específicas associadas.
- *A metáfora é adequada à audiência alvo?* mesmo que a metáfora possua todas as outras características, pode tornar-se inútil se não for compreendida pelo utilizador. Gulliksen apresenta as características de interfaces orientadas para o domínio de aplicação [Gulliksen95a] e [Gulliksen95b].

- *A metáfora é evolutiva?* a metáfora pode conter alguns “bits” estruturais possíveis de serem usados posteriormente.

Serão as metáforas um meio indicado para o desenvolvimento de interfaces com o utilizador?

Apesar de ser mais ou menos universalmente aceite a necessidade da implementação de metáforas na interface com o utilizador, existem algumas opiniões divergentes que colocam em questão a sua utilidade, afirmando que assim se obriga o utilizador a ficar dependente de uma dada metáfora, tornando-se esta um peso morto [Nelson90].

Alan Cooper, citado em [Marcos95], defende que procurar a metáfora ideal é um dos maiores erros que podemos cometer quando projectamos uma interface com o utilizador. Algumas das razões apontadas são as seguintes:

- Uma metáfora que funciona bem para um processo simples geralmente falha quando o processo cresce em tamanho e complexidade. Veja-se o caso dos ícones de ficheiros, que para um disco de pequena capacidade (quando a metáfora *desktop* foi desenvolvida os discos rígidos dos PC's possuíam capacidades na ordem dos 10, 20 Mb, o que para a altura era suficiente) funciona bem, mas para discos actuais de vários gigabytes, que armazenam milhares e milhares de ficheiros, esta representação torna-se pouco apropriada.
- O sucesso do projecto da Xerox PARC, e posteriormente da Apple, baseia-se principalmente no facto de o utilizador não ter que memorizar os comandos relativos a cada acção que pretende e poder escolhê-los com um conjunto de acções do rato.
- Quando se começa a tentar encontrar metáforas para representar conceitos, em vez de objectos concretos (impressoras, monitor, folha de papel), a probabilidade de o utilizador as reconhecer baixa consideravelmente.
- As metáforas restringem o nosso pensamento, limitando o poder do *software*.

Alguns autores referem o ícone do caixote do lixo da GUI da Apple Macintosh como uma metáfora usada indevidamente. Todos já tivemos ocasião de ver ou usar um PC Apple Macintosh e sabemos que para ejectar uma disquete basta arrastar o ícone correspondente a esta para o ícone do caixote do lixo. Ora, na vida real, o caixote do lixo serve para deitar fora objectos que não desejamos guardar, o que não corresponde na realidade à acção agora referida. Talvez tivesse sido preferível encontrar outra forma de representar a ejeção de uma disquete.

[Brock96] refere também que, mais importante do que encontrar a metáfora ideal, o que é realmente necessário é fornecer aos utilizadores, de uma forma expedita, instruções simples que lhe permitam entender o sistema e a forma como está estruturado. Refere como exemplo a história de uma senhora que, depois de ter desempacotado o seu computador, telefonou para o serviço de assistência técnica pedindo ajuda porque não conseguia por a funcionar o pedal que vinha com o equipamento (tinha confundido o rato com o pedal de

uma máquina de coser). O facto pode explicar-se por não lhe terem sido dadas quaisquer instruções no sentido de ela conhecer a funcionalidade de tal dispositivo.

4.2.2 Regras Gerais de Concepção

Desde os princípios da utilização de interfaces gráficas para efectuar a interligação homem-máquina que as instituições, os responsáveis pela concepção das interfaces, os projectistas, os académicos e os investigadores em geral efectuaram pesquisas sobre quais as regras fundamentais para produzir interfaces com um elevado grau de usabilidade. Além das regras gerais relativas à apresentação da informação, referidas com algum rigor no capítulo 3.3, existem actualmente um número infindável de regras elaboradas por empresas como a Apple [Apple92], a Sun Microsystems [OPEN LOOK90], a Microsoft, a Xerox, por instituições governamentais como a NASA [DSTL-95-003], o departamento de defesa dos E.U.A. [MIL-STD-1472C] ou a divisão de sistemas electrónicos da força aérea [Smith86], por equipas de investigadores ou como resultado de projectos do tipo ESPRIT ou TRIDENT [Vanderdonck93].

Não se pretende, aqui, fazer uma exposição exaustiva das regras existentes, mas sim um apanhado das regras gerais de concepção definidas em cada um dos documentos (que de uma forma geral são bastante semelhantes) e que são mais orientadas para o desenvolvimento de uma interface com um sistema domótico. Os exemplos que se apresentam são consequência dessa orientação.

4.2.2.1 Conformidade com as expectativas do utilizador

- ***Consistência***

A consistência é talvez um dos principais factores a observar. A consistência na interface permite ao utilizador transferir conhecimento e aptidão de uma aplicação para outra ou dentro da mesma aplicação.

Componentes similares devem operar similarmente no mesmo ou em diferentes dispositivos; a mesma acção deve ter sempre o mesmo resultado.

Podemos falar de consistência na interface visual, que ajuda os utilizadores a aprender e a reconhecer com facilidade a linguagem gráfica da interface, e consistência no comportamento da interface, que permite ao utilizador ter que aprender somente uma vez procedimentos como apontar ou premir o botão do rato ou arrastar e largar ícones [Apple92]. Para verificar a consistência da interface, [Mahajan95] propõe uma série de ferramentas de verificação da consistência, evitando o mais possível, desta forma, o recurso a testes de usabilidade.

Um dos aspectos importantes na consistência é mantê-la nas várias versões de uma

determinada aplicação. [Tognazzini90] refere que, se for necessário efectuar grandes alterações numa aplicação, se deve observar os seguintes pontos:

- Seguir as regras sempre que possível.
- Não efectuar mudanças excepto se tal for imprescindível.
- Se tiver que efectuar uma mudança fazê-la grande e óbvia.
- Adicionar novas ferramentas em vez de alterar a funcionalidade das ferramentas já existentes.
- A interpretação consistente do comportamento do utilizador por parte do sistema é mais importante do que a consistência de objectos ou apresentação.

Exemplo:

- Os eventos e parâmetros para programação temporal em diferentes dispositivos que integram um sistema domótico devem ser consistentes na mesma ordem. A entrada da hora ou data deve ser feita da mesma forma tanto no painel frontal do dispositivo como no comando à distância, no ecrã da televisão ou no monitor do PC.

• *Estabilidade na percepção*

Deve ser definida uma quantidade limitada de objectos e acções. Um produto deve ter pontos de referência para o utilizador. A mesma acção deve ter o mesmo ícone ou instrução associado. O mesmo botão deve situar-se mais ou menos na mesma zona entre produtos. Uma pessoa que tem um sistema de alta fidelidade por módulos, da mesma marca, e que os quer desligar todos, começa por desligar o amplificador através do botão que se situa no lado superior esquerdo, avança para o leitor de cassetes e desliga-o através do botão que fica situado no lado superior esquerdo e quando segue para o último módulo, o leitor de CD's, ao aproximar o dedo do canto superior esquerdo só nessa altura se lembra que afinal naquele módulo o botão para o desligar se situa no canto superior direito. Ora isto não é razoável, ainda por cima em produtos do mesmo fabricante. O projectista só se preocupou em “arrumar” os botões de forma mais ou menos estética sem ter em atenção a estabilidade na percepção.

Exemplo:

- Botões com a mesma funcionalidade básica, como o *Play* num leitor de cassetes, num video-gravador ou num leitor de CD, devem ter o mesmo ícone.

• *Continuidade entre produtos*

Deve ser possível que o utilizador aplique às novas áreas conhecimentos, anteriores

sem ter que aprender por tentativas.

Exemplo:

- Quando o utilizador evolui de um VCR sem *On Screen Display* para um outro que o possua, devem ser usados os mesmos botões no painel frontal e no *On Screen Display*

- ***Apelo estético***

O projectista deve usar as normas recentes de representação da informação; a aparência do interface de um produto pode afectar significativamente a atitude do utilizador perante o interface. Neste aspecto é importante o uso de ícones bem desenhados, a utilização da cor de forma cuidada e a disposição dos vários objectos de interacção segundo as regras abordadas no capítulo sobre apresentação da informação.

- ***Organização gráfica***

Uma organização gráfica bem estruturada é útil para que o utilizador identifique com facilidade as diferentes partes do interface, os grupos de itens, para perceber a relação entre eles. Para além de fornecer uma aparência mais agradável, o agrupamento ajuda a acção de memorizar, o que resulta numa pesquisa mais rápida no monitor. Existem várias maneiras de efectuar o agrupamento ou distinção de itens, já abordados em capítulos anteriores:

- algumas características gráficas diferentes dos itens (forma, cor, etc.)
- arranjo espacial dos itens (alinhamento, proximidade ou separação, etc.)
- definir fronteiras pela utilização de caixilhos ou molduras, etc.

Exemplo:

- Tentar definir grupos como áreas do monitor correspondentes ao ângulo visual de 5° em relação ao qual o olho humano é mais sensível. O tamanho do grupo dependerá da distância ao monitor: 40 cm (3,5 cm); 50 cm (4,4 cm); 60 cm (5,2 cm), etc.

4.2.2.2 Tolerância ao erro e prevenção do erro

- ***Filosofia da não existência de erro do utilizador***

Quando ocorre um erro isso, em princípio, significa que o utilizador efectuou uma acção que não estava prevista nesse momento. Isto acontece se o utilizador não percebe o que é pedido pelo sistema ou quando ele é usado de uma forma inadequada. Nestes casos o sistema deve proporcionar elementos informativos que permitam ao

utilizador perceber o que se está a passar, evitando que ele desenvolva sentimentos de frustração ou incompetência. Qualquer mensagem de erro deve orientar o utilizador na correcção desse mesmo erro dando-lhe, inclusivamente, algumas pistas para a sua resolução. Como medida preventiva adicional, o sistema deve ainda prevenir a entrada incorrecta de dados por parte do utilizador.

Exemplos:

- Se for usado indicador acústico, este não deve ser embaraçoso.
- As mensagens de erro devem indicar ao utilizador o que deve ser feito para terminar a tarefa, tanto quanto possível, e não indicar somente o que está mal (Figura 36).

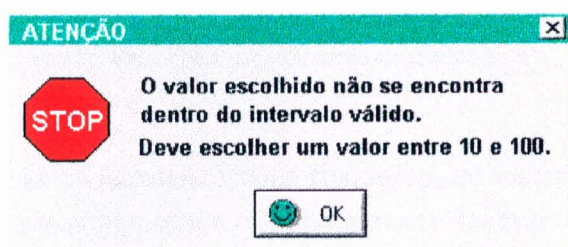


Figura 36 - Exemplo de uma mensagem de erro.

• Acções reversíveis

As pessoas devem sentir que podem tentar novas acções sem danificar o sistema ou bloquear o seu funcionamento [Apple92].

Os projectistas devem fazer com que as acções dos utilizadores sejam reversíveis sempre que possível. Deve-se informar o utilizador, sempre que uma dada acção seja irreversível e que a sua efectivação possa originar situações eventualmente indesejáveis. Nestes casos deve ser mostrada uma mensagem de aviso. Isto suporta o comportamento tentativa/erro com o qual muitos utilizadores estão acostumados, que facilita a aprendizagem de um sistema e que permite a correcção imediata quando uma acção é inadvertidamente iniciada.

Exemplos:

- Deve ser possível apagar uma programação temporal.
- Sempre que um programa foi iniciado num dispositivo, p.e. numa máquina de lavar roupa ou louça, o utilizador deve ter a facilidade de interromper ou parar o programa.
- Quando é necessário introduzir uma longa sequência de dados, o utilizador deve poder corrigir os erros em qualquer altura e não ter que repetir a sequência completa.

- ***Protecção contra erros destrutivos***

Os erros destrutivos são aqueles que provocam alteração definitiva em certos elementos de um dispositivo. Mudar, por exemplo, o comutador “110-220V” para 110V e ligar o equipamento a uma tensão de 220V pode provocar danos irreversíveis.

O dispositivo deve ser capaz de prevenir e detectar erros, especialmente se esses erros podem ter consequências destrutivas. Se for possível não se deve deixar o utilizador entrar numa sequência que conduza a resultados destrutivos.

Exemplo:

- Em certos dispositivos existem acções que servem para modificar alguns parâmetros importantes e que são usados muito raramente. Estes comandos devem estar escondidos, de forma a que o utilizador aja de forma explícita para aceder a eles: o utilizador deve abrir uma caixa situada atrás do dispositivo.

Não se devem mostrar os comandos mais frequentes ao mesmo nível e de uma forma similar que aqueles menos frequentes e eventualmente destrutivos.

Se um comando deve ser usado frequentemente mas pode conduzir a resultados destrutivos, isso deve ser indicado claramente ao utilizador:

- providenciar um código específico (cor, ícone, símbolo) para indicar os comandos críticos.
- se uma acção provocar uma alteração não reversível, e a alteração for crítica, deve ser pedido ao utilizador que confirme a alteração.

4.2.2.3 Controlabilidade

Permitir que seja o utilizador e não o sistema a iniciar e controlar as acções. As pessoas aprendem melhor quando estão envolvidas de uma forma mais directa [Apple92].

Sistemas mal concebidos requerem que o utilizador reaja ao computador em vez de permitir que o utilizador peça ao computador para realizar as tarefas que deseja. Todos os utilizadores vão desejar efectuar as coisas à sua maneira, de forma que as tarefas se tornem mais atractivas e menos fastidiosas e aborrecidas.

O utilizador deve decidir sempre o próximo passo e deve sentir que possui o controlo do diálogo.

- ***Acções iniciadas pelo utilizador***

Devem estar disponíveis as mais amplas escolhas para que o utilizador possa decidir o

que fazer a seguir e como resolver uma determinada tarefa.

Alguma orientação pode ser fornecida ao utilizador (p.e. posicionamento do cursor), mas ele deve ter sempre a possibilidade de alterar ou cancelar essas sequências.

O sistema deve processar somente as acções explicitamente requeridas pelo utilizador e só quando lhe é indicado para o fazer. Não iniciar acções derivadas de efeitos colaterais de outras acções.

- ***Atenção para os modos***

Um modo é definido como qualquer contexto no qual uma acção do utilizador é interpretada de forma diferente que a mesma acção noutra contexto.

No sentido de proteger o utilizador de erros derivados do uso de modos, o sistema deve usá-los apenas quando for estritamente necessário e em número tão reduzido quanto possível.

Exemplo:

- Seleccionar um número de telefone de uma agenda telefónica electrónica deve ser sempre possível e não somente quando o auscultador do telefone está levantado. O utilizador pode querer saber o número de telefone de uma determinada pessoa sem ter que levantar o auscultador. Assim a acção de selecção de um número de telefone não fica dependente de um determinado estado do sistema.

Quando haja necessidade de usar modos o utilizador deve sempre ser claramente informado do modo em que se encontra.

Exemplo:

- Funções de um menu que não sejam seleccionáveis no modo corrente devem ser indicadas como não-seleccionáveis (p.e. mostrar esses itens em cinzento).

Existem situações em que o uso de modos se torna imprescindível, como nos casos em que é necessário fornecer ao utilizador alguma informação que necessite de ser confirmada antes de continuar. Nestes casos deve ser mostrada uma mensagem modal e não permitir que o utilizador faça qualquer coisa que não seja a confirmação de que tomou conhecimento da situação.

4.2.2.4 Possibilidade de individualização

Segundo [Shneiderman92] o desenvolvimento de uma interface deve considerar três tipos de utilizadores de acordo com a sua experiência:

- **utilizador principiante**: não tem conhecimento sintáctico da aplicação e apenas algum conhecimento semântico.
- **utilizador ocasional**: mantém o conhecimento semântico ao longo do tempo mas vai perdendo o conhecimento sintáctico adquirido.
- **utilizador frequente**: também chamado utilizador poderoso, tem conhecimento sintáctico e semântico do sistema.

Cada um destes utilizadores tem necessidades diferentes: o utilizador principiante necessita de clareza e simplicidade, assim como de mensagens elucidativas e de extensos manuais, o utilizador ocasional necessita de manuais concisos e de comandos simples e fáceis de memorizar, e o utilizador frequente necessita de interacção rápida, mensagens breves e de elevada customização.

O desafio da concepção de uma interface é conseguir satisfazer todas estas necessidades numa única interface.

- **Diferentes níveis de dificuldade**

Funções mais sofisticadas e menos frequentes devem ser escondidas em níveis menos imediatos, devendo contudo continuar acessíveis.

Um utilizador principiante necessita de funções mais básicas, usadas com maior regularidade, e por conseguinte é natural que essas estejam num nível mais imediato. Não parece conveniente que devamos colocar as funções mais frequentes num nível menos imediato.

No entanto, à medida que o utilizador vai ganhando novas competências, e o seu nível de experiência vai evoluindo, as suas necessidades vão-se alterando e as funções que antes seriam menos frequentes podem tornar-se mais utilizadas. O sistema deve permitir, também, que essas funções possam ser deslocadas para níveis mais próximos e mais imediatos.

- **Teclas de acesso rápido: atalhos**

Uma aplicação eficiente deve fornecer os meios para minimizar o número de operações necessárias para realizar uma dada operação.

Quando se constrói uma interface ela é destinada a vários tipos de utilizadores, desde o

utilizador “novato” até ao utilizador mais experiente. O utilizador novato irá concertiza efectuar acções sobre um determinado objecto premindo o indicador do rato sobre um ícone que representa a acção a realizar ou até escolher uma das opções do menu existente, opção essa que pode estar num nível primeiro ou em níveis seguintes. Este processo, para o utilizador experiente, torna-se bastante moroso e até algo fastidioso. Por este motivo se atribui a possibilidade de o utilizador experiente usar as chamadas teclas de acesso rápido (atalhos) que não são mais que uma combinação de teclas que dão acesso directo a uma determinada função. As interfaces mais recentes permitem, inclusivamente, ir mais longe e dão ao utilizador experiente a possibilidade de efectuar uma configuração à sua escolha da combinação de teclas que achar mais convenientes para cada uma das acções.

- **Customização**

Um produto deve permitir aos utilizadores a configuração do estado do sistema e a selecção de preferências.

Deve-se, no entanto, tomar algumas precauções no sentido de não permitir que o utilizador configure de tal forma o sistema que ele passe a infringir regras básicas de apresentação da informação, de atribuição de cores ou sons, de alteração das mensagens, etc.

Exemplo:

- O utilizador deve poder escolher o ícone referente a cada uma das dependências da habitação.

4.2.2.5 Auto descrição / Aprendizagem

Um diálogo deve ser auto descritivo e de simples aprendizagem. Para se conseguir estes requisitos devem ser seguidas regras gerais que se apresentam nas secções seguintes e que, se forem seguidas, conduzirão a uma interface mais usável.

- **Simplicidade e clareza**

Os sistemas interactivos actuais, por serem demasiado complexos, resultam na maior parte dos casos em interfaces também bastante complexas.

O monitor não deve estar demasiado cheio de objectos e itens informativos. *Designs* simples são de mais fácil aprendizagem e uso, dando ao interface um aspecto mais consistente. Não se deve, no entanto, levar esta consideração ao extremo de tornar a interface desinteressante e com poucas potencialidades.

Exemplo:

- Usar ícones de reduzida complexidade visual ajuda o utilizador no seu reconhecimento.

- ***Linguagem simples***

Nas expressões usadas nos rótulos ou nas mensagens textuais, para fornecer ao utilizador informação acerca da acção a tomar, deve ser usada uma linguagem directa, sem ambiguidades.

- ***Recurso mínimo à memória***

A informação assimilada pelo utilizador é processada e copiada para a sua memória de curta duração. É aqui que se realizam as funções ligadas ao raciocínio. Esta memória tem a capacidade aproximada de sete itens e a informação nela contida desaparece ao fim de cerca de 20 segundos. É fácil de compreender que a informação relativa ao sistema não se encontra na memória de curta duração, devendo pelo contrário a interface ser elaborada no sentido de esta memória ser minimamente usada. As interfaces devem exigir que o utilizador se lembre de uma reduzida quantidade de factos, números ou palavras. As pessoas recordam com maior facilidade estratégias e técnicas para encontrar itens do que se lembram de um item em particular.

Não deverá ser necessário ao utilizador recorrer à memória quando o produto o pode fazer por ele. Como as pessoas são mais eficazes no reconhecer que no recordar, o produto deve apresentar alternativas e deixar que o utilizador escolha entre elas. Deve manter também o estado escolhido pelo utilizador, deixando sempre a possibilidade de reverter para a situação definida por omissão.

Exemplo:

- Para a selecção de um determinado canal de televisão pode ser usado o nome do canal em vez de um número.

- ***Resposta imediata***

Com o fim de atingir os seus objectivos as pessoas agem da seguinte forma:

- recolhem informação do estado actual,
- realizam uma acção,
- analisam o resultado da acção e verificam se atingiram ou não o seu objectivo.

Sendo assim, a resposta à acção do utilizador deve ser imediatamente observável. O

feedback efectivo é uma das características que têm maior impacto no utilizador [Marcos95]. A resposta imediata permite ao utilizador avaliar se os resultados são os esperados.

Quando o utilizador selecciona um determinado ícone ele deve mudar a sua aparência de forma a indicar que foi pressionado.

A ausência de resposta provoca ansiedade nos utilizadores. Sempre que o sistema necessite de efectuar processos extremamente lentos ou demorados, ou que a acção a realizar obrigue o sistema a procurar a informação noutra local geograficamente diferente, com um grande atraso na comunicação, deve ser dada indicação desse facto ao utilizador de forma a que ele não pense que algo de errado se passa e, devido à sua impaciência inata, aborte a acção. Este facto é observado frequentemente nas ferramentas de pesquisa na Internet. Como a transmissão sofre de um atraso considerável devido à estreita largura de banda para o número de utilizadores existentes, qualquer acção efectuada tem uma resposta que por vezes demora alguns minutos a completar-se, dependendo do tipo de dados pedido, sem, no entanto, ser dada ao utilizador uma informação directa de que o seu pedido foi aceite e que o sistema está a efectuar a transmissão dos dados (é certo que na barra de estados é fornecida alguma informação, mas ela só tem algum significado para um utilizador experiente; o indicador do rato muda para a ampulheta mas não é indicado se a acção é realizável ou não).

- ***Sugestão ao utilizador***

Devem ser indicadas ao utilizador as acções possíveis ou esperadas num determinado estado do sistema, ou mesmo ser guiado na introdução dos dados necessários à operação a realizar. Esta indicação ao utilizador conduz a uma utilização mais produtiva já que diminui a quantidade de erros e a quantidade de operações necessárias para alcançar um determinado objectivo.

Exemplo:

- Nas entradas longas de informação, o utilizador pode ser guiado pela existência do cursor e seu posicionamento automático.
- Quando é necessário introduzir a data de um determinado evento pode ser indicada, automaticamente, no campo de entrada o valor da data actual. O mesmo com a hora ou uma determinada temperatura, considerada ideal em termos de conforto.

- ***Terminologia orientada ao utilizador***

O vocabulário e os códigos usados na interface devem ser compreensíveis para o utilizador, de forma a facilitar a aprendizagem e a memorização. “Compreensíveis” não

significa necessariamente a utilização de termos usados em equipamentos já existentes. Os novos códigos e vocabulário devem ser escolhidos com muito cuidado e sempre dentro da perspectiva do utilizador [FACE94]. Este facto é válido tanto para os termos usados (verbos, rótulos, comandos, etc.), como para as abreviaturas e códigos utilizados (ícones, símbolos, sinais, etc.).

A terminologia deve ser familiar para o utilizador, como se fosse usada no seu dia-a-dia. A terminologia técnica deve ser evitada.

Exemplos:

- Como as palavras curtas são as mais usadas no dia-a-dia são, por isso, mais familiares e de fácil compreensão.
- Havendo alternativa devem-se evitar palavras que contenham sufixos ou prefixos.
- Usar expressões afirmativas ou positivas pois são bastante mais simples de memorizar.

4.2.2.6 Documentação

Um dos requisitos fundamentais no desenvolvimento de interfaces gráficas é fornecer ao futuro utilizador a documentação que torne a interface inteligível. A documentação deve ser considerada como uma parte integrante da interface global com o utilizador uma vez que ele pode usá-la como um recurso para realizar as tarefas pretendidas.

A documentação é entendida como os manuais, o *help* (ajuda *on-line*), programas de treino e apoio ao utilizador. Todos estes componentes ajudam o utilizador a compreender a interface e a estabelecer um modelo conceptual apropriado.

Apesar de ser discutível a utilização de manuais (existem autores que afirmam que os utilizadores não os lêem) torna-se cada vez mais importante o seu uso já que a complexidade das aplicações é cada vez maior e seria praticamente impossível memorizar todas as sequências possíveis para efectuar os vários tipos de tarefas. A fonte à qual recorrem varia de caso para caso: pode ser o manual, o administrador do sistema ou uma linha telefónica de suporte ao cliente.

- ***Manuais***

O manual continua a ser uma boa fonte de informação para muitos utilizadores. Um bom manual é, regra geral, a extensão mais importante da interface propriamente dita.

Seguem-se algumas regras básicas que devem ser observadas aquando da escrita de um manual do utilizador, para que este seja eficaz [Santos95]:

- Ser sucinto - A brevidade é um ponto fulcral da escrita de um manual. O utilizador

geralmente não encontra a informação que necessita nos manuais demasiado longos pois está disseminada numa grande quantidade de texto irrelevante.

- Fazer uma apresentação geral e uma sequência de passos para cada tarefa - A apresentação geral deve ser simples e a apresentação da tarefa deve ser simples mas real.
- Usar a linguagem do utilizador - Deve ser tido em conta que se está a escrever um “manual do utilizador” e não um manual técnico para peritos na concepção de interfaces gráficas.
- A apresentação deve ser atractiva e funcional.

• *Help*

A ajuda fornecida no próprio sistema deve ser adaptada ao tipo de interface usado. Os utilizadores normalmente apresentam queixas sobre a forma como a ajuda está implementada [Sellen90]:

- Dificuldade em encontrar a informação pretendida
- Incapacidade de apresentação da informação relevante
- Dificuldade em alternar entre a ajuda e o contexto de trabalho
- A complexidade da interface de ajuda
- A qualidade e o *layout* da informação de ajuda

Deve ser feito um esforço para tentar fornecer ao utilizador a ajuda que necessita de acordo com a dificuldade encontrada. O contexto em que o sistema se encontra deve apontar ou servir de indicação para o tipo de ajuda prestada. O tempo de pesquisa pode ser reduzido se o tópico de ajuda corresponder à questão esperada [Sellen90].

Na ajuda *on-line* não devem ser usados textos demasiado longos. A sua ineficácia é uma conjugação de diversos factores: texto no ecrã é geralmente menos legível que no papel; a probabilidade de as pessoas se perderem a “navegar” na ajuda é maior no ecrã que nas folhas do papel onde existe uma orientação física; os ecrãs não são actualizados tão rapidamente quanto o virar das páginas; a janela da ajuda sobrepõem-se frequentemente à parte da interface sobre a qual o utilizador tem dúvidas; e as pessoas não praticaram a leitura de texto no ecrã tanto como o fizeram no papel.

A ajuda no ecrã é o local por excelência para colocar factos breves, que os utilizadores necessitam frequentemente ou que é provável que esqueçam.

- **Treino**

O treino engloba os cursos, os *tutorials*, *on-line* ou não, que o utilizador pode fazer sozinho. Estes devem abordar as tarefas representativas da interface gráfica. Ao contrário do manual, o treino deve forçar o utilizador a envolver-se activamente no funcionamento do sistema. Uma versão mínima do manual pode servir como ponto de partida para efectuar um programa de treino.

A ideia do manual mínimo vai mais longe no conceito de brevidade do manual básico; o manual mínimo é intencionalmente incompleto. Neste manual a falta de detalhe é cuidadosamente associada a tarefas com objectivos claros, forçando o utilizador a investigar a interface para descobrir como funciona. Esta abordagem resulta em tempos de aprendizagem consideravelmente mais reduzidos que o treino tradicional completamente orientado.

- **Apoio ao utilizador**

O apoio ao utilizador traduz-se quase sempre em linhas telefónicas ou correio electrónico, operados por técnicos especialistas que fornecem ao utilizador a ajuda pretendida. Esta comunicação entre o técnico especialista e o comum utilizador pode revelar-se deveras problemática, já que a linguagem usada pelos dois será certamente bastante diferente e de difícil entendimento por parte do interlocutor, agravado pelo facto de a conversa ser efectuada pelo telefone ou correio electrónico em vez de pessoalmente.

No sentido de minimizar os problemas inerentes a este tipo de comunicação uma das regras fundamentais é, novamente e sempre que possível, o uso da linguagem do utilizador.

4.2.3 *Desenho de ícones*

Ícones são representações gráficas de objectos como documentos, impressoras, aplicações, dispositivos de armazenagem de informação, etc.

As pessoas reconhecem e compreendem com maior facilidade a representação de certas coisas do que a sua explicação verbal. Como é costume dizer-se “uma imagem vale mais que mil palavras”. Estudos mostram que os sinais de trânsito que contêm símbolos são mais facilmente reconhecíveis a uma certa distância que aqueles que só têm palavras [Apple92]. Além disso os símbolos usados nos ícones ocupam menos espaço que as palavras necessárias para os definir.

A utilização de ícones é, actualmente, fortemente utilizada em zonas de acesso público como estações de comboio e metro, correios, aeroportos e centros comerciais, por forma a

orientar o utilizador com uma linguagem perfeitamente acessível. Esta linguagem permite, também, ultrapassar barreiras culturais e de analfabetismo. São sobejamente conhecidos os símbolos referentes, por exemplo, a casas de banho, deficientes motores, escadas, saída de emergência.

A aplicação destes ícones em sistemas para habitação é também fortemente recomendável pelas seguintes razões:

- Aproveitamento da capacidade humana de reconhecimento de padrões.
- Independência face à língua.
- Redução no raciocínio mental exigível.
- A informação por ícones ocupa menos espaço.

Neste tipo de sistemas a escolha dos ícones torna-se mais simples porque na maior parte dos casos eles representam objectos da vida real, electrodomésticos, aparelhos, luzes, aquecedores, alarmes, e não conceitos, que são mais difíceis de representar.

O desenho de ícones deve seguir um conjunto de regras empregues em diferentes níveis de avaliação:

- ***Representação das principais características do objecto.***

Um ícone deve ser típico para um objecto, uma acção ou uma noção abstracta. Deve mostrar as características relevantes que caibam dentro do modelo mental do utilizador. Devem ser aplicadas as leis sobre agrupamento de objectos.

- ***Suporte para reconhecimento espontâneo***

O ícone deve ser suficientemente significativo para que um utilizador inexperiente o possa reconhecer, dentro do ambiente próprio ou da tarefa a executar.

- ***Redução da complexidade visual***

Os ícones devem ter a menor complexidade visual possível, reduzindo assim a densidade de informação para a estritamente necessária ao reconhecimento inequívoco dos objectos. Existe um compromisso entre o reconhecimento inequívoco de um objecto e o uso da menor quantidade de informação possível.

- ***Distinção***

Os ícones têm que ser avaliados dentro de um certo contexto. O seu significado pode variar bastante conforme os ícones adjacentes usados.

- ***Consistência no grau de abstracção e no estilo de design***

O grau de abstracção deve ser uniforme para todos os ícones usados no mesmo dispositivo. O estilo de *design* deve, também, ser o mesmo para todos os ícones para

que assim haja um maior apelo estético para o produto.

Além destas regras convém referir que a Apple aconselha a começar por desenhar o ícone de maior dimensão (no Macintosh existem vários tamanhos do mesmo ícone, 32x32, 16x16, 8x8) a preto e branco e só depois atribuir-lhe a cor. As cores a usar devem ser escolhidas de uma *palette* própria para o desenho de ícones. Esta *palette* deve ter um número de cores limitado (36 cores) para que haja uma distinção clara entre elas por parte do observador. Quando se usam ícones tridimensionais não esquecer da consistência da fonte de luz (canto superior esquerdo).

4.2.4 O uso das cores

Tal como a tipografia e o layout, a cor é um utensílio de *design* bastante útil. Como acontece com as outras ferramentas, a cor pode ajudar ou prejudicar o *design*, dependendo da forma como o *designer* a aplica [Shubin96]. Escolher cores ao acaso, sem considerar quantas ou porquê se devem usar, conduz geralmente a um desempenho inferior com o sistema do que quando não se usam cores [Anthony96a].

A cor pode parecer avançar ou recuar, fazer as margens vibrarem ou fundirem-se, esbater ou pronunciar a forma e provocar a alteração das outras cores. Cada combinação de cores pode, potencialmente, provocar afeitos não encontrados nos elementos individuais.

Esta problemática foi estudada e apresentada de uma forma bastante aprofundada por Josef Albers, citado em [Salomon90][Shubin96], no seu livro “*The Interaction of Color*”, onde afirma que a cor é influenciada pela sua localização, colocação e pela forma e tamanho da área que enche. Assim, a aplicação de cor no desenvolvimento da GUI, deve ter em conta o contexto de aplicação e não uma escolha isolada.

Existem vários exemplos dados por Albers para demonstrar estas interações. Mostramos na Figura 37 um desses exemplos.

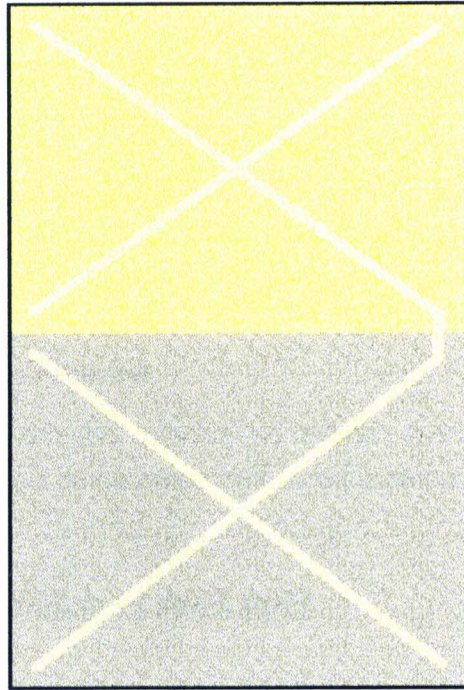


Figura 37 - A aparência das cores varia conforme a sua relação com outras cores. Cada "X" parece ser da cor do fundo contrário, no entanto os dois são da mesma cor.

Os monitores dos computadores, como é do conhecimento geral, variam na sua calibração. Não existe a garantia que uma combinação particular de cores num dado monitor tenha o mesmo aspecto ou produza o mesmo efeito num outro monitor.

Importante também é a luz ambiente do local onde a aplicação vai ser usada. Conforme a luz seja fluorescente, incandescente ou natural, assim ela é percebida de forma diferente [Salomon90]. Basta lembrarmo-nos do número de vezes em que comprámos um artigo numa loja de pronto a vestir ou de outro tipo pensando que ele é de uma determinada cor e quando o levamos para a luz natural a nossa percepção da cor altera-se de forma considerável levando-nos, por vezes, a rejeitar o produto.

Na transferência de informação entre o sistema e o utilizador o uso de cores tem quatro principais áreas de aplicação:

- ***Codificação da informação nas tarefas de busca.***

As cores podem ser usadas para codificar os elementos de informação de forma a permitirem uma busca mais acelerada. A cor como princípio de codificação é superior à forma ou ao tamanho se usados unidimensionalmente.

- ***Separação de regiões***

A segmentação com o uso de cores produz uma boa organização dos elementos e

assim permite ao utilizador uma maior performance na distinção de regiões no monitor.

- **Orientação do utilizador**

O código de cores permite ao sistema orientar o utilizador nas transições de estado das várias sequências de diálogo.

- **Melhoria na aceitação do sistema**

Segundo [Santos95] a vantagem principal no uso da cor em GUI reside no facto de ela tornar a apresentação mais interessante para os utilizadores.

4.2.4.1 Associação semântica e codificação

A maioria dos utilizadores associa um determinado estado ou sentimento a uma determinada cor. Este facto deve ser aproveitado para o *design* da interface com o utilizador.

A Tabela 3 mostra as associações que a maior parte das pessoas do mundo ocidental faz com cada cor e a sua possível aplicação:

Tabela 3 - Relação entre as cores, a associação efectuada e o seu uso preferencial [FACE94].

Cor	Associação	Uso preferencial
Branco / Preto / Cinzento		Cores de base
Vermelho	Stop, Quente, Perigo	Alarme, Erros, "Stop", Gravação num VCR
Amarelo	Precaução	Mensagens que requerem atenção
Verde	Seguir, "OK"	Normal, Estado perfeito
Azul	Frio, Água	Não ênfase, Sombra: não usado para dados críticos
Magenta	Semelhante ao vermelho	Alarmes secundários, Alerta

4.2.4.2 Codificação redundante com o uso da cor

A cor, sendo percebida de forma diferente das outras características do objecto, pode ser usada para uma informação redundante. Se, num determinado gráfico que mostre a evolução de uma característica crítica de um dado parâmetro, a cor das barras se for alterando conforme o estado do parâmetro vai passando de uma situação para outra (menos

crítica para mais crítica ou vice-versa), isso fornece ao utilizador uma informação mais elaborada, permitindo assim uma melhor percepção por parte deste. Outro exemplo é o sinal de trânsito de “STOP”. A informação de “parar” é dada pela palavra “STOP”, que significa “parar”, assim como pela cor vermelha que significa perigo e parar, na cultura ocidental, e ainda pela forma octogonal do sinal (de tal forma que um condutor identifica o sinal mesmo pela parte de trás). O uso simultâneo dos vários factores de codificação torna este sinal, que é dos mais importantes, mais efectivo.

4.2.4.3 Regras no uso das cores

Antes de fazer qualquer tipo de consideração sobre as regras a usar no uso da cor convém definir certos termos usados para orientar o leitor. Estes termos, as suas definições e exemplos encontram-se na Tabela 4.

Tabela 4- Termos sobre a cor, definições e exemplos [Anthony96a].

Termo	Definição	Exemplo
Tom ou Matiz	Sensação visual que produz a percepção de ser azul, verde, vermelho, etc.	É simplesmente a cor que vemos: azul, vermelho, etc.
Luminância	A quantidade de luz que entra no olho.	A actual medida fisiológica da quantidade de luz que entra no olho.
Brilho	Sensação visual que produz a percepção de mais ou menos luz emitida por uma determinada área. (semelhante à luminância, mas no aspecto qualitativo).	Uma luz azul parecerá mais brilhante que uma luz branca com a mesma luminância. Não existe diferença real mas ela é percebida pelo observador.
Saturação	A pureza da cor numa escala do Cinzento até à variante mais forte da cor percebida.	Um azul concentrado será mais saturado que um “azul céu”.
Luminescência	Sensação visual em que um corpo parece reflectir mais ou menos quantidade de luz.	O-branco reflecte mais luz que o azul.

Existem um sem número de regras para conduzir o projectista a elaborar uma interface de forma eficaz. Apresentamos nas

Tabela 5, Tabela 6 e Tabela 7, um conjunto de regras extraídas de [Anthony96a] [Apple92] e [FACE94].

Tabela 5 - Regras fisiológicas no uso da cor.

Regras Fisiológicas ou Físicas
Não usar conjuntamente cores dos extremos do espectro altamente saturadas.
Não usar o azul para texto, linhas finas e áreas pequenas.
Não usar cores adjacentes que só diferem na quantidade de azul.
Pessoas idosas necessitam de níveis superiores de brilho.
A cor altera-se com a quantidade e qualidade da luz.
A quantidade detectável de mudança numa cor é diferente ao longo do espectro.
Não colocar o vermelho ou verde na periferia de grandes monitores.
Usar cores de baixa saturação em grandes áreas para não provocar desconforto nem imagens gravadas na retina.
Cores opostas conjugam harmoniosamente.
Cores adjacentes devem ter diferentes níveis de brilho.

Tabela 6 - Regras de percepção no uso da cor.

Regras de Percepção
Valores de luminância iguais em diferentes cores nem sempre aparentam o mesmo brilho.
Diferentes tons têm níveis de saturação diferentes.
Nem todas as cores têm a mesma legibilidade.
Utilizar tons mais brilhantes, do centro do espectro ou menos saturados para texto.
Elementos de texto escuros em fundo claro têm melhor leitura que elementos de texto claros em fundo escuro.
Os tons alteram-se conforme as cores de fundo e a sua intensidade.
Não forçar a discriminação de cores em áreas muito pequenas.
Não usar o tom para mostrar informação quantitativa.
Usar altas saturações e grandes intensidades para tons que representem grandes quantidades.
Usar duas cores diferentes em duas grandes áreas provoca uma divisão ou partição do monitor.

Tabela 7 - Regras cognitivas no uso da cor.

Regras Cognitivas
Não usar demasiada cor.
Agrupar elementos relacionados usando uma cor de fundo comum.
Cores similares podem ter significados semelhantes.
O brilho e a saturação atraem a atenção.
Cores quentes ou frias podem indicar níveis de acção.
A mesma cor pode ter diferentes significados em diferentes culturas.
Usar codificação redundante de cor e forma sempre que possível.

Ainda existem regras de carácter geral que providenciam uma orientação para o projectista. A Apple e a Sun, por exemplo, aconselham os projectistas a desenharem inicialmente a sua interface para monitores monocromáticos (preto e branco) e só depois introduzir a cor como elemento suplementar [Apple92] [OPEN LOOK90]. Assim os utilizadores que possuem monitores monocromáticos não ficam impedidos ou até confundidos pelo facto da interface ter sido somente pensada para monitores policromáticos.

Também não se deve usar mais de sete cores diferentes para a codificação de uma variável, nem mostrar mais de quatro cores simultaneamente [FACE94] ou ainda usar um elevado número de cores na nossa interface [Apple92].

Uma das regras menos abordadas e que é mencionada em [Apple92] [Kobara91] é a de usar a simulação de fonte de luz de forma coerente. No *design* para Macintosh e para OSF/Motif, o conselho é usar a simulação de fonte de luz sempre no canto superior esquerdo do monitor.

4.2.4.4 Vantagens e desvantagens no uso da cor

Estudos desenvolvidos por vários autores demonstram algumas diferenças de desempenho na utilização de equipamentos, conforme o uso ou não de cor nas GUI. Estas diferenças situam-se em vários níveis:

- **Precisão**

Segundo [Christ75], a cor deve ser usada como meio de codificação se o utilizador conhecer essa codificação, ou seja, se o utilizador souber que, para realizar uma determinada tarefa, deve procurar um elemento de uma determinada cor. A precisão do utilizador na identificação da cor, nos casos citados anteriormente, é 176% melhor que o tamanho, 202% melhor que a forma e 32% melhor que o brilho. Quando a cor não é

usada de forma conveniente, não é o objectivo da procura, o desempenho baixa para valores negativos: 29% menos que o tamanho, 43% para a forma e 14% para os símbolos alfanuméricos.

- ***Tempo de reacção***

Segundo [Christ75], os tempos de resposta diminuem quando a cor é usada para codificar a informação: o tempo de resposta diminui 40% relativamente ao tamanho, 43% relativamente ao brilho, 63% relativamente à forma e 43% relativamente aos símbolos alfanuméricos. Estes factos são aplicáveis apenas quando a cor é realmente um factor relevante.

A codificação redundante com o uso da cor é mais efectiva que em condições monocromáticas. Quanto maior é a quantidade de informação mostrada no monitor e quanto maior é o nível de ocupação do utilizador mais a cor, comparativamente com os monitores monocromáticos, melhora o desempenho do utilizador [Anthony96b].

- ***Efeitos da prática***

Mesmo com a prática num determinado sistema, apesar dos desempenhos se manterem sempre melhores com a cor do que sem ela, a relação vai diminuindo ao longo do tempo até um valor estacionário [Anthony96b]. O uso da cor também se mostrou útil quando a rotatividade dos utilizadores era elevada [Anthony96b]. Com o uso da cor o tempo de aprendizagem diminui e o utilizador também começa a operar com um maior nível de desempenho.

- ***Processamento cognitivo e memorização***

Estudos feitos por [Christ75] demonstraram que a memória das cores diminui menos rapidamente do que a memória das formas ou tamanhos. Também se verificou que, quando o utilizador está sob tensão, o uso das cores se revela mais útil na ajuda ao utilizador, quando comparado com os monitores monocromáticos, porque mais informação pode ser extraída e processada de forma mais rápida.

Apesar das vantagens no uso das cores serem óbvias na maior parte dos casos, convém ter em atenção alguns aspectos relativos aos equipamentos a utilizar e aos utilizadores finais.

O tipo de equipamento ou orçamento disponíveis pode determinar a obrigatoriedade de usar monitores monocromáticos. Sabemos que os monitores policromáticos são bastante mais caros do que os monocromáticos e por isso este pode ser um factor limitativo para o desenvolvimento da nossa GUI.

Como é do conhecimento geral, existe um número significativo de pessoas (8% de homens e 0.5% de mulheres) que não tem uma percepção muito apurada das diferenças entre determinadas cores (Daltonismo ou outro tipo de disfunção; a mais comum é na dificuldade de distinção entre o vermelho e o verde), logo existe uma preocupação acrescida na sua escolha [Solomon90].

As pessoas idosas também apresentam algumas dificuldades na percepção das cores, especialmente nos tons azuis, e necessitam geralmente de maiores níveis de brilho.

Por forma a não prejudicar os utilizadores menos capazes em termos de distinção das cores, torna-se necessário observar determinados princípios:

- Usar redundância de cor e forma sempre que possível.
- Usar diferenças significativas no brilho das cores a serem distinguidas.

As diferenças culturais são um importante factor a considerar quando a aplicação se destina a uma vasta gama de utilizadores em zonas distintas do mundo. É sabido que as cores têm significado diferente para culturas diferentes. Esta diferença encontra-se mais fortemente entre o ocidente e o oriente. Como exemplo podemos lembrar que o preto representa o luto no ocidente enquanto que no oriente tal situação é representada pelo branco.

4.2.5 O uso do som e voz.

Neste subcapítulo aborda-se o uso do som quer por parte do sistema, como complemento do interface gráfico, quer por parte do utilizador como forma de interacção directa com o sistema (interface de voz).

4.2.5.1 O uso de som

O som é um elemento com o qual vivemos no dia a dia e cuja informação que nos transmite passa por vezes despercebida devido ao hábito do cidadão comum. Quem não pousou o ouvido nos carris da linha de comboio quando era miúdo para saber se ele já vinha perto. O barulho da água a correr para dentro de uma garrafa que se altera à medida que vai enchendo, o barulho do automóvel que se torna diferente consoante se aproxima ou se afasta (efeito Doppler), o barulho de qualquer máquina ou aparelho (se não faz ruído alguma coisa de anormal se passa).

Actualmente o uso do som faz-se sobretudo como complemento da GUI por forma a transmitir ao utilizador uma série de informações. Podemos dividir este uso do som em vários aspectos:

- ***Som como confirmação***

Como dissemos antes, qualquer aparelho faz ruído. O computador não é excepção e o ruído produzido pelos vários componentes pode transmitir-nos confirmação sobre uma determinada acção. Por exemplo, quando mandamos gravar um dado ficheiro numa disquete basta-nos escutar o barulho produzido pela *drive* para ter a confirmação que o nosso comando foi realizado. O mesmo acontece com o disco duro. As teclas do teclado também são ruidosas (produzem um *click*) o que nos confirma que o carácter foi introduzido. Os utilizadores comuns não gostam de teclados extremamente silenciosos.

Assim pode-se usar um determinado som como confirmação que uma dada acção foi realizada pelo sistema. Quando um determinado botão for pressionado, por exemplo, um som afirmativo pode ser dado pelo computador. Este procedimento aplica-se mais usualmente a botões com alguma importância ou que não se alteram quando o utilizador os pressiona. Os sons a usar devem ser diferentes e de acordo com o botão pressionado. Um botão de “OK” deve ter um som “afirmativo” enquanto que um botão de “CANCEL” deve ter um som mais grave.

- ***Som como informação de estado***

Como referimos anteriormente, no caso da garrafa e do automóvel, o estado de uma determinada acção pode ser identificado pelo som que dela advém. Este princípio poderia ser adaptado para os sistemas informáticos de forma a indicar-nos o estado, por exemplo, de uma impressora, se o documento a imprimir estava a chegar ao fim ou se, pelo contrário, ainda estava no início, ou de uma gravação no disco rígido em que o som produzido pelo computador, que deverá ser bastante suave para não incomodar o utilizador, pode fornecer um ideia da fase temporal em que ela se encontra pela diminuição progressiva do volume.

- ***Som como ajuda ao utilizador***

Uma das características do Windows95 é a de permitir ao utilizador com algum tipo de incapacidade ou inferioridade de um dos sentidos efectuar uma customização do comportamento do sistema através das “Opções de Acessibilidade”. Esta opção permite alterar o teclado, o som, o monitor e o comportamento do rato para deficientes motores, visuais e auditivos. Permite-nos definir os chamados “sons visuais” para deficientes visuais ou então características sonoras especiais para utilizadores invisuais.

A ajuda não se faz somente ao utilizador com dificuldades especiais mas também ao chamado utilizador comum. A ajuda através de som pode consistir em chamar a atenção do utilizador para um determinado facto ou adverti-lo que uma determinada acção não foi a mais correcta, como em caso de erro.

O som pode também ser usado para dar-nos uma orientação espacial. Este processo é bastante usado nos jogos de computador. O jogador recebe informação sonora por parte do sistema que lhe permite saber da sua localização espacial num determinado cenário. A proximidade de uma máquina ou de outra fonte sonora pode ser indicada pelo volume do som produzido por elas. Quanto mais alto é o som mais próximo o jogador está do seu objectivo. Este método pode ser usado em certas aplicações e em certos sistemas que usam metáforas diferentes da de *desktop*.

4.2.5.2 Interfaces com uso da voz.

À medida que a tecnologia evolui e com o desenvolvimento de novas técnicas para reconhecimento de voz alguns projectistas de *software* têm enfrentado o desafio de produzir interfaces baseadas na voz. Estas interfaces apresentam o mesmo tipo de problemas que as primeiras interfaces baseadas em comandos, em que a funcionalidade da aplicação não é mostrada e onde as possibilidades de interacção são invisíveis. Assim terão que ser usadas algumas técnicas complementares para guiar o utilizador.

O problema em saber o que dizer a uma aplicação de voz tem dois componentes: o utilizador pode assumir que o computador é capaz de compreender mais do que realmente é possível e o utilizador pode não conhecer as potencialidades disponíveis.

Tal como acontece com outros problemas de *design* as técnicas usadas para orientar o utilizador estão sujeitas a limitações tecnológicas.

A maior limitação advém do tipo de sistema de reconhecimento de voz usado pela aplicação. Alguns sistemas conseguem suportar um vocabulário amplo e contínuo enquanto outros só suportam um vocabulário limitado (tipicamente mais fiáveis). Outros sistemas requerem o uso de palavras soltas em vez de discurso contínuo. Existem ainda sistemas que permitem o chamado “*barge-in*” (reclamar, interrompendo) que consiste em poder-se interromper a resposta do sistema e continuar a ser compreendido por este.

A segunda limitação é o ambiente em que vai ser usado o sistema de reconhecimento de voz. Algumas aplicações são para funcionar através do telefone enquanto que outras são para usar conjuntamente com uma interface gráfica.

A terceira limitação tem a ver com o perfil do utilizador. Algumas aplicações são para serem usadas pouco frequentemente por uma vasta gama de utilizadores enquanto que outras são usadas muito frequentemente e por um limitado número de utilizadores. Ainda outras se situam numa zona intermédia entre as duas anteriores.

Uma das propostas de Yankelovich [Yankelovich96] para ajudar os utilizadores no *input* de voz para que os comandos sejam entendidos consiste na tentativa de encontrar os *prompts* ideais para estabelecer a comunicação com o utilizador. Estes *prompts* funcionam como na interface por comandos em que um *prompt* é apresentado quando o sistema está pronto

para receber um comando. Na aplicação baseada na voz um *prompt* será como que uma apresentação pelo sistema do cenário relativo a cada estado.

Por exemplo, numa aplicação para um banco, em que o utilizador pode telefonar e obter vários dados sobre as suas contas os *prompts* possíveis seriam do tipo:

1. Bem vindo ao Banco Imaginário Português. Que operação deseja efectuar?
2. Bem vindo ao Banco Imaginário Português. Pode verificar o saldo de conta, efectuar uma transferência ou fazer um pagamento. Que operação deseja efectuar?
3. Bem vindo ao Banco Imaginário Português. Pode verificar o saldo de conta, efectuar uma transferência ou fazer um pagamento. Diga uma das seguintes escolhas: saldo de conta, transferência ou pagamento.

As técnicas para implementar os vários tipos de *prompts* dividem-se em explícitas e implícitas ou em técnicas intermédias:

- ***Prompts explícitos***

O terceiro *prompt* apresentado é um exemplo de um *prompt* do tipo explícito. Este tipo de *prompts* pode ser usado quando o sistema de reconhecimento de voz tem um vocabulário bastante limitado. Estes tipos de *prompts* são também chamados de *prompts* directivos pois direccionam o utilizador numa das escolhas possíveis indicando-lhe exactamente o que deve dizer. No entanto, quando a lista de opções é muito extensa, este tipo de técnica torna-se pouco conveniente. Para solucionar este caso o sistema deve permitir o “*barge-in*” já antes mencionado. Quando a lista de escolhas é extensa o utilizador pode interromper o sistema quando ouvir a que lhe interessa sem ter que esperar até ao fim.

- ***Prompts implícitos***

O primeiro *prompt* apresentado nos exemplos anteriores é do tipo implícito, ou seja, não orienta o utilizador nas resposta a dar ao sistema.

Este tipo de *prompts* pode ser usado quando o sistema possui capacidade suficiente para lidar com gramáticas flexíveis e com uma lista de opções bastante grande. Esta técnica é também conhecida como conversacional porque o sistema estabelece uma conversa com o utilizador. Um sistema deste tipo está a ser desenvolvido pelo projecto ARPA para uma companhia de aviação.

- **Prompts incrementais e expansíveis**

Um *prompt* incremental começa por fornecer ao utilizador um *prompt* implícito e espera pela resposta deste. Este tempo de espera é predeterminado e, se for excedido ou o utilizador disser algo que não é interpretado pelo sistema, este fornece um *prompt* mais explícito. Esta técnica pode, por exemplo, ser implementada no caso do Banco Imaginário Português, em que o sistema começa por enviar o primeiro *prompt* e se o utilizador não fornecer uma resposta o segundo *prompt* é enviado. Se, mesmo assim, o utilizador não reagir convenientemente será então enviado o terceiro *prompt* que indica ao utilizador para dizer uma das três escolhas possíveis.

Um *prompt* expansível difere um pouco do incremental. Quando o utilizador fornece uma resposta não interpretada pelo sistema, este envia um *prompt* que é uma expansão do primeiro ou mais explícito: exemplo - 1. Quem está ao telefone? - O utilizador não indica o nome de forma conveniente - 2. Diga o primeiro e último nome por favor?

- **Redução**

A técnica de redução consiste em utilizar prompts mais simples (mais implícitos) à medida que o utilizador se vai familiarizando com o sistema. O sistema começa o diálogo com um *prompt* explícito e quando, mais tarde, necessita apresentar um *prompt* da mesma situação é enviado um mais implícito porque se depreende que o utilizador já sabe como interactuar com o sistema.

- **Pistas**

Esta técnica consiste em enviar um *prompt* implícito seguido de uma ou mais pistas sobre as possíveis respostas. Apesar de poder ser confundida com a técnica do *prompt* explícito esta difere da outra em dois factores. O primeiro é que ela não fornece ao utilizador todas as possibilidades (pistas possíveis) mas somente apresenta algumas das mais úteis ou mais comuns. Segundo, as pistas são usadas de forma decrescente com a experiência do utilizador, ou seja, à medida que o utilizador adquire mais prática no uso do sistema as pistas deixam de ser usadas e só são apresentados os *prompts* implícitos.

- **Técnicas específicas para sistemas multimodais.**

As técnicas anteriores são técnicas usadas em sistemas com interfaces baseadas na voz. Em interfaces com o utilizador que utilizem não só a voz mas que sejam também gráficas a estratégia aconselhada é induzir o utilizador a “dizer o que vê”. Em interfaces baseadas em janelas, menus e botões, o utilizador pode dizer qualquer comando visível. Num auto-rádio, por exemplo, o condutor pode dizer os nomes dos botões como volume, *play*, etc.

Outra possibilidade é o sistema enviar um *feedback* verbal quando o utilizador selecciona um dado comando. Assim quando o utilizador pressiona o botão do rato em cima de, por exemplo, o comando de “informação” acerca de um dado hotel, o sistema diz “Mostrar informação acerca do Sheraton”, o que permite ao utilizador emitir um comando falado da próxima vez que usar o sistema. Desta forma o sistema ensina o utilizador à medida que este o vai usando. Esta técnica é bastante útil porque o sistema vai mostrando ao utilizador as formas correctas de dizer um dado comando.

Estudos sobre sistemas com interfaces baseados na voz têm demonstrado que a percentagem de erro do utilizador baixa para cerca de metade ao fim de 10 minutos de interacção [Yankelovich96]. Não só o utilizador assimila com facilidade o vocabulário e gramática que pode indicar ao sistema como também tende a adaptar a forma de falar de maneira a ser melhor compreendido por este, esperando que o sistema termine de falar antes de dizer algo ou evitar pausas no meio das frases de forma a que o sistema não processe somente o que foi dito até à pausa.

4.3 Ferramentas de apoio ao desenvolvimento de GUIs

A sofisticação das interfaces gráficas ao longo dos tempos tem aumentado de tal forma que o tempo necessário para implementar do “zero” uma interface actual seria de tal ordem que comprometeria à partida qualquer projecto de aplicação.

Nas primeiras interfaces com o utilizador baseadas em comandos o código associado à interface era escasso e não representava mais do que uma pequena parte do tempo total do desenvolvimento da aplicação. À medida que as interfaces foram evoluindo e que se tornaram em interfaces do tipo WIMP o tempo gasto para as desenvolver tornou-se mais representativo relativamente ao tempo total. A necessidade de desenvolver ferramentas de suporte ao desenvolvimento de interfaces tornou-se mais óbvia.

Pode-se dizer que desde que existem interfaces com o utilizador que existem também sistemas especiais de *software* que ajudam o *design* e a implementação do *software* da interface. Muitos desses sistemas demonstraram ganhos de produtividade significativos para o programador e tornaram-se produtos comerciais bastante importantes. Outros, pelo contrário, não conseguiram fornecer ao programador o suporte necessário para o desenvolvimento de boas interfaces.

Actualmente as ferramentas de desenvolvimento são quase sempre constituídas por vários pacotes um dos quais é o de ajuda ao desenvolvimento da interface gráfica com o utilizador.

4.3.1 Classificação das ferramentas

As ferramentas de ajuda ao desenvolvimento de interfaces com o utilizador foram classificadas de várias formas através dos tempos. Alguns autores classificaram-nas como *Toolkits*, *User Interface Development Environments*, *Interface Builders*, *Interface Development Tools*, *Application Frameworks*, sendo, no entanto, a *designação* mais popular, *User Interface Management Systems (UIMS)*. Esta *designação* (UIMS) é mais utilizada para nomear sistemas de mais alto nível, que além de ajudar o programador na concepção da interface também permitem definir sequências de operações (o que acontece depois de cada evento). Outras *designações* são usadas para nomear ferramentas de mais baixo nível, sendo a mais frequente *Toolkit*.

Alguns autores têm optado por englobar todo o tipo de *software* vocacionado para a ajuda no desenvolvimento de interfaces com o utilizador numa só *designação*: *User Interface Tool (UIT)*. Estes UIT englobam desde os *toolkits*, passando pelos *Interface Builders*, até aos UIMS.

Os *Toolkits*, consideradas as ferramentas mais simples, são bibliotecas de subrotinas que criam e gerem os vários objectos de uma interface gráfica, como janelas, botões e menus. Basicamente, a estes objectos, que é dado o nome de *widgets*, com uma determinada aparência e comportamento, está associado um conjunto elaborado de rotinas, que comunicam com a aplicação através de *callbacks*, e que os especificam e manipulam. Os *toolkits* não suportam o *display* directo de dados da aplicação. O programador tem que utilizar ferramentas adicionais para implementar este processo. Geralmente o programador escreve a sua aplicação em C ou C++ e posteriormente compila e executa a interface e a aplicação. Estas ferramentas não permitem uma rápida prototipagem para o desenvolvimento de uma interface nem para o desenvolvimento incremental e recursivo de uma aplicação sem ter que escrever novo código, recompilar e correr o executável de cada vez que se pretende alterar algum aspecto.

Apesar de tudo estas ferramentas foram bastante utilizadas numa primeira fase de desenvolvimento de interfaces. Elas permitiam poupar tempo de programação e manter a consistência com aplicações existentes mas ainda deixavam a maior parte do trabalho para o programador. Estudos efectuados em 1991 [Myers92] mostraram que os utilizadores de um *toolkit* gastavam cerca de 60% do seu tempo de desenvolvimento da aplicação na parte da interface gráfica. Em contrapartida, no mesmo estudo, em que se verificou que mais de 40% dos programadores usavam uma ferramenta de suporte ao desenvolvimento de interfaces gráficas mais poderosa que um *toolkit*, os utilizadores dessas ferramentas gastavam cerca de 40% do seu tempo de desenvolvimento da aplicação na parte da interface gráfica. Essas ferramentas, como já referimos anteriormente, são as chamadas UIMS.

A Figura 38 apresenta o diagrama com os vários níveis de *software* de interface com o utilizador. É evidente que alguns sistemas, como os UIMS, englobam vários níveis.

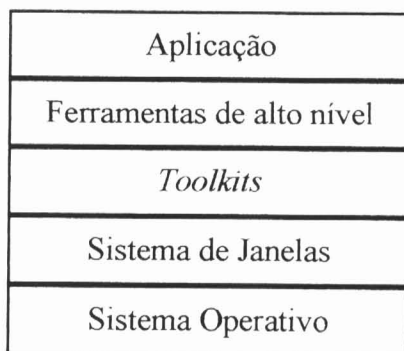


Figura 38 - Os vários níveis de *Software* de Interface

Os *UIMS* estão bastante acima dos *toolkits* em termos de funcionalidade. Estes sistemas fornecem, além do *toolkit*, incluído um ambiente de programação. Existem *UIMS* com diferentes capacidades. Desde aquele que são mais limitados e que só providenciam um desenvolvimento de programas simples ou de protótipos iniciais até aqueles que, apesar de requererem mais tempo e esforço de aprendizagem, permitem desenvolver desde um primeiro protótipo até à interface final com toda a funcionalidade incluída.

Os *UIMS* mais recentes assentam numa filosofia de programação visual. Esta forma de programação pode ser entendida como o desenvolvimento de aplicações através do desenho de imagens. Na programação visual os componentes gráficos da aplicação são desenhados no ecrã de um modo interactivo. Uma vez a aplicação construída o *UIMS* gera o código necessário para produzir a aplicação desenhada.

Apesar de os *UIMS* permitirem que se possam desenvolver aplicações sem se possuir conhecimentos profundos sobre regras de desenvolvimento de GUI's as aplicações daí resultantes não serão provavelmente as mais adequadas à melhor interacção com o utilizador. Como seria possível a um programador inexperiente, sem os mínimos conhecimentos sobre as regras fundamentais de interfaces, produzir aplicações tão “*user friendly*” como aquelas produzidas por uma equipa multidisciplinar conhecedora dos aspectos fundamentais sobre a interacção com o utilizador. É necessário que o desenvolvimento de aplicações seja, no mínimo, acompanhado por um projectista ou por um programador com bastantes conhecimentos de *design*.

Os *UIMS* incrementam a produtividade do programador, acelerando o processo de desenvolvimento e facilitando o refinamento iterativo da interface. No entanto, os *UIMS* actuais assumem a existência de um conjunto de primitivas (objectos de interacção) cuja apresentação é manipulada por eles. Se nenhuma dessas primitivas for exactamente aquilo que o projectista pretende então ou ele se resigna e elabora um novo projecto com os elementos de interacção disponibilizados pelo *UIMS* ou então “mete as mãos à obra” e codifica o diálogo pretendido através de uma linguagem de programação, isto porque a maior parte dos sistemas actuais não permite a definição de novos objectos de interacção.

A Figura 39 apresenta o diagrama que representa a estrutura básica de um UIMS.

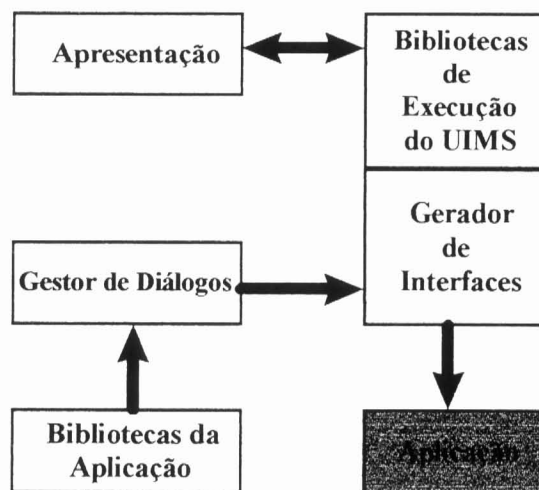


Figura 39 - Estrutura de um UIMS.

Na generalidade dos casos o modo como um UIMS funciona é o seguinte:

- Seleccionar e desenhar os componentes da interface.
- Configurar esses componentes através das facilidades oferecidas pelo UIMS para a edição das suas propriedades.
- Especificar o comportamento da interface, associando funcionalidades aos diversos componentes.
- Testar os componentes interactivamente durante o processo de desenvolvimento.
- Guardar o trabalho que foi realizado.
- Gerar o código fonte e todos os ficheiros de suporte para a aplicação.
- Compilar o código fonte no próprio UIMS ou num sistema exterior para produzir o executável da aplicação.

4.3.2 Vantagens e desvantagens no uso das ferramentas

Existem muitas **vantagens** no uso de ferramentas de apoio ao desenvolvimento de interfaces com o utilizador. Estas vantagens pode ser classificadas em dois grandes grupos:

- **A qualidade da interface será superior.** Isto porque:
 - Os *designs* podem ser rapidamente prototipados e implementados mesmo antes ser escrito o código da aplicação.

- As alterações propostas pelos testes com o utilizador são mais simples de incorporar.
- Podem existir interfaces múltiplas para a mesma aplicação.
- Pode-se dispensar mais tempo ao estudo da ferramenta pois ela será usada, provavelmente, várias vezes.
- Aplicações diferentes terão interfaces consistentes se forem criadas com a mesma ferramenta.
- Torna-se mais fácil o envolvimento de uma maior variedade de especialistas. Um profissional em interfaces gráficas, que não é um programador, pode ficar responsável pelo *design*.
- ***O código da interface com o utilizador será mais fácil e económico de criar e manter.*** Isto porque:
 - As especificações da interface podem ser representadas, validadas e avaliadas mais facilmente.
 - Haverá menos código para escrever pois a maior parte será providenciado pela ferramenta.
 - Haverá uma melhor modularização devido à separação da componente da interface e da aplicação. A interface pode ser alterada sem afectar a aplicação e também o contrário, alterar a aplicação sem afectar a interface.
 - O conhecimento de programação por parte dos *designers* e implementadores da interface pode ser menor visto que a ferramenta esconde a maior parte da complexidade do sistema.
 - A fiabilidade da interface será maior visto que o código é gerado automaticamente pela ferramenta.
 - Haverá maior facilidade de exportar a aplicação para outras plataformas.

Entre as **desvantagens** no uso das ferramentas podemos referir algumas.

- A dificuldade que algumas apresentam na sua aprendizagem.
- A funcionalidade limitada de algumas ferramentas (especialmente de alguns *toolkits* que só providenciam apenas uma parte do desenvolvimento da interface). A maior parte das ferramentas limita o projectista ao uso dos seus objectos de interacção não permitindo a criação de novos OI.
- A ideia de que uma ferramenta resolve o problema da usabilidade de uma interface. O facto da ferramenta ser bastante completa, só por si, não leva a um desenvolvimento de

uma interface “ideal”. Será sempre necessário ter um conhecimento mínimo das regras gerais abordadas no capítulo anterior.

4.3.3 A escolha de uma ferramenta

A escolha de uma boa ferramenta envolve factores bastante diferentes e por vezes contraditórios. Muitas vezes a escolha dá-se por ferramentas que não apresentam a funcionalidade pretendida e assim o projecto de desenvolvimento ficará limitado ao uso de uma ferramenta “menor”. Noutras os factores financeiros limitam o orçamento disponível e consequentemente a escolha da ferramenta. Quase sempre o custo é directamente proporcional à qualidade da ferramenta devendo, no entanto, prestar-se algum cuidado na escolha desta visto que o resultado pode ter um grande impacto na moral e produtividade de uma equipa assim como no sucesso do produto.

Autores como Lewis, Myers, Hix, definem alguns critérios para efectuar a escolha de ferramentas. Esses critérios podem ser mais ou menos específicos, definidos a um nível menos ou mais elevado.

Segundo [Myers95] a ferramenta deve:

- Escolher automaticamente quais os estilos, botões, dispositivos, etc., a serem usados na interface.
- Ajudar ao *design* gráfico e à disposição dos objectos.
- Validar os *inputs* do utilizador.
- Ser capaz de detectar erros do utilizador.
- Suportar o abortar e o desfazer (*undo*) de operações.
- Providenciar o *feedback* necessário para mostrar a validação dos *inputs*.
- Providenciar ajuda.
- Actualizar o *display* quando os dados da aplicação se alteram.
- Ajudar na sequência de operações.
- Providenciar facilidades de customização aos utilizadores finais.
- Avaliar a disposição gráfica, a usabilidade, e a facilidade de aprendizagem da interface.

[Lewis93] define critérios situados a um nível superior. Destacamos os seguintes:

- **Registo de acções**

A existência de elementos com capacidade de capturar a sequência de acções do utilizador é bastante útil para algumas fases do teste de usabilidade assim como para

determinar a existência de erros intermitentes na fase de teste de versões *beta* da interface. Esta descrição de acções deve ser efectuada a um nível relativamente elevado (o menu Editar foi aberto), e não se limitar a fornecer uma lista de teclas pressionadas e posições X,Y de *clicks* do rato.

- ***Suporte para programas extensos***

Devem-se procurar linguagens de programação e características de ambiente que suportem a aplicação de boas técnicas de programação: modularidade (ficheiros múltiplos, compilação por partes, suporte de versões), um bom programa de detecção e eliminação de erros. Devemos assegurar-nos de que a velocidade dos executáveis gerados pela ferramenta satisfaz as necessidades da nossa aplicação.

- ***Geração de código***

O UIMS gera código executável ou produz ficheiros de código fonte para serem compilados pelo nosso compilador habitual? A primeira hipótese é mais conveniente enquanto que a segunda nos dá maior flexibilidade. Não esquecer que o código gerado automaticamente é geralmente impossível de ser mantido por programadores humanos.

- ***Extensibilidade da interface***

Podem ser adicionados novos elementos de controlo e de interacção ao ambiente de programação que terão a mesma facilidade de programação que os iniciais? Igualmente importante é saber que extensões estão disponíveis.

- ***Extensibilidade do programa***

Pode o programa gerado chamar subrotinas escritas noutras linguagens?

- ***Técnicas específicas do sistema operativo***

O UIMS deve suportar todas as técnicas especiais de interface com que utilizadores de outros programas do mesmo sistema operativo estejam familiarizados. Um UIMS para Macintosh deve, por exemplo, suportar o *clipboard* do Macintosh e deve gerar aplicações que possam ser iniciadas pelo duplo *click* do rato nos seus ficheiros. Um UIMS para Windows deve suportar OLE (Object Linking and Embedding) e ajudar a construir um instalador da aplicação.

- ***Capacidade de desenvolvimento do protótipo até à aplicação final***

É desejável começar e acabar o desenvolvimento de uma dada aplicação com o mesmo UIMS, desde a elaboração de protótipos para teste até à implementação da solução final, reduzindo assim o tempo de desenvolvimento.

- ***Geração de aplicações Stand-alone***

É conveniente que o UIMS tenha a capacidade de produzir aplicações e que não seja necessária a aquisição de um pacote adicional de suporte para esse efeito.

- ***Suporte para regras de estilo***

Alguns sistemas UIMS verificam ou indicam automaticamente algumas regras de estilo de interface como “todas as caixas de diálogo devem ter um botão *OK*” ou “não devem ser usados mais do que 7 itens num menu”.

- ***Suporte comercial e longevidade***

O vendedor do sistema UIMS tem reputação? Efectua actualizações do UIMS adaptando-as aos avanços do sistema operativo. Será que a empresa que vende o UIMS ainda está em actividade quando a nossa aplicação se tornar um sucesso?

- ***Programação visual***

É possível definir visualmente a disposição no ecrã dos vários objectos e elementos de interacção? É possível a definição compulsiva do desenho da interface (alinhar à esquerda) ou a especificação por programação (todos os botões desta classe devem ter 75 pixels de altura)?

4.3.4 Ferramentas específicas de apoio ao desenvolvimento

A quantidade de ferramentas disponíveis no mercado para apoio ao desenvolvimento de GUIs é tão numerosa que não fazia sentido apresentarmos aqui todas elas. Refira-se no entanto que Myers possui uma lista actualizada da maior parte das ferramentas com os respectivos contactos da empresa responsável, que pode ser encontrada em [Myers97].

Neste subcapítulo apresentam-se algumas ferramentas, umas mais simples (*toolkits*), indicados para funcionar sobre sistemas de gestão de janelas (*Window Management System*) como o *X Windows System* do UNIX, o *Windows* da *Microsoft* ou o *MacOS* da *Apple*, outras mais completas, e faz-se uma avaliação mais pormenorizada daquelas para o sistema

Windows.

O sistema de janelas *X* é um dos mais poderosos e sofisticados jamais desenvolvido [Mikes91]. Para o sistema *X Windows* existem vários *toolkits* para o desenvolvimento de interfaces. Uns são destinados a trabalhar sobre o sistema MOTIF, sistema constituído por uma biblioteca de subrotinas na linguagem C que definem um conjunto de objectos da interface gráfica e técnicas para a combinação desses objectos numa aplicação. A definição da interface recorrendo somente ao MOTIF torna-se uma tarefa bastante árdua já que, por exemplo, para escrever um simples programa que permita visualizar uma única janela de diálogo requererá aproximadamente 100 linhas de código C. Entre as várias ferramentas disponíveis citamos as seguintes, algumas das quais (sublinhadas) avaliadas por [Sastry93]:

- *Toolkits*

Xt, **XView** da Sun, **Interviews** da Stanford University, **Open Interface** da Neuron Data, **Motif Toolkit** da Open Software Foundation, **XVT** da XVT Software Inc.

- *Sistemas intermédios e UIMS*

UIMX (Visual Edge Software ou Bluestone, Inc ou Black and White Software), **AIC** da IBM, **Builder Xcessory** da Integrated Computers Solutions, Inc., **DevGuide** da Sun, **X-Designer** da Imperial Software Technology, **XFaceMaker** da Non Standard Logics, **XMove** da Siemens AG.

Para o sistema Macintosh citamos como exemplo os seguintes:

- *Toolkits*

HyperCard e **Macintosh Toolbox** da Apple Computer

- *Sistemas intermédios e UIMS*

Action! Da ExperTelligence, **AppMaker** da Bowers Development, **Prograph** da Pictorius Incorporated

Para o sistema *Windows* vamos referir algumas ferramentas de desenvolvimento com especial ênfase naquelas que usam a filosofia de programação visual, que são as mais actuais e que, por serem de mais fácil utilização, se recomendam.

O Windows é um sistema operativo, predominantemente gráfico, inicialmente implementado nas plataformas do tipo Intel 80x86. A versão 3 do Windows, introduzida em 1990, foi um

dos produtos mais populares alguma vez desenvolvido. Com a introdução, em 1995, do Windows 95 este sistema reforçou a sua posição no mercado e o número de aplicações que correm sob o Windows acompanhou esse crescimento atraindo assim a atenção de inúmeras organizações de desenvolvimento de produtos incluindo algumas que desenvolvem UIMSs.

Com a popularidade do Windows a competitividade entre empresas aumentou consideravelmente o que levou ao aparecimento de algumas concorrentes da Intel, e consequentemente ao desenvolvimento de *hardware* compatível com aquele produto. Processadores como o Cirix ou o AMD estão já instalados no mercado disputando, inclusivamente, com a Intel o domínio deste. Por outro lado a evolução dos componentes, como os processadores, efectuou-se de uma forma muito rápida. À poucos anos atrás uma máquina topo de gama seria um PC com processador 80486 com 8MB de RAM, placa gráfica de 1MB, disco rígido de 600 MB, monitor de 14", CD-ROM de dupla velocidade. Hoje este tipo de PCs já não suporta as exigências do *software* que entretanto foi desenvolvido.

Para apoio à produção desse *software* as empresas responsáveis pela elaboração dos UIMSs desenvolveram sistemas mais completos que vão ao encontro das novas exigências. Actualmente existem sistemas orientados para a programação visual e por objectos. Esses sistemas têm demonstrado uma maior aceitação por parte dos projectistas e programadores.

Desses sistemas destacamos os seguintes [Anderson97]:

- **Delphi 2.0**

O Delphi da Borland é, provavelmente, a melhor ferramenta de programação visual para uso geral.

Esta ferramenta baseia-se no Object Pascal, uma versão da linguagem Pascal que sofreu tantas adaptações e melhoramentos que se tornou numa linguagem proprietária. O Delphi utiliza a Visual Component Library (VCL), uma biblioteca de classes que envolve a API do Windows. Os componentes VCL podem ser visualmente posicionados através do criador de formulários do Delphi ou inteiramente manipulados através de código. As chamadas à API do Windows são fáceis e sem limites e aspectos mais complicados como funções de *callback* e controlo de gráficos não constituem problema. É possível usar elementos OLE mas não controlos ActiveX.

- **Optima++ 1.5**

Desenvolvido a partir do Watcom C++, o Optima acrescenta-lhe potencialidades transformando-o numa ferramenta de desenvolvimento visual. O Optima funciona ao estilo do Visual Basic. Os componentes são inseridos numa paleta com aletas e os controlos ActiveX podem ser instalados através da criação de uma classe envolvente com um *Wizard* de componentes. A interface é desenhada num formulário onde as

propriedades dos componentes são definidas através de uma folha de propriedades e se pode fazer um duplo *click* sobre uma lista de eventos para escrever os recursos que permitem lidar com os eventos.

Apesar do Optima constituir um excelente produto que proporciona um desenvolvimento rápido e que dispõe de ferramentas visuais superiores ao Visual C++, apresenta alguns inconvenientes: não proporciona um modo fácil de criar novos componentes C++ e de os instalar na sua paleta e, apesar de permitir a criação de controlos ActiveX, não presta qualquer auxílio para esta tarefa. Também a falta de portabilidade é outro factor negativo, sendo a sua biblioteca, mesmo escrita em C++, exclusiva não permitindo a sua compilação noutra local.

- **Power Objects 2.0**

O Power Objects é uma ferramenta vocacionada para o desenvolvimento de aplicações para clientes de bases de dados. É especialmente adequado para sistemas Oracle e foi desenvolvido, em parte, como alternativa ao Visual Basic devido ao facto dos utilizadores de sistemas Oracle utilizarem frequentemente essa ferramenta para construir clientes para as suas bases de dados.

O seu ambiente de desenvolvimento é especialmente vocacionado para o Windows 95. O editor de código foi melhorado relativamente à versão 1.0 e o programa dispõe de um criador de relatórios incorporado. As aplicações são interpretadas e é possível efectuar a compilação para pseudocódigo, executável, para distribuição. Outra opção consiste em criar um executável independente que inclui uma biblioteca *runtime* num único ficheiro grande.

O Power Objects suporta ainda a programação por objectos, apesar de limitada, através de classes visuais. O ambiente de concepção desta ferramenta é um servidor de automação OLE. É possível servir de *host* a controlos ActiveX, fáceis de instalar através de uma caixa de diálogo de controlos personalizados. A edição Web é suportada através de um *plug-in* Netscape, também fornecido sob a forma de um controlo ActiveX para o Internet Explorer, que permite executar aplicações Power Objects na Web.

- **VisualAge Basic**

O VisualAge é uma ferramenta da IBM que se aplica a uma variedade de plataformas como o OS/2, AIX e Windows. Todos os produtos VisualAge se centram no desenvolvimento a partir de elementos separados que podem ser montados sob a forma de aplicações por um construtor visual. Este produto tem como base o Visual Basic 3.0 ao qual a IBM, apostando na compatibilidade total, acrescentou extensões orientadas por objectos e suporte para tecnologias estratégicas para a companhia, como objectos

SOM, OpenDoc e integração com o DB2, a base de dados multiplataforma da IBM.

Pelas características atrás mencionadas, esta ferramenta não traz grandes vantagens aqueles que criam aplicações para Windows, mostrando no entanto a sua utilidade para os utilizadores do DB2.

- ***Visual Basic 4.0***

Apesar de o Visual Basic (VB) ser menos potente que o C++ e não ser propriamente orientado para objectos (possui uma orientação por objectos parcial), continua a ser uma ferramenta de desenvolvimento que os utilizadores podem dominar em pouco tempo, tornando-se assim rapidamente produtivos.

O ambiente de desenvolvimento de “arrastar e largar” é altamente produtivo e é possível passar facilmente do modo de concepção para o modo de execução a fim de testar uma aplicação. O VB pode convocar DLLs (Dynamic Link Libraries) externas, além de utilizar controlos ActiveX, podendo, por isso, efectuar-se extensões escritas noutra linguagem, como o C++, para desenvolver operações que se encontrem para além do seu alcance.

A prioridade da Microsoft tem sido a de tornar o VB num ambiente ideal para usar componentes de utilização fácil e difícil de bloquear o sistema, em vez de investir em funções mais potentes, mas mais perigosas, como os ponteiros ou a alocação directa de memória. O êxito deste produto testemunha a validade desta abordagem.

- ***Visual C++ 4.2***

O Visual C++ é a escolha acertada para os programadores que queiram manter-se actualizados à medida que o Windows vai evoluindo, isto porque fornece extensões à Microsoft Foundation Classes, a biblioteca de classes que encapsula a maior parte das funções do Windows. O Visual C++ tornou-se um produto confortável para os programadores em C++. Possui um bom editor de recursos, pode facilmente acomodar os controlos ActiveX e gera código de execução rápida.

Este produto acusa, em parte, a sua idade à medida que a qualidade de outras ferramentas de desenvolvimento vai aumentando. Em termos de concepção visual tudo o que o Visual C++ pode oferecer é um editor de diálogos. Um assistente de classes cria os gestores de eventos por detrás dos controlos de diálogo.

Nos casos em que é necessário dedicar muito esforço à criação da interface este produto mostra algumas desvantagens relativamente a outros.

- **Visual FoxPro 5.0**

O FoxPro adquirido à Fox *Software* pela Microsoft, foi por esta melhorado e adaptado para o Windows.

À semelhança do Visual Basic, o FoxPro é uma linguagem interpretada que é acompanhada de um ambiente de desenvolvimento integrado, fornecendo várias ferramentas visuais. Esta ferramenta é altamente apropriada para lidar com bases de dados para computadores de secretária em que os dados estão no formato nativo do produto.

A orientação a objectos é completa suportando, inclusivamente, a herança e pode-se introduzir componentes ActiveX em subclasses. Este produto pode também ser chamado através de páginas Web.

- **C++ Builder**

O C++ Builder não é mais do que o Delphi para C++, empregando a biblioteca de componentes visuais (Visual Component Library) do Delphi em C++. Uma vez que o C++ é mais complexo que o Object Pascal a compilação é um pouco mais demorada mas é ainda suficientemente rápida para permitir a criação frequente de versões de teste. É possível partilhar componentes Delphi e C++ num único projecto o que se revela útil por permitir a migração de um para o outro.

Algumas ferramentas mais actuais começam a utilizar a linguagem de programação Java especialmente pela sua afinidade com a Internet. O Java é na sua concepção totalmente orientado por objectos e, além disso, possui capacidades multiplataforma. Embora a sua linguagem e ferramentas sejam ainda imaturas as novas versões têm resolvido a maior parte dos problemas. Seguidamente abordamos algumas ferramentas baseadas nessa linguagem.

- **Java Workshop**

O Java Workshop é escrito em Java. Cada ferramenta é uma página HTML que contém uma *applet*, incluindo um gestor de projecto, um editor, um gestor de versão, um *browser* de fontes, um desenhador de estruturas gráficas e um depurador.

- **Visual J++**

Ferramenta Java da Microsoft que se encontra englobada no Developer Studio.

Numa janela é possível passar do visionamento de classes do projecto, para uma vista de ficheiro ou para a árvore de auxílio *online*, no entanto um dos aspectos negativos reside na ausência de ferramentas visuais de concepção. Espera-se que o utilizador use

os editores de recursos do Visual C++ para criar um ficheiro de recursos padrão, de seguida um assistente (wizard) converte os recursos em código Java, ignorando tudo aquilo que o Java não puder concretizar e, finalmente, é necessário editar manualmente o código a fim de acrescentar os gestores de eventos. O Visual J++, em termos de concepção visual e criação de componentes através de um método “arrastar e largar” parece bastante mais fraco do que ferramentas como o Visual Café abordado mais à frente.

- **Visual InterDev**

O Visual InterDev (VI), englobado no Developer Studio como o Visual C++ e o Java C++, procura simplificar a tarefa de desenvolvimento de aplicações destinadas à Web.

Ao iniciar um novo projecto com o VI é criada uma nova área no servidor. Para proceder à criação de uma nova aplicação trabalha-se com vários ingredientes: ficheiros HTML, Layouts ActiveX, páginas activas de servidor contendo *scripts* que são executados no servidor, um ficheiro global que contém gestores de eventos, ficheiros de multimédia.

- **Visual Café**

O sucessor do Symantec Café tem sido, desde há algum tempo, o principal produto de desenvolvimento Java.

O Visual Café exhibe um Interface Development Environment (IDE) muito melhorado e uma ferramenta de concepção de layout gráfico que vai além da simples criação de interfaces. Uma inovação é o interface *wizard* que gera código para as tarefas mais comuns. Esta ferramenta dispõe de várias funções que a tornam numa verdadeira ferramenta de desenvolvimento visual. Uma é a facilidade com que os componentes podem ser criados e reutilizados. A janela da biblioteca de componentes exhibe todos os componentes disponíveis numa vista em árvore. Para incluir um destes componentes num projecto basta arrastá-lo para um formulário ou para a janela de projecto que também exhibe os seus componentes numa vista em árvore. Depois de, eventualmente, se personalizar os componentes alterando o seu aspecto ou acrescentando-lhes algum código basta arrastá-los de volta para a janela da biblioteca se pretendermos acrescentá-los à biblioteca de componentes.

O *browser* de classes é bem conseguido funcionando através de uma janela com três painéis. Um destes expõe as classes do projecto, outro mostra os métodos da classe que está activa e uma janela de código representa a fonte para o método activo. É possível usar o método de “arrastar e largar” para alterar as relações entre classes ou para criar novas classes.

Depois de, neste capítulo, termos abordado factores relacionados com o processo de desenvolvimento de interfaces, os princípios de concepção e apresentado as ferramentas de desenvolvimento, vamos de seguida apresentar os principais requisitos da interface gráfica com o utilizador para os sistemas domóticos.

Capítulo 5

Especificação de Requisitos da G.U.I. nos Sistemas Domóticos

Neste capítulo pretendemos efectuar a especificação dos requisitos relativos a cada um dos componentes da interface com o utilizador num sistema domótico. Esta especificação abrange desde o texto que é apresentado até aos objectos de interacção básicos a serem usados na interface, passando pela função de auto-repetição, bastante usada nos dispositivos de controlo por *hardware*, e pelos elementos de controlo por *software*. Esta especificação é, em grande parte, compilada, com algumas alterações introduzidas pelo autor, do documento FACE do projecto ESPRIT [FACE94].

5.1 Introdução

As interfaces com o utilizador em sistemas domóticos devem seguir alguns princípios básicos derivados da sua especificidade. O uso dos dispositivos de controlo na habitação segue regras diferentes daquelas para outros ambientes, como o de fabrico. Os vários grupos que actualmente desenvolvem sistemas nesta área apresentam diferentes abordagens

relativamente à forma como é realizada a interacção com o utilizador. O sistema X10, por exemplo, baseia a sua interacção em dispositivos de controlo por *hardware*, consolas com uma série de botões, que permitem identificar a unidade a controlar e a função pretendida. Outros sistemas, como o novo HES, usa uma interface baseada em *software*, apresentada ao utilizador através de um *touch-screen*, que pode estar situado em qualquer zona da habitação.

Thomas Heimer em [Heimer95], fornece-nos uma ideia do tipo de interfaces existentes à época para sistemas domóticos. Segundo Heimer uma interface com o utilizador comum em sistemas domóticos deve seguir algumas regras básicas, tanto a nível de *software* como de *hardware*.

Segundo este autor o *software* deve fornecer as seguintes características:

- Os menus devem ser estruturados de uma forma intuitiva e plausível.
- A estrutura para as várias tarefas deve ser idêntica. Isto significa que o uso de símbolos comuns aos vários menus deve ser efectuado de forma consistente.
- A interface deve informar sempre o utilizador sobre a função de um dado botão e como ele deve agir para voltar a uma dada configuração do sistema.
- A função “Ajuda” deve indicar ao utilizador onde está, o que fazer e como o conseguir.
- O número de passos para realizar uma dada tarefa não deve ser superior a quatro.
- Não devem ser apresentados no monitor mais que sete símbolos diferentes simultaneamente.

Quanto ao *hardware*, ele deve providenciar as seguintes características:

- Deve a interface com o utilizador ser um novo “móvel” ou, pelo contrário, deve ser integrada num dispositivo já existente? Por exemplo, a Televisão.
- A interface deve ser apropriada ao contexto da habitação. O uso de um *touch-screen* na cozinha é problemático, porque o utilizador, nessa situação, terá com certeza os dedos sujos.
- O *display* deve ser de fácil leitura.
- A interface não deve ser ruidosa como muitos PC’s.
- O *design* da interface com o utilizador deve inserir-se esteticamente na habitação.

Outros aspectos relativos à interface são abordados por Heimer, mas aqueles que apresentámos aqui são os que parecem mais importantes.

Seguidamente vamos indicar os principais requisitos da interface com o utilizador em sistemas domóticos.

5.2 Output de texto

Este subcapítulo especifica as regras gerais que devem ser seguidas na apresentação de texto nos vários dispositivos de apresentação.

5.2.1 Regra geral

O tamanho do texto a utilizar depende do sistema usado (painel, monitor, ecrã de televisão, LCD), mas, sempre que possível aconselha-se o uso da norma DIN 66234-1 que especifica que a altura mínima do texto a utilizar deve ser proporcional à distância ao observador pela fórmula:

$$h_{altura} = 0.0052 \times d_{distancia}$$

h-altura do texto

d-distancia de leitura (mm)

5.2.2 Legendas, abreviaturas, designações.

As legendas devem ser colocadas de forma a permitir a sua visualização mesmo quando o botão é pressionado (mais indicado no caso de painéis).

Devem ser usadas *designações* familiares para o utilizador e usadas no dia a dia. Evitar o uso de palavras demasiado técnicas ou “palavrões” de programador.

Fazer o menor uso possível de abreviaturas; usar, sempre que possível, as palavras completas. De contrário usar abreviaturas que não produzam confusão ou más interpretações.

Quando as abreviaturas forem estritamente necessárias devem ser usadas algumas das seguintes regras dependendo da tarefa a realizar:

- **Truncagem** - Eliminar o fim das palavras (ex. nos dias da semana, Segunda = Seg.,...)
- **Acrónimos** - Usar as primeiras letras de cada palavra (ex. Video Cassete Recorder = VCR)
- **Eliminação de vogais** - (ex. horas = hr)

5.2.3 Mensagens

5.2.3.1 Descrição

As mensagens são informação mostrada ao utilizador como resposta a eventos ou situações inesperadas as quais podem originar consequências indesejáveis. As mensagens são diferenciáveis segundo a sua importância como mensagens de informação, mensagens de aviso e mensagens de alarme.

5.2.3.2 Aplicação

As mensagens de informação devem ser mostradas ao utilizador sempre que ele deve ser informado sobre uma situação em particular mas para a qual não necessita de agir imediatamente.

As mensagens de aviso devem ser mostradas sempre que se possa vir a entrar numa situação indesejável ou crítica e quando seja necessária a escolha do utilizador para seguir ou abortar o processo.

As mensagens de alarme devem ser mostradas sempre que seja necessária a intervenção do utilizador por forma a evitar efeitos indesejáveis.

5.2.3.3 Apresentação

A apresentação das mensagens deve seguir algumas regras práticas:

- A mensagem devem indicar ao utilizador quais os eventos que provocaram a situação crítica e quais os passos a serem tomados para corrigir tal situação.
- O texto das mensagens não deve ser cínico, nem arrogante, nem atribuir sentimento de culpa.
- Devem ser evitadas as duplas negativas.
- As mensagens devem ser curtas, concisas e ditas na forma activa.
- Os símbolos relativos a cada tipo de mensagem deve ser colocado no lado esquerdo da janela da mensagem

5.2.3.4 Interacção

Se a mensagem tiver que ser confirmada isso deve ser claramente indicado ao utilizador.

Se a “Ajuda” estiver disponível deve ser fornecido um botão de “Ajuda” de forma a permitir ao utilizador o acesso a informação mais pormenorizada.

5.2.4 Mensagem de informação

As mensagens de informação devem ser usadas para manter o utilizador informado sobre o desenrolar de processos não críticos.

Estas mensagens podem ser colocadas na barra de *status* ou, em certos casos, abrir uma janela de mensagem quando a situação é nova ou inesperada e deve ser reconhecida pelo utilizador.

Deve ser fornecida uma orientação visual para as mensagens de informação, mas não deve ser fornecida uma orientação auditiva.

O ícone a utilizar será o carácter “i”. Este carácter pode ser estilizado mas deverá indicar claramente ao utilizador que se trata de uma mensagem de informação. A Figura 40 mostra um exemplo de uma mensagem de informação.

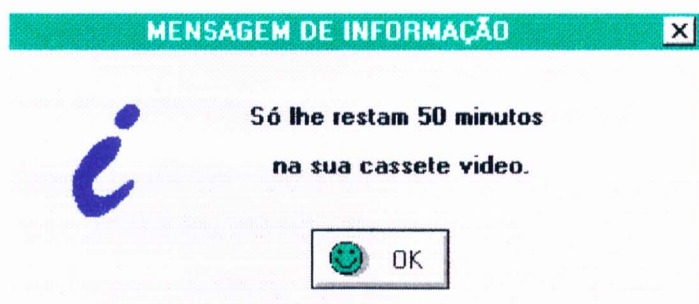


Figura 40- Nas mensagens de informação deve ser usado o símbolo "i".

A mensagem de informação pode ser desactivada por “*time out*”, pela seguinte acção ou pelo botão de “OK” que deve ser fornecido juntamente com a mensagem.

5.2.5 Mensagem de aviso

As mensagens de aviso devem ser usadas quando a continuação do processo depende da opção do utilizador.

Estas mensagens devem ser colocadas em janelas próprias.

Deve ser fornecida uma orientação visual para as mensagens de aviso e pode ser fornecida uma orientação auditiva por forma a atrair a atenção do utilizador para a mensagem.

O ícone a utilizar será o carácter “ ! ”. Este carácter pode ser estilizado mas deverá indicar claramente ao utilizador que se trata de uma mensagem de aviso. A Figura 42 mostra um exemplo de uma mensagem de aviso.

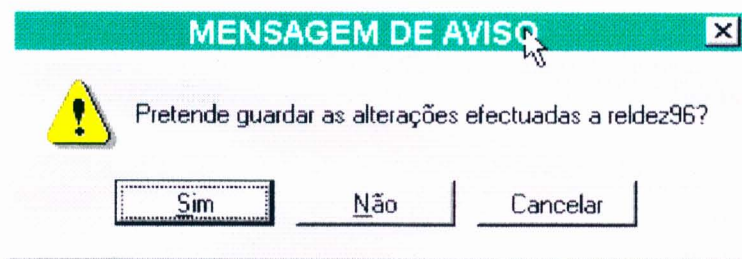


Figura 41 - Nas mensagens de aviso deve ser usado o símbolo "!"

Se forem indicadas várias opções deverá estar seleccionada por defeito aquela menos crítica ou a mais provável.

A mensagem de informação deve ser desactivada somente quando for realizada a acção necessária pelo utilizador.

5.2.6 Mensagem de alarme

As mensagens de alarme devem ser usadas quando é necessária a intervenção imediata do utilizador de forma a evitar estados indesejáveis do sistema.

Estas mensagens devem ser colocadas em janelas próprias e deve ser fornecida uma orientação auditiva por forma a atrair a atenção do utilizador para a mensagem.

O ícone a utilizar será o sinal de “Stop”. A Figura 42 mostra um exemplo de uma mensagem de alarme.

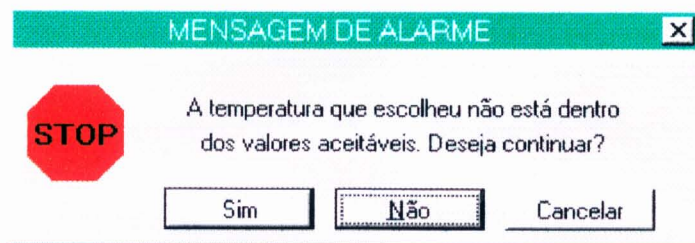


Figura 42 - Nas mensagens de alarme deve ser usado o símbolo "Stop"

Devem ser indicadas várias opções que permitam ao utilizador ultrapassar a situação que provocou o alarme, como mera informação ou como acções seleccionáveis.

Se forem indicadas várias opções deverá estar seleccionada por defeito aquela menos crítica ou a mais provável.

A mensagem de alarme deve ser desactivada somente quando for realizada a acção necessária pelo utilizador.

5.3 A Auto-Repetição

5.3.1 Introdução

A função de auto-repetição é uma das técnicas mais comuns quando existem muitos valores para serem incrementados ou decrementados. O utilizador invoca esta função e o sistema reage até que o utilizador a pare.

Apesar desta função ser mais indicada no caso do controlo por *hardware*, em que o utilizador não pode inserir directamente o valor desejado e tem que percorrer todos os valores intermédios até atingir o valor pretendido, ela pode também ser implementada no controlo por *software* em ambiente domótico sempre que o utilizador não possui o teclado para a introdução de valores numéricos e, por exemplo, o controlo é efectuado através de um *touch-screen* instalado numa mesa ou balcão ou mesmo na parede.

O aspecto de usabilidade mais importante desta função é precisamente o de permitir percorrer os vários valores automaticamente quando não é possível uma entrada directa.

Esta secção indica algumas recomendações para a implementação desta função. As recomendações têm como base as provas efectuadas aos utilizadores no âmbito do projecto FACE.

5.3.2 Conceitos

5.3.2.1 Intervalo inicial e intervalo de repetição

O intervalo inicial é o tempo desde a activação da função até ao início da auto-repetição.

O intervalo de repetição é o intervalo de tempo até que um valor seja automaticamente alterado pelo sistema.

O intervalo inicial e o intervalo de repetição não têm necessariamente o mesmo valor mas no exemplo da Figura 43 o tempo fixado para estes dois parâmetros foi de 0,2 segundos. Quando o utilizador pressiona o botão o valor do relógio avança um minuto e enquanto o utilizador mantiver o botão pressionado o relógio avançará um minuto em intervalos de 0,2 segundos.

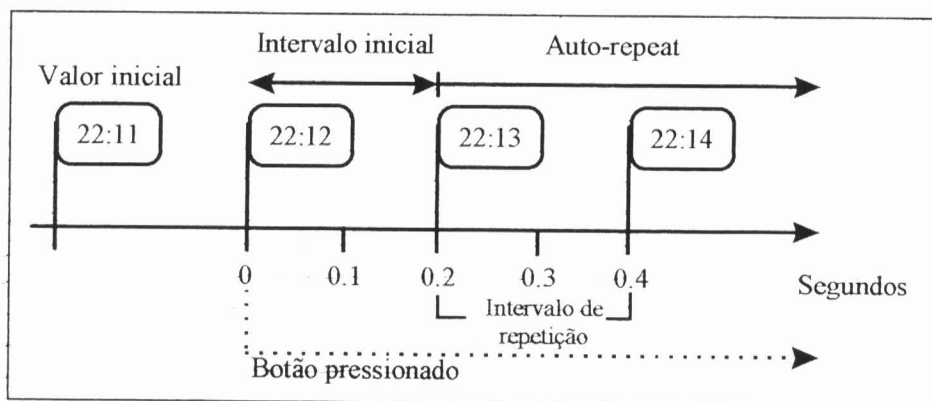


Figura 43 - Exemplo da função de auto-repetição.

5.3.2.2 Incremento / Decremento

O Incremento/Decremento é definido como a unidade de medida adicionada ou subtraída do valor corrente do contador.

Exemplo:

Incremento de 10 usado para acertar um contador de um VCR. Depois de o utilizador pressionar e manter premida a tecla de “+” o contador aumenta de 10 em 10 .

valor inicial:	124	
pressiona +:	125	incremento para cada toque = 1
mantém premida:	135	incremento para auto-repetição = 10
mantém premida:	145	incremento para auto-repetição = 10

5.3.2.3 Fases de repetição

As fases são definidas para permitir uma aceleração do auto-repetição. Se o utilizador tem que percorrer grandes distâncias para atingir um determinado valor, diferentes fases com diferentes velocidades são bastante úteis.

Se a função de auto-repetição for usada num determinado dispositivo, será usada pelo menos uma fase. Se forem implementadas duas fases, a auto-repetição acelera após um determinado período de tempo. Se forem implementadas três fases, a auto-repetição acelera novamente ao fim do tempo especificado para a segunda fase.

A Tabela 8 representa um exemplo da função de auto-repetição com três fases para um sistema com mudança de incremento e um intervalo de repetição constante.

Tabela 8 - Exemplo de auto-repetição com três fases.

Fase	Duração da fase	Parâmetros	Display
1	2 seg.	Intervalo de repetição: 0,2 Incremento: 1	valor inicial 20 muda 21,22,23,24,...30
2	2 seg.	Intervalo de repetição: 0,2 Incremento: 10	muda 40,50,60,70,...130
3	até que o utilizador termine a acção	Intervalo de repetição: 0,2 Incremento: 100	muda 230,330,430,...

5.3.3 Recomendações

As seguintes recomendações para os diferentes parâmetros da auto-repetição providenciam uma primeira orientação para a implementação desta função. As recomendações provêm de testes efectuados dentro do projecto FACE com mostradores de 2 dígitos (máximo 100 unidades), mostradores de 3 dígitos (máximo de 1000) e um relógio de 24 horas (máximo 1440 minutos; formato “hora hora : minuto minuto”, ex. 15:30).

5.3.3.1 Intervalo inicial

O valor do tempo para o intervalo inicial deve ser seleccionado entre 0,2 e 2 segundos. O valor a seleccionar depende da aplicação.

5.3.3.2 Incremento/Decremento

Deve-se usar o menor valor permitido pelo mostrador para o incremento ou decremento. Por exemplo, o menor valor possível para um relógio é geralmente o minuto. Sendo assim, deve-se usar o minuto como valor de incremento/decremento para acertar o relógio.

Em algumas tarefas, no entanto, o valor do incremento pode ser superior ao valor mínimo permitido pelo mostrador. Por exemplo, para programar períodos de aquecimento de um controlador de temperatura, é suficiente usar incrementos de 5 minutos porque valores menores tornam-se pouco significativos.

5.3.3.3 Intervalo de repetição e fases

Recomendações para intervalos de repetição e fases relativamente ao intervalo de valores a serem usados encontram-se na Tabela 9.

Tabela 9 - Recomendações para parâmetros da função de Auto-repetição.

Intervalo de valores	Intervalo de repetição e Fases	Duração das fases	Comentários/Exemplos
Intervalos pequenos	Fase 1 = 0,2 seg.	*	Para aumentar ou diminuir o volume de um amplificador ou aparelho de TV.
Máximo 100 unidades	Fase 1 = 0,2 seg. Fase 2 = 0,1 seg.	2 seg. *	Por ex., seleccionar 100 unidades num mostrador de 2 dígitos para um valor mínimo de incremento de 1.
Mais de 100 unidades	Fase 1 = 0,2 seg. Fase 2 = 0,1 seg. Fase 3 = 0,08-0,05 seg.	2 seg. 2 seg. *	Por ex., seleccionar mais de 100 unidades num mostrador de 3 ou 4 dígitos para um valor mínimo de incremento de 1.
Tempo Horas e minutos separados	Fase 1 = 0,2 seg. Fase 2 = 0,1 seg.	2 seg. *	Devem-se aplicar os parâmetros indicados se podemos seleccionar separadamente horas e minutos e se o valor mínimo de incremento é 1.
Tempo 12 ou 24 horas e minutos	Fase 1 = 0,2 seg. Fase 2 = 0,1 seg. Fase 3 = 0,08-0,05 seg.	2 seg. 2 seg. *	Devem-se aplicar os parâmetros indicados se o tempo a seleccionar é baseado num relógio de 12 ou 24 horas e se o valor mínimo de incremento é 1.

* dependente do utilizador

5.3 Elementos de Controlo

Os elementos de controlo abrangem somente os elementos de controlo por *software*. Abordamos os elementos mais comuns na implementação de uma interface, quando devem ser aplicados, a forma como eles devem ser apresentados e como se efectua a interacção com o utilizador.

5.4.1 Elementos de Controlo por Software

Esta secção trata dos tipos de elementos de controlo por *software* que podem ser usados no interface com o utilizador em aplicações para sistemas domóticos.

5.4.1.1 Lista de Opções (Drop Down List)

- **Descrição**

Uma Lista de Opções contém uma lista de objectos ou escolhas. O utilizador pode seleccionar apenas um item. A selecção actual é indicada num campo individual e os outros itens são apresentados a pedido do utilizador.

- **Aplicação**

Usar a “Lista de Opções” para situações em que o espaço é reduzido e é necessário apresentar todas as possibilidades ao utilizador. Por exemplo no preenchimento de um

formulário.

Usar a “Lista de Opções” para situações em que somente existe uma escolha possível.

- **Apresentação**

A “Lista de Opções” apresenta um item inicial até que o utilizador actue sobre uma seta “para baixo” situada no canto direito deste (Figura 44).

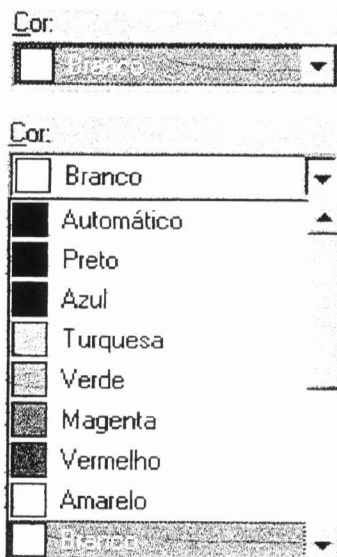


Figura 44 - Um exemplo de lista de Opções (Drop Down List)

O número de itens mostrado de cada vez deve ser entre 6 e 8 e cada um deve ser mostrado numa única linha. Quando o número de itens for superior deve existir uma barra de deslocamento para permitir o acesso aos itens ocultos. O espaço deve ser suficiente para mostrar todo o texto incluído em cada item.

Os itens devem ser mostrados numa ordem que facilite a tarefa do utilizador, minimizando o tempo de escolha. Esta ordem pode, por exemplo, ser alfabética.

- **Interacção**

Os itens são apresentados na lista conforme o seu estado (seleccionado ou não), isto é, com ou sem ênfase.

O item seleccionado deve aparecer também na lista e a selecção de um novo item desselecciona o item anterior. Deve ser também possível sair de uma “Lista de Opções” sem efectuar uma selecção.

5.4.1.2 Campo de Saída

- **Descrição**

Um “Campo de Saída” é um elemento de controlo de diálogo que apresenta informação estática ou dinâmica do sistema para o utilizador.

- **Aplicação**

Usar um campo de saída para mostrar alteração de informação estática ou dinâmica do sistema.

Exemplo

- A informação num formulário pode ser gerada e actualizada automaticamente através da informação introduzida pelo utilizador.

- **Apresentação**

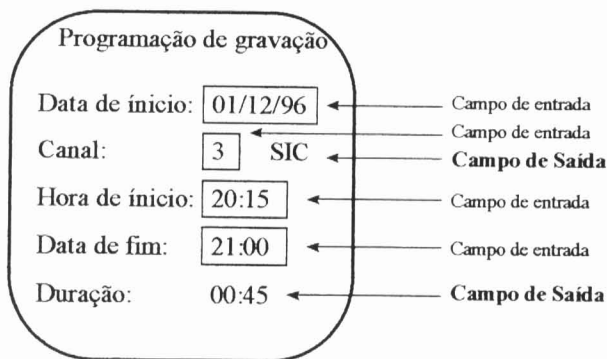


Figura 45 - Um exemplo de campos de saída e de entrada.

É recomendável usar uma legenda para cada campo de saída, colocada lateralmente ou superiormente ao campo que lhe está associado (admite-se que a legenda esteja colocada em baixo, mas, neste caso, os dois devem estar alinhados à esquerda).

Mostrar o conteúdo do campo de saída sem qualquer ênfase visual adicional (Recomendação: não usar caixas nem cor de fundo).

Na Figura 45 mostra-se um exemplo da programação de gravação num VCR com quatro campos de entrada e dois de saída.

5.4.1.3 Campo de Entrada

- **Descrição**

Apesar de o “Preenchimento de Formulário” constituir uma forma bastante acessível de entrada de dados, ele requer operações de movimento entre os campos, introdução de valores, alteração de valores, finalização de sequências de entrada, etc., para que seja perfeitamente compreendido pelo utilizador. Assim torna-se necessário estabelecer algumas regras para desempenhar estas operações:

Um campo de entrada é um objecto de diálogo que permite ao utilizador a introdução de dados. Consiste numa ou mais linhas que podem ter uma extensão fixa ou variável de entrada.

- **Aplicação**

Usar o campo de entrada se a entrada necessária não puder ser apresentada numa lista de escolhas.

Usar o campo de entrada para mostrar valores por omissão, que podem ser alterados pelo utilizador.

- **Apresentação**

Definir o desenho do campo para conter a extensão máxima de entrada quando esta é fixa ou previsível.

O campo de entrada deve ser bem definido por um rectângulo envolvente ou uma zona de outra cor.

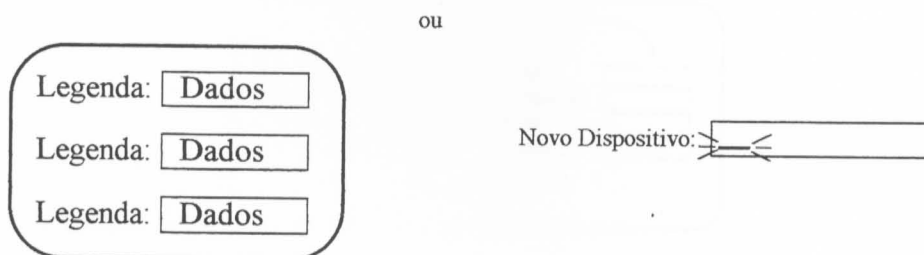


Figura 46 - Campos de entrada

As legendas dos campos de entrada devem ser curtas mas significativas e devem distinguir-se com facilidade dos campos de entrada. Evitar as abreviaturas que possam originar alguma ambiguidade para o utilizador.

Alinhar à esquerda tanto as legendas como os campos de entrada (Figura 47).

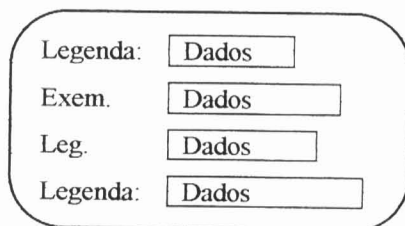


Figura 47 - As legendas e os campos de entrada devem ser alinhados à esquerda.

Evitar grandes distâncias entre a legenda e o campo de dados (Figura 48 a). Neste caso deve-se partir as legendas em partes (Figura 48 b) ou alinhar as legendas à direita (Figura 48 c).

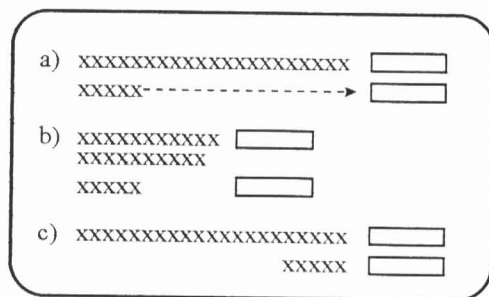


Figura 48 - Relação entre as legendas e os campos de entrada.

Os itens devem ser agrupados logicamente e usarem-se colunas verticais (Figura 49).

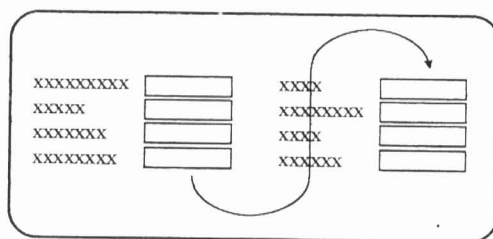


Figura 49 - Organização espacial dos campos de entrada.

Deve-se providenciar uma legenda para cada campo de entrada. A exceção a esta regra acontece quando os campos de entrada estão sob a forma de tabela.

Colocar a legenda à esquerda ou acima do campo de entrada. Quando colocada acima, a legenda e o campo devem estar alinhados à esquerda.

Usar disposição em grupo para campos que estejam relacionados por um determinado factor. Quando a disposição em grupo e a sua separação por título ou linha em branco não for suficiente para distinguir os determinados grupos, devem-se usar caixas para delimitar cada grupo.

- **Interacção**

Quando apropriado, deve providenciar-se a entrada de dados num determinado campo através da passagem pelos valores possíveis (incremento ou decréscimo através das teclas de ajuste) ou através da entrada directa (com teclas numéricas).

Exemplo

- Entrada de tempo (hora), níveis de temperatura.

Em determinados casos, é mais apropriado restringir a entrada a uma série de valores pré-definidos do que permitir a entrada directa.

Exemplo

- Num VCR a programação pode ser “uma vez”, “diária”, “semanal”, “mensal”.

Providenciar um valor por omissão possível que o utilizador possa alterar se o entender.

Exemplo

- Na hora de início de gravação de um determinado programa pode ser mostrada a hora corrente.

O campo de entrada em uso deve ser claramente distinguido dos outros através de um fundo diferente e quando se introduz informação o cursor deve piscar no sítio onde se encontra.

Devem ser estabelecidos limites quando o utilizador ajusta alguns valores.

Exemplo

- Ajustar valores de temperatura (entre 5°C e 30°C).

Se a ajuda estiver disponível, deve providenciar-se uma tecla de ajuda que permita ao utilizador obter uma explicação da forma como deve preencher o campo específico.

Tanto a validade da entrada como a sua coerência devem ser determinadas logo que o cursor saia do campo de entrada. Quando haja uma entrada incorrecta ou crítica, isso deve ser imediatamente indicado ao utilizador através de uma mensagem.

- **Retorno de informação ao utilizador (Feedback)**

Quando a entrada for de alguma forma codificada deve ser fornecido retorno ao utilizador através duma legenda suplementar num campo de saída.

Exemplo

- Canal de televisão.

Canal: RTP 1

Canal: SIC

5.4.1.4 Caixas de Selecção (Check Box)

- **Descrição**

Uma caixa de selecção é um controlo que permite ao utilizador modificar atributos ou parâmetros dos objectos individualmente. A Figura 50 mostra um exemplo de caixas de selecção.

Ligar luzes:

- Entrada
- Quarto
- Sala
- Casa de banho

Figura 50 - Caixas de selecção.

- **Aplicação**

Quando se pretende que o utilizador escolha entre ligar (on) ou desligar (off).

Usar uma caixa de selecção quando se pretende efectuar uma selecção múltipla duma série de itens que não são mutuamente exclusivos.

Usar uma caixa de selecção se o estado de seleccionado ou não seleccionado for

claramente compreendido e não apresentar ambiguidade.

- **Apresentação**

Uma caixa de selecção é apresentada juntamente com uma legenda-título. A caixa de selecção é colocada sempre no lado esquerdo.

Grupos de caixas de selecção podem ser apresentados em colunas linhas ou ambos.

Algumas das caixas de selecção podem aparecer inacessíveis, indicando que os valores correntes não podem ser alterados. Neste caso, tanto a caixa como o texto respectivo devem aparecer acinzentados.

- **Interacção**

Uma caixa de selecção passa a on quando seleccionada, se estiver no estado de não seleccionada, em off, ou com um padrão cinzento.

Uma caixa de selecção passa a off quando seleccionada, se estiver no estado de on.

5.4.1.5 Botão Rotação (Spin)

- **Descrição**

O botão rotação é um controlo que permite ao utilizador alternar por uma sequência de escolhas mutuamente exclusivas. É apresentado um exemplo de botões de rotação na Figura 51.

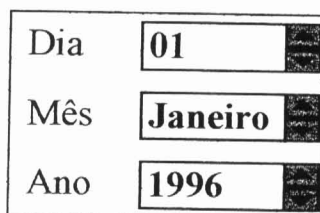


Figura 51 - Botão de rotação (Spin Button).

- **Aplicação**

Usar um botão de rotação para apresentar uma lista de escolhas que têm uma ordem lógica consecutiva (ex. dias do mês).

Usar um botão de rotação quando se deve escolher só um dos itens da lista.

- ***Apresentação***

Um botão de rotação deve ser apresentado com um campo de entrada, uma legenda e dois botões com setas para cima e para baixo.

A este tipo de botão aplicam-se também as regras referentes ao campo de entrada.

- ***Interacção***

Usar a seta para cima para mudar para a posição anterior em listas alfabéticas ou para valores mais baixos em listas numéricas.

Usar a seta para baixo para mudar para a posição posterior em listas alfabéticas ou para valores mais altos em listas numéricas.

Apresentar a lista como um anel de itens que salta automaticamente para o primeiro item quando passa o último e vice-versa.

Permitir que o utilizador possa introduzir os valores teclando directamente no campo de entrada do botão de rotação, especialmente se a lista for muito longa e o percorrer da mesma gastar demasiado tempo.

5.5 Elementos básicos comuns

Os elementos básicos comuns apresentados neste subcapítulo serão os elementos a utilizar, preferencialmente, num sistema domótico.

A filosofia inerente à utilização de elementos básicos comuns consiste em partir do pressuposto que se torna bastante mais simples para o utilizador comum reconhecer uma série de objectos que já fazem parte do seu dia a dia. Os diversos aparelhos que se encontram numa habitação actual possuem elementos que, pelo seu uso generalizado e utilização frequente, se tornam em elementos chave para a interface com o utilizador em sistemas domóticos. Podemos dar uma série de exemplos: a seta que aponta para a direita que é usada nos nossos VCR's que significa "andar para a frente em modo de leitura", a dupla seta que significa "andar a uma velocidade superior à normal", a tecla vermelha para gravação, a tecla verde das nossas máquinas de lavar que iniciam o seu funcionamento no programa escolhido, as teclas de "+" e "-" nos controladores por infra-vermelhos da nossa televisão ou outro dispositivo e as teclas "para cima" e "para baixo" que existem já há algum tempo nos "telecomandos" para mudar de canal de televisão, para um canal de número superior ou inferior.

Para além destes elementos comuns deve procurar-se encontrar elementos que representem cada um dos dispositivos da habitação e, conseqüentemente, do sistema domótico, por forma a permitir ao utilizador uma fácil identificação dos mesmos. Estes elementos novos serão esboçados no capítulo seguinte quando se falar sobre a elaboração do protótipo.

5.5.1 Os Botões

5.5.1.1 Botão OK

O botão de OK é empregue para providenciar uma forma de validação para qualquer tipo de entrada de informação, ex. validação de um formulário, activação de um item de menu previamente seleccionado, confirmação de uma mensagem, etc.

A legenda “OK” em combinação com o ícone ☺ é o aconselhável para o botão OK:

Exemplo:

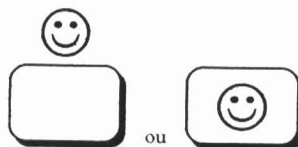


Opções:

Somente a legenda OK ou somente o ícone:



ou



ou



O botão OK é preferencial em qualquer dispositivo doméstico.

5.5.1.2 Botão CANCELAR ou CANCEL

O botão CANCELAR é empregue para providenciar uma forma de abortar o corrente diálogo, por ex. abortar o corrente menu, abortar um diálogo de formulário, abortar qualquer tipo de entrada de dados.

A legenda “CANCELAR” em combinação com o ícone ☹ é o aconselhável para o botão CANCELAR:

Exemplo:



Opções:

Somente a legenda CANCELAR ou somente o ícone:



Recomenda-se que a legenda seja escrita na linguagem do país em que a aplicação vai ser utilizada.

Por forma a providenciar uma correcção de situações de diálogo inesperadas é fortemente recomendada a utilização do botão CANCELAR.

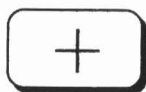
Depois de usar o botão CANCELAR, quaisquer alterações em parâmetros ou selecções não serão consideradas, voltando-se ao estado anterior ao da chamada da tarefa em questão.

5.5.1.3 Botão MAIS

O botão MAIS incrementa o valor corrente num campo de entrada de dados, executa selecção em campos de selecção e selecciona caracteres para entrada de texto.

Se não houver cursor o botão MAIS altera o valor de todo o campo. Se houver cursor o botão MAIS altera o valor da posição corrente do cursor.

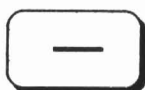
Usa-se o seguinte ícone para o botão MAIS

**5.5.1.4 Botão MENOS**

O botão MENOS decrementa o valor corrente num campo de entrada de dados, executa desselecção em campos de selecção e selecciona caracteres para entrada de texto na ordem contrária ao botão MAIS.

Se não houver cursor, o botão MENOS altera o valor de todo o campo. Se houver cursor, o botão MENOS altera o valor da posição corrente do cursor.

Usa-se o seguinte ícone para o botão MENOS



5.5.1.5 Botão *BAIXO (SEGUINTE)*

O botão *BAIXO (SEGUINTE)* providencia a forma de seleccionar itens de um formulário, de um menu, ou de uma lista em sentido descendente.

A apresentação do ícone do botão *BAIXO (SEGUINTE)* é preferentemente um triângulo isósceles apontando para baixo.



Opções:

Em vez do triângulo isósceles pode-se, alternativamente, usar a legenda “*SEGUINTE*” ou também usar uma combinação da legenda e ícone.



Pode-se também substituir o triângulo isósceles por uma seta qualquer estilizada, ou ainda fazer um botão em forma de triângulo ou seta.

5.5.1.6 Botão *CIMA (ANTERIOR)*

O botão *CIMA (ANTERIOR)* providencia a forma de seleccionar itens de um formulário, de um menu, ou de uma lista em sentido ascendente.

A apresentação do ícone do botão *CIMA (ANTERIOR)* é preferentemente um triângulo isósceles apontando para cima.



Opções:

Em vez do triângulo isósceles pode-se, alternativamente, usar a legenda “ANTERIOR” ou usar uma combinação da legenda e ícone.



Pode-se também substituir o triângulo isósceles por uma seta qualquer estilizada, ou ainda fazer um botão em forma de triângulo ou seta.

O botão CIMA (ANTERIOR) é recomendado como complemento do botão BAIXO (SEGUINTE).

5.5.1.7 Botão DIREITA

O botão DIREITA move o cursor para o sub-campo seguinte, à direita, de um determinado campo. Quando se atingir o último sub-campo o próximo pressionar do botão move o cursor para o primeiro sub-campo.

A apresentação do ícone do botão DIREITA é preferentemente um triângulo isósceles apontando para a direita.

**Opções:**

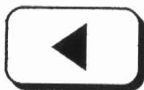
Pode-se também substituir o triângulo isósceles por uma seta qualquer estilizada apontando para a direita.

5.5.1.8 Botão ESQUERDA

O botão ESQUERDA move o cursor para o sub-campo anterior, à esquerda, de um determinado campo. Quando se atinge o primeiro sub-campo o próximo pressionar do botão move o cursor para o último sub-campo.

A apresentação do ícone do botão ESQUERDA é preferentemente um triângulo isósceles

apontando para a esquerda.



Opções:

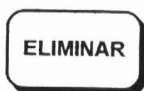
Pode-se também substituir o triângulo isósceles por uma seta qualquer estilizada apontando para a esquerda.

O botão ESQUERDA é recomendado como complemento do botão DIREITA de forma a permitir uma maior facilidade de movimentação do cursor.

5.5.1.9 Botão *ELIMINAR* ou *CLEAR*

O botão ELIMINAR apaga o valor do corrente campo.

“ELIMINAR” é a legenda preferencial para o botão ELIMINAR.



É aconselhável que o botão ELIMINAR esteja colocado numa zona menos usada ou separado dos outros botões de forma a evitar uma activação casual.

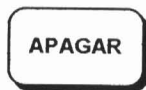
5.5.1.10 Botão *APAGAR*

O botão APAGAR deve estar disponível somente se existir a possibilidade de movimentação do cursor.

O botão APAGAR apaga o valor da posição corrente do cursor sendo que o restante texto ou número fecha o espaço ocupado, excepção feita para entradas fixas, como relógio formato 12 ou 24 horas, em que o apagar mantém um espaço no local do carácter apagado.

Se o cursor estiver no fim de uma sequência de caracteres ou número, o botão APAGAR funciona como BACKSPACE (PARA TRÁS), isto é, apaga o carácter imediatamente atrás e move o cursor para a esquerda.

“APAGAR” é a legenda preferencial para o botão APAGAR.



É aconselhável que o botão APAGAR esteja colocado numa zona menos usada ou separado dos outros botões, de forma a evitar uma activação casual.

Se o botão ELIMINAR e APAGAR estiverem presentes simultaneamente é preferível que sejam agrupados num local próprio.

5.5.1.11 Bloco Numérico

Apesar de o bloco numérico não ser um elemento usado em *software*, visto ele já estar presente nos teclados usados (*hardware*), sempre que seja necessário constituir uma entrada especial por *hardware* ou se se pretender implementar esta função quando usamos um *touch screen* deve-se respeitar a seguinte regra:

O bloco numérico é colocado em forma de bloco 3x3 com o zero por baixo da tecla 8.



Uma disposição alternativa é em linha:



Nos exemplos apresentados neste subcapítulo os botões aparecem com a legenda respectiva na sua parte superior, que, por princípio, se apresenta como preferencial em sistemas domóticos de controlo onde se torna necessário accionar o botão com o dedo. O facto da legenda ainda se apresentar visível mesmo quando o utilizador tem o dedo por cima do botão acrescenta um factor de usabilidade ao dispositivo. Se o botão estivesse tapado pelo dedo haveria alturas em que se tornaria necessária a confirmação visual do símbolo ou legenda inferior para evitar erros desnecessários. Na interface gráfica, em princípio, esta regra não é tão rígida e pode optar-se por colocar as legendas na parte inferior do botão ou

mesmo no próprio botão. Note-se que no desenvolvimento do protótipo se optou por colocar a legenda na parte inferior do botão por ser o método mais frequente nos dispositivos deste tipo.

5.5.2 Combinações

As combinações que se seguem, apesar de se aplicarem preferencialmente a dispositivos de entrada por *hardware* servem também como princípio de disposição de certas teclas na interface gráfica de forma a que se mantenha uma estrutura familiar para o utilizador.

5.5.2.1 Regras gerais de combinação

Se a combinação consistir na utilização de botões com funções opostas, eles devem ser colocados em pares (ex. MAIS e MENOS, CIMA e BAIXO, DIREITA e ESQUERDA).

Se as funções a executar consistirem principalmente em acerto de valores, é aconselhável juntar o botão MENOS ao botão MAIS; se consistirem em navegação por listas ou menus, recomenda-se juntar o botão CIMA (ANTERIOR) ao botão BAIXO (SEGUINTE).

Se os botões CIMA, BAIXO, DIREITA e ESQUERDA forem juntos em grupo deve-se usar o mesmo tipo de ícone para os quatro botões (triângulo isósceles ou outro tipo de seta) e devem-se dispor em forma de cruz.



5.5.2.2 MAIS e MENOS

Os botões de MAIS e MENOS são agrupados horizontalmente ou verticalmente.

A ordem mais natural para dispor horizontalmente os dois botões é o MENOS à esquerda e o MAIS à direita.

A ordem mais natural para dispor verticalmente os dois botões é o MENOS em baixo e o MAIS em cima.



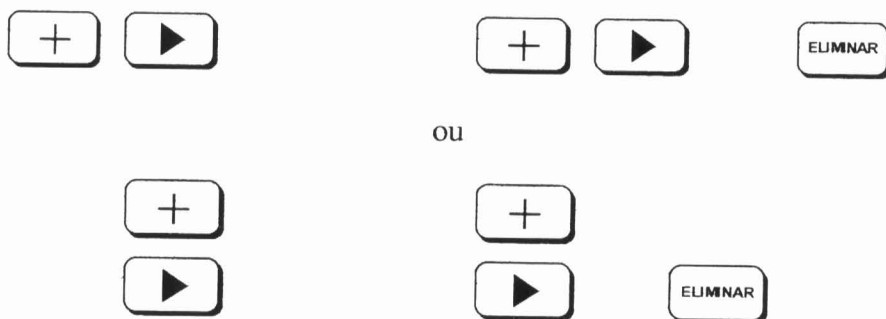
5.5.2.3 CIMA (ANTERIOR) e BAIXO (SEGUINTE)

A ordem mais natural para dispor os dois botões BAIXO (SEGUINTE) em baixo e o CIMA (ANTERIOR) em cima.



5.5.2.4 MAIS, DIREITA e MAIS, DIREITA e ELIMINAR

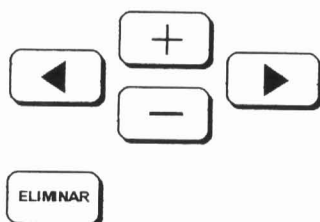
A ordem dos botões é MAIS, DIREITA da esquerda para a direita ou de cima para baixo. O botão ELIMINAR, quando usado, deve ser colocado separadamente dos outros dois botões.



5.5.2.5 MENOS, MAIS, DIREITA e ESQUERDA ou MENOS, MAIS, DIREITA, ESQUERDA e ELIMINAR.

A ordem natural da disposição destes quatro botões é colocar os botões de DIREITA e ESQUERDA à direita e à esquerda respectivamente. Consequentemente, os botões de MENOS e MAIS serão colocados em baixo e em cima respectivamente, formando os quatro uma cruz.

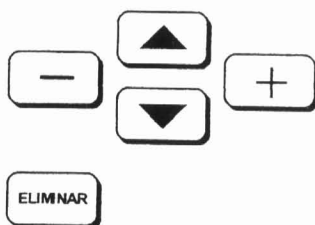
Quando usado, o botão de ELIMINAR deverá ser colocado separadamente dos outros quatro botões.



5.5.2.6 MENOS, MAIS, CIMA e BAIXO ou MENOS, MAIS, CIMA, BAIXO e ELIMINAR.

A ordem natural da disposição destes quatro botões é colocar os botões de CIMA e BAIXO em cima e em baixo respectivamente. Consequentemente, os botões de MENOS e MAIS serão colocados à esquerda e à direita respectivamente, formando os quatro uma cruz.

Quando usado, o botão de ELIMINAR deverá ser colocado separadamente dos outros quatro botões.



Apresentados os requisitos da interface gráfica com o utilizador em sistemas domóticos vamos, no capítulo seguinte, apresentar o *design* de um protótipo para uma interface com o utilizador, WinDomo, explicando sucintamente, a modo de manual, as acções possíveis.

Capítulo 6

Protótipo da Interface com o Utilizador da Aplicação WinDomo

Este capítulo apresentará a estrutura da interface gráfica com o utilizador da aplicação WinDomo, desenvolvida, sempre que possível, de acordo com as regras indicadas nos capítulos anteriores; serão mostrados as várias janelas da interface e será aproveitada essa apresentação para explicar o seu funcionamento e opções de interactividade.

6.1 O Projecto da Aplicação WinDomo

A aplicação WinDomo será incluída num sistema domótico, constituído por um conjunto de módulos instalados sobre a rede de 220V, efectuando a comunicação com este sistema através de um módulo especial que recebe as comunicações da porta RS-232 do computador e as envia para a rede. Em sentido inverso a placa envia a informação recebida pela rede para a porta RS-232.

Os dispositivos a instalar sobre a rede, responsáveis pela recepção e envio da informação, estão a ser desenvolvidos num outro projecto de mestrado. A Figura 52 mostra o esquema

do sistema a implementar.

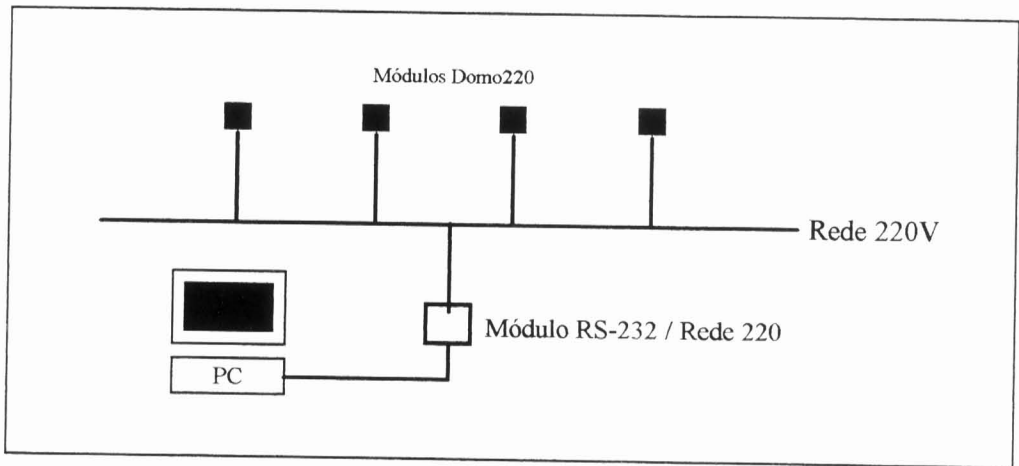


Figura 52 - A Estrutura do sistema domótico

O protocolo de comunicações da aplicação WinDomo com o módulo encontra-se sucintamente explicado no anexo A.

Sendo este sistema utilizado sobre a rede dos 220V destina-se fundamentalmente a ser usado em habitações já construídas, devido ao facto de a instalação de um barramento de comunicações novo ser economicamente menos vantajoso.

O protótipo aqui apresentado focará principalmente os aspectos relativos à interface gráfica com o utilizador comum. Os aspectos relativos quer à manutenção do sistema, quer ao apoio aos técnicos e instaladores não é abordado neste protótipo, sendo um dos trabalhos a desenvolver no futuro. A preocupação do autor, com este protótipo, é a de tentar fornecer um exemplo de uma estrutura de interface gráfica com o utilizador, concebida dentro do espírito de usabilidade e aplicabilidade para o utilizador comum. Assim, o *design* da interface pretende ser simples e de fácil utilização, apresentando uma estrutura facilmente aprendida pelo utilizador.

Alguns aspectos relacionados com a implementação da aplicação necessária para controlar o sistema são abordados, não sendo, no entanto, bastante aprofundados, por não constituírem, por si só, a finalidade deste trabalho.

6.2 Estrutura da Aplicação WinDomo

A estrutura da aplicação WinDomo está apresentada na Figura 53, e é constituída por uma janela principal dividida em grandes grupos, sendo cada um dos grupos, quando acedidos, constituídos por vários locais, e cada um dos locais podendo conter vários objectos.

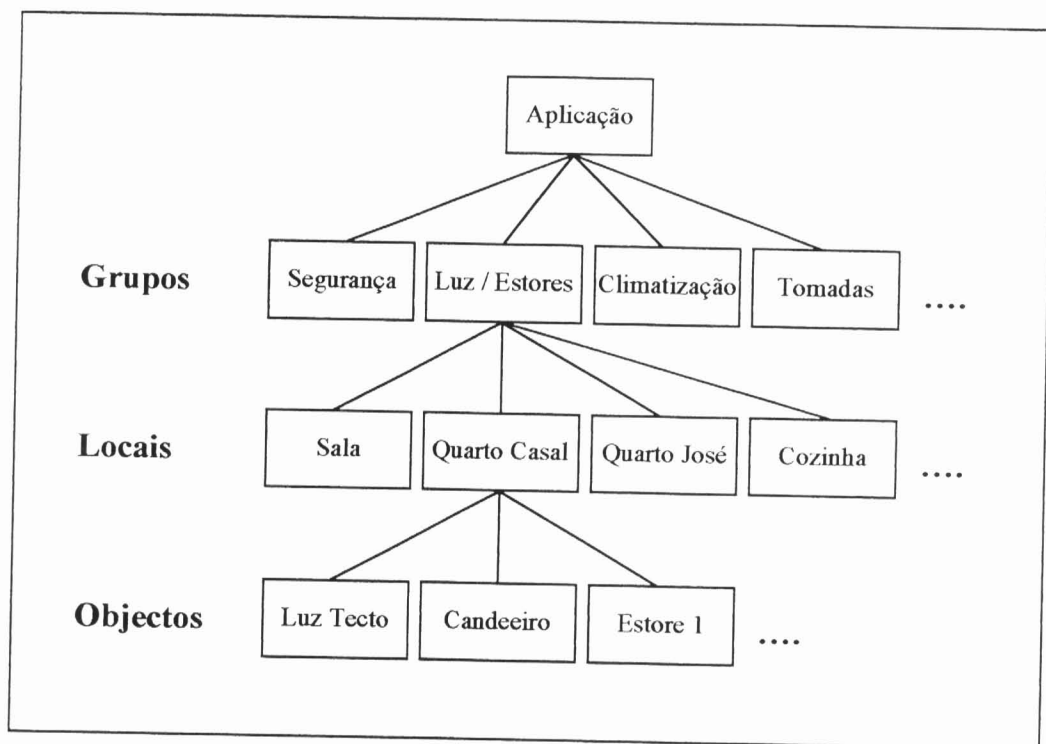


Figura 53 - Estrutura da aplicação WinDomo

O objectivo foi o de tentar atribuir à aplicação e respectiva interface um aspecto similar à estrutura de uma habitação, dividida por grupos de funcionamento (luzes, climatização, AVAC, electrodomésticos, aparelhos, segurança, etc.), instalados nas diferentes divisões da habitação (sala, cozinha, quartos, etc.) e com vários componentes do mesmo tipo em cada uma delas (várias luzes, estores, tomadas, dispositivos, etc.).

Para esclarecimento do leitor convém referir que a ideia inicial era o desenvolvimento de uma aplicação em Borland C++ para controlo de um sistema doméstico, no entanto, com o decorrer do projecto, as questões que se enfrentaram relativas à interface gráfica com o utilizador fizeram com que houvesse uma alteração do objectivo do trabalho e, por isso, o *design* que se apresenta é efectuado sobre o editor do Borland. A situação ideal seria tentar desenvolver a aplicação com o auxílio de uma ferramenta de programação visual do tipo Visual C++. É ainda de referir que, na altura em que o projecto foi idealizado, as ferramentas de programação visual, com grande ênfase no apoio ao desenvolvimento da interface, ainda não estavam tão difundidas como actualmente.

Em termos da aplicação, ela será orientada ao objecto e estruturada por classes. A classe inicial estará ligada por apontadores às classes do nível seguinte (grupos), sendo cada uma delas ligada às classes seguintes (locais) e por sua vez ligadas às classes que representam os vários objectos. Quando uma nova classe é criada é também construído o botão que a representa e imediatamente apresentado na janela correspondente.

6.3 A Interface com o Utilizador

A interface foi desenvolvida por forma a permitir o seu uso através de um ecrã do tipo *touch-screen*, pelo que a maior parte das janelas são desenhadas com esse objectivo. O tamanho dos botões é de forma a permitir a acção através do pousar do dedo sobre o monitor. No entanto, pode-se usar também o rato para realizar a mesma acção.

A Figura 54 apresenta um exemplo de janela inicial da aplicação. Os vários grupos são inicialmente elaborados pelo utilizador ou pelo instalador, mas aconselha-se a que se mantenha a estrutura idealizada por pensarmos que é a ideal. O sistema permite, no entanto, que o utilizador construa um sistema diferente mas que tem que obedecer à estrutura apresentada.

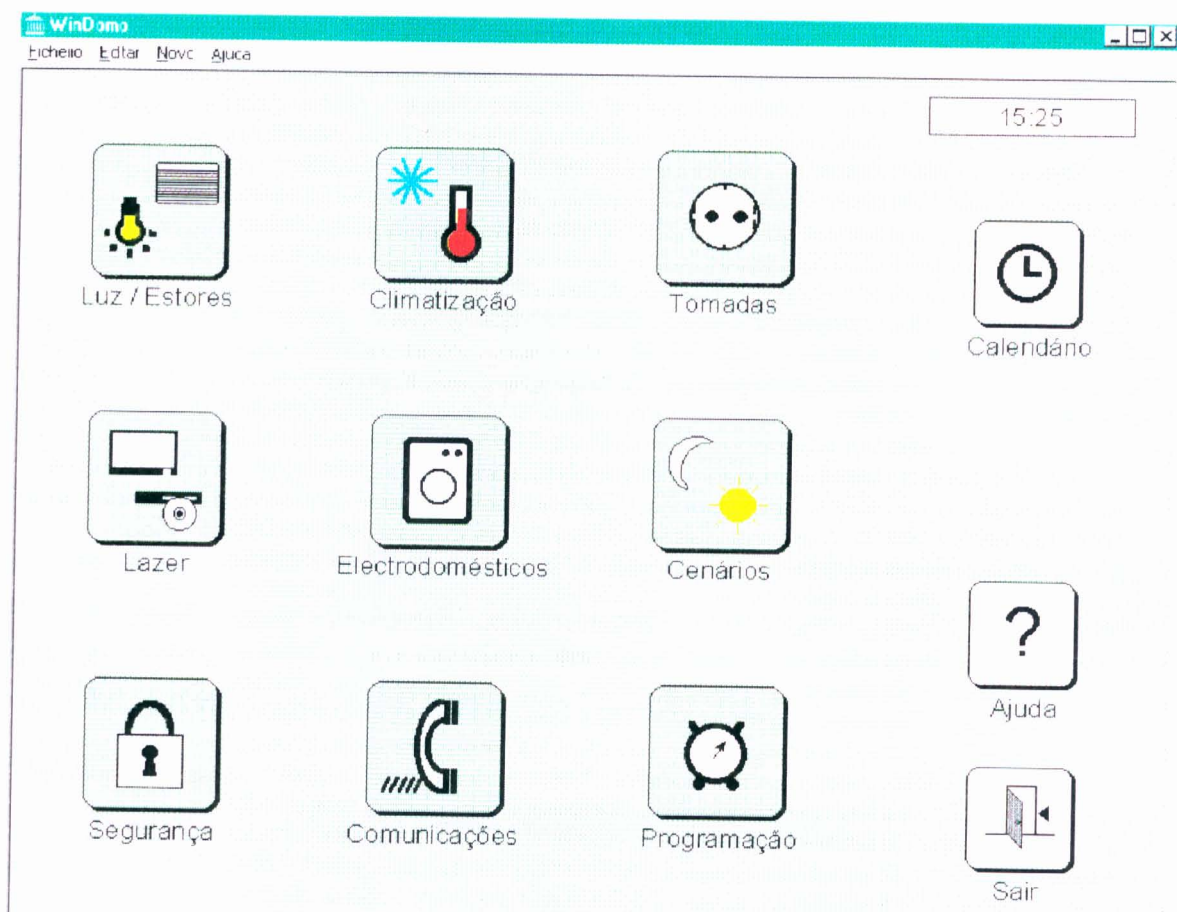


Figura 54 - A janela inicial da interface.

Neste protótipo não foram implementadas janelas de diálogo para todos os objectos possíveis mas, tão somente, os necessários para se poder concluir da usabilidade da interface. Para isso criaram-se janelas para os objectos representativos de cada um dos

grupos para assim se poderem efectuar os futuros testes com o utilizador.

Nos pontos seguintes explica-se a forma de interagir com a interface, a constituição das várias janelas e o seu objectivo.

6.3.1 O Menu

O menu da aplicação é constituído por quatro submenus, cada um deles constituído por vários itens.

O submenu “Ficheiro” efectua a gestão de ficheiros relativos à aplicação.

A Figura 55 mostra os itens constituintes deste submenu.

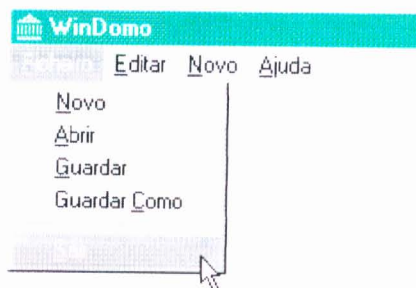


Figura 55 - Os vários itens do submenu "Ficheiro".

Novo

Através deste comando a interface actual é eliminada e a aplicação encontra-se na sua estrutura mais simples sem nenhum grupo ou objecto associado. Se tiver havido alguma alteração na interface anterior a aplicação abrirá uma janela de diálogo onde se pergunta ao utilizador se pretende guardar as alterações efectuadas (Figura 56).

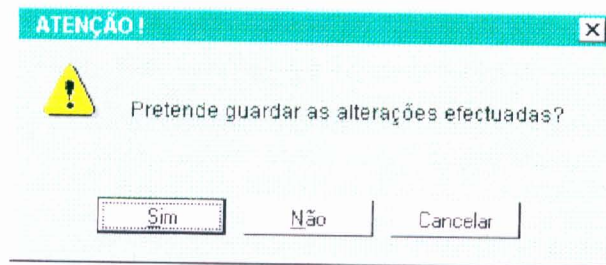


Figura 56 - Mensagem de aviso.

Abrir

Permite fazer a leitura de um ficheiro que contém uma interface criada anteriormente pelo utilizador ou instalador. Se tiver havido alguma alteração na interface actualmente aberta a mesma acção explicada anteriormente é efectuada.

Guardar

Permite guardar a interface actual. Se esta for a primeira vez que este comando é chamado para a interface actual então o comando seguinte (Guardar Como) é chamado.

Guardar Como

Permite guardar a interface actual com um determinado nome dado pelo utilizador numa dada localização.

Sair

Apaga a interface actual e abandona a aplicação WinDomo. Se tiver havido alguma alteração na interface actual a aplicação questiona o utilizador se pretende guardar as alterações efectuadas.

O itens do submenu “Editar” encontram-se representados na Figura 57.

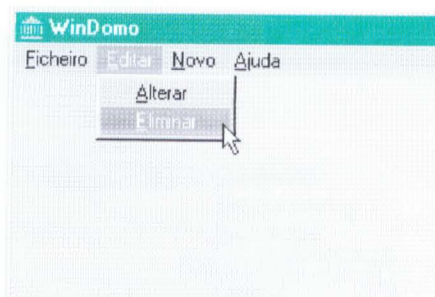


Figura 57 - Os itens do submenu "Editar".

Alterar

O comando alterar permite ao utilizador alterar as características de um determinado botão, tal como os ícones utilizados para representar os seus estados, o endereço do módulo que contém os serviços fornecidos pelo respectivo objecto ou o nome do botão.

Eliminar

Este comando permite ao utilizador eliminar um determinado botão da interface. Esta acção é confirmada através de uma janela de mensagem de aviso do tipo da Figura 56.

O submenu “Novo”, representado na Figura 58, permite adicionar novos elementos à nossa interface.

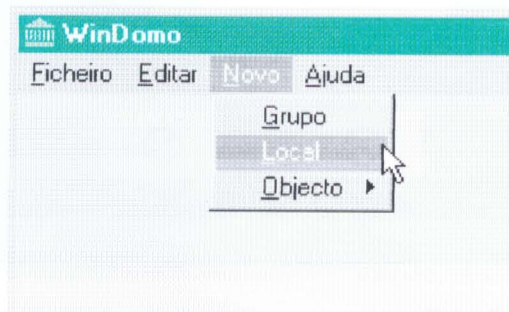


Figura 58 - Os itens do submenu "Novo".

Este submenu subdivide-se em três itens diferentes. Os vários elementos que podem ser adicionados à interface dividem-se em três áreas: grupo, local e objecto.

O item dos objectos é dividido em várias áreas. Exemplos das áreas encontram-se representados na Figura 59.

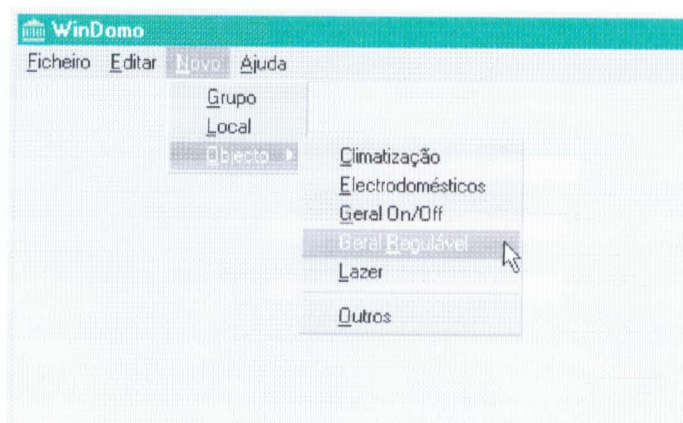


Figura 59 - As áreas em que se divide o submenu "Objecto".

Grupo e Local

Quando se pretende formar um novo grupo ou local é aberta uma janela de diálogo, representada na Figura 60, onde são introduzidos os elementos relativos ao botão: nome e imagem do ícone correspondente. Antes de se adicionar um novo botão é necessário que o utilizador entre no painel referente ao grupo onde esse objecto se vai inserir ou, no caso de um novo grupo, se encontre no painel inicial.

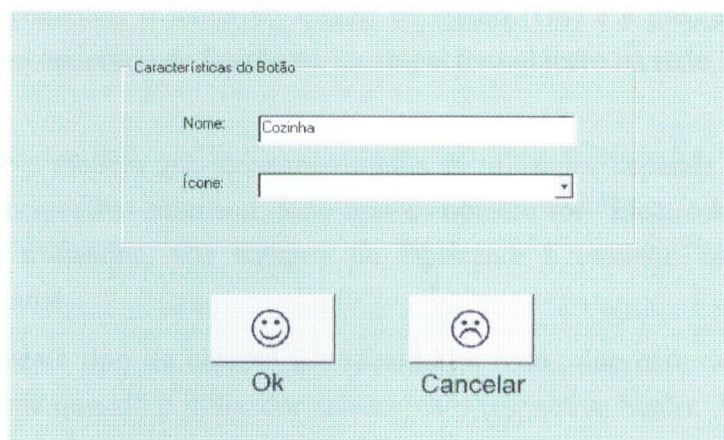


Figura 60 - Janela de criação de um novo botão “Grupo” ou “Local”.

Objecto

Quando se pretende adicionar um novo objecto a um determinado local, depois de escolher o tipo de objecto que se pretende a partir do submenu “Objecto” mencionado anteriormente, é apresentada uma janela para que o instalador ou utilizador possa indicar os dados relativos ao objecto pretendido. A Figura 61 representa um exemplo do tipo de janela apresentada quando se escolhe um novo objecto do tipo “Electrodoméstico”.

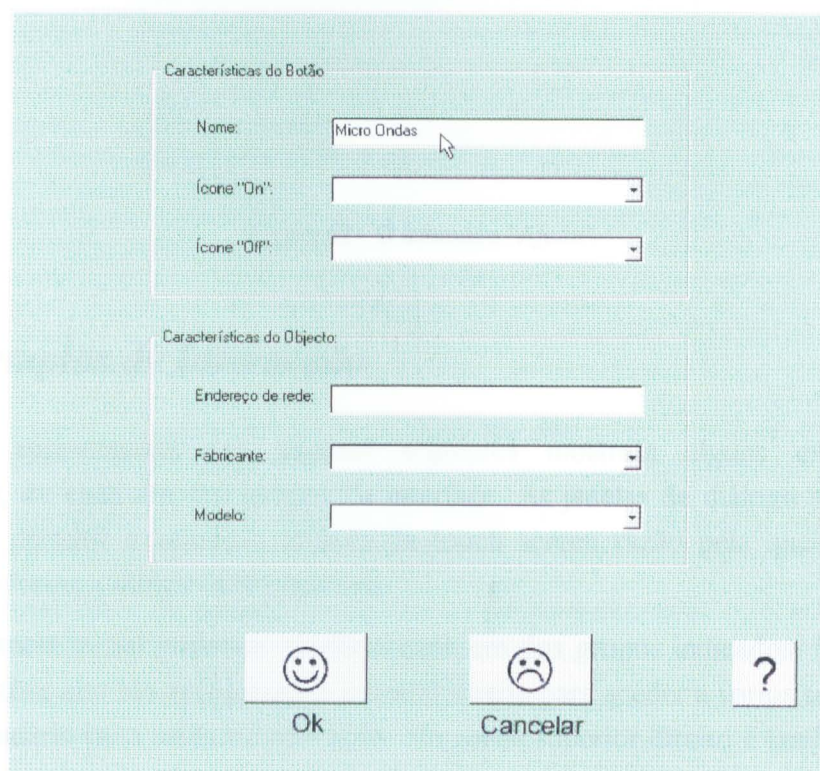


Figura 61 - A janela de criação de um novo botão “Objecto”.

Os dados a introduzir dizem respeito ao nome que vai constituir a legenda do botão, à imagem que vai constituir o ícone do estado de ligado (On) e à imagem para o ícone de desligado (Off), ao endereço de *hardware* do respectivo objecto na rede, o tipo de objecto e o seu modelo.

O tipo de objecto e modelo que são apresentados ao utilizador depende do tipo de objecto seleccionado no respectivo submenu. Se o tipo de objecto for “Electrodoméstico” só serão apresentadas ao utilizador, nos campos de fabricante e modelo, os dispositivos que pertencem a esta área.

Conforme o diferente tipo de objecto introduzido na rede, diferente deve ser a janela de diálogo apresentada quando o utilizador selecciona o respectivo botão. A janela de diálogo para uma máquina de lavar será concerteza diferente daquela para uma televisão, os eventos serão completamente diferentes, e os factores envolvidos díspares. Só se poderá instalar novos objectos se eles fizerem parte da lista incluída com a aplicação ou, então, a partir de novos “*drives*” dos respectivos dispositivos a instalar.

O submenu “Ajuda”, mostrado na Figura 62, permite aceder à ajuda geral sobre o sistema.

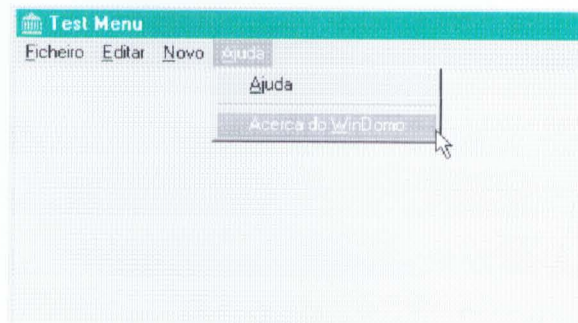


Figura 62 - O submenu "Ajuda".

6.3.2 Exemplos de Interação

Os painéis apresentados nas páginas seguintes mostram alguns exemplos, mais significativos, de cada um dos grupos da interface. As janelas de diálogo são desenhadas por forma a permitir a actuação através de touch screen razão pela qual o layout está elaborado de forma a atingir esse objectivo.

A partir da janela inicial podemos aceder a cada um dos grupos indicados. Podemos ainda utilizar os botões que são apresentados na parte direita para aceder a vários serviços como a ajuda, o calendário ou a saída da aplicação. No canto superior direito é também indicada a hora actual.

Vamos apresentar cada um dos grupos por ordem de leitura (esquerda-direita; cima-baixo):

Luz/Estores

Quando o utilizador acede ao grupo de Luz/Estores é apresentado um novo conjunto de botões, cada um representando um local da habitação (Figura 63).

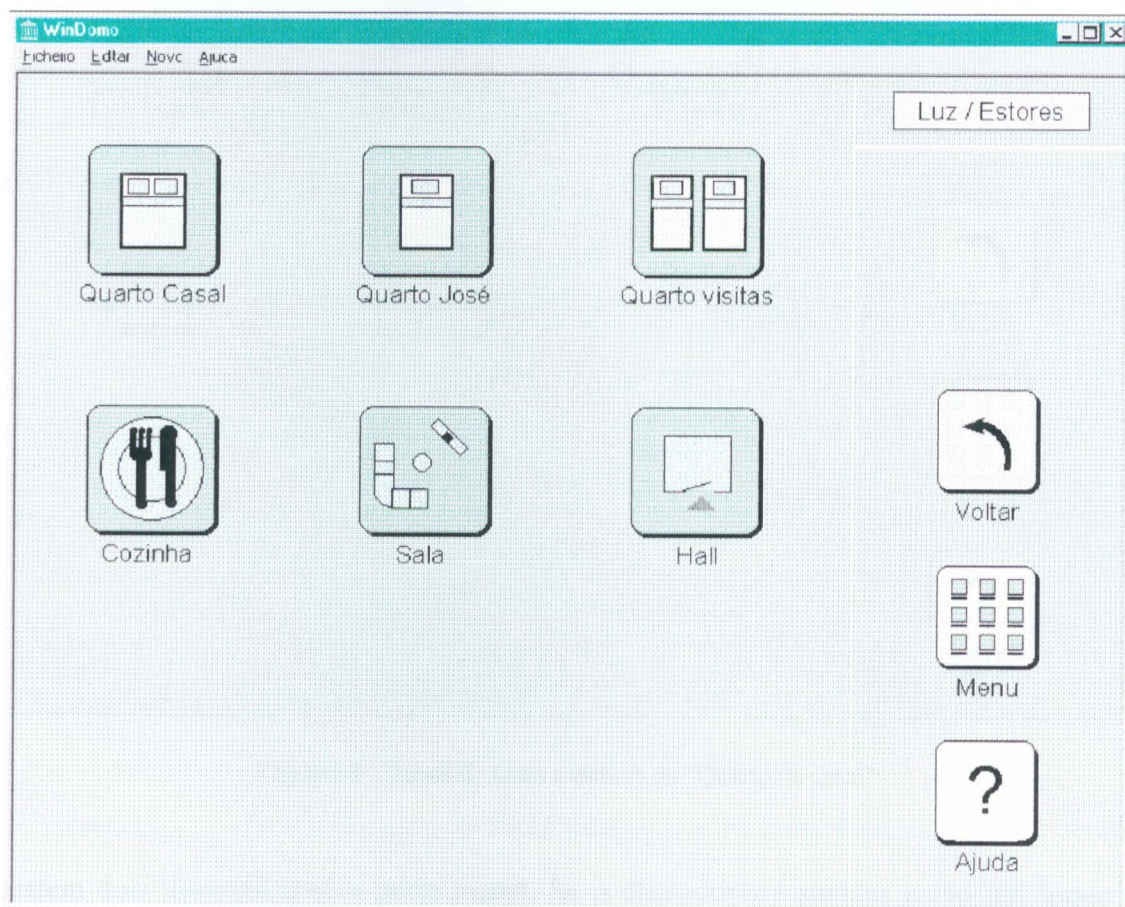


Figura 63 - Painel de "Luz/Estores".

Através deste painel o utilizador pode seleccionar um dos locais. O utilizador pode, também, voltar ao painel anterior através do botão de “Voltar” situado na zona direita do painel. Pode também obter ajuda através do botão “Ajuda”. Os botões situados na zona direita do painel encontram-se presentes ao longo de toda a interface. O botão de “Voltar” permite voltar ao painel anterior, o botão “Menu” permite regressar directamente ao menu inicial e o botão “Ajuda” abre a ajuda ao utilizador. A ajuda que é fornecida ao utilizador é diferente conforme o contexto em que ele se encontra. Se, por exemplo, o utilizador está perante um painel de “Luz/Estores” a ajuda que lhe é fornecida diz respeito às acções possíveis para atingir objectivos relacionados com o acender e apagar de luzes e com o abrir ou fechar os estores.

O exemplo seguinte, Figura 64, apresenta o painel relativo ao “Quarto de casal” onde estão representados cada um dos botões de luzes e estores.

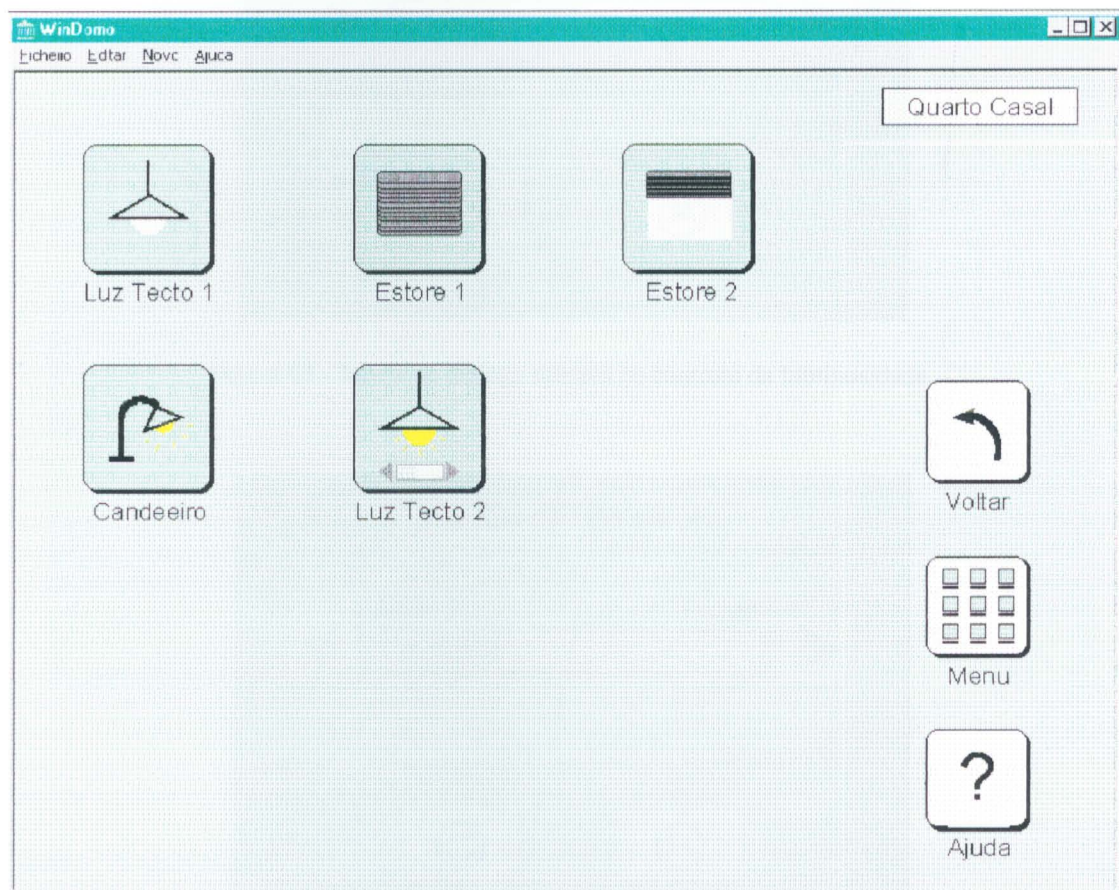


Figura 64 - Painel de luzes e estores do "Quarto de casal".

Existem dois tipos de botões neste painel. Se o dispositivo é simples então um toque no botão realiza um comando imediato. Após o comando ter sido executado, a imagem do botão é substituída por forma a fornecer uma indicação visual ao utilizador do novo estado do dispositivo; exemplo deste facto são os botões “Candeeiro” e “Luz do tecto 2” que têm a cor da lâmpada amarela, indicando que a respectiva luz está acesa, ou, no caso do “Estore 2” o estore encontra-se aberto. Se o dispositivo é do tipo “*Dimmer*”, antes que se realize qualquer acção é apresentada uma janela de diálogo para que o utilizador forneça os elementos necessários. As Figura 65 e Figura 66 apresentam as janelas de diálogo usadas para o controlo de luminosidade e regulação dos estores.

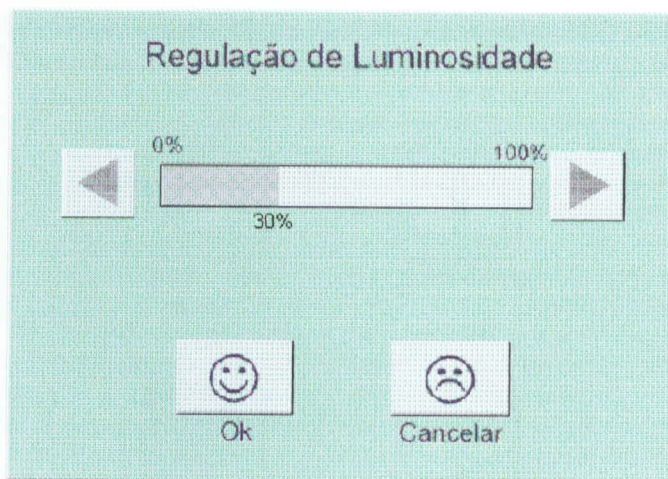


Figura 65 - Janela de diálogo relativa ao controlo da luminosidade.

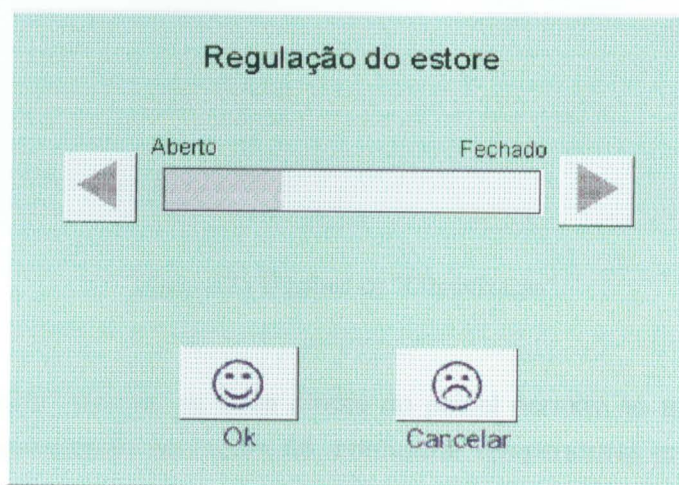


Figura 66 - Janela de diálogo relativa à regulação do estore.

Climatização

O painel apresentado na Figura 67 representa o grupo de “Climatização”.

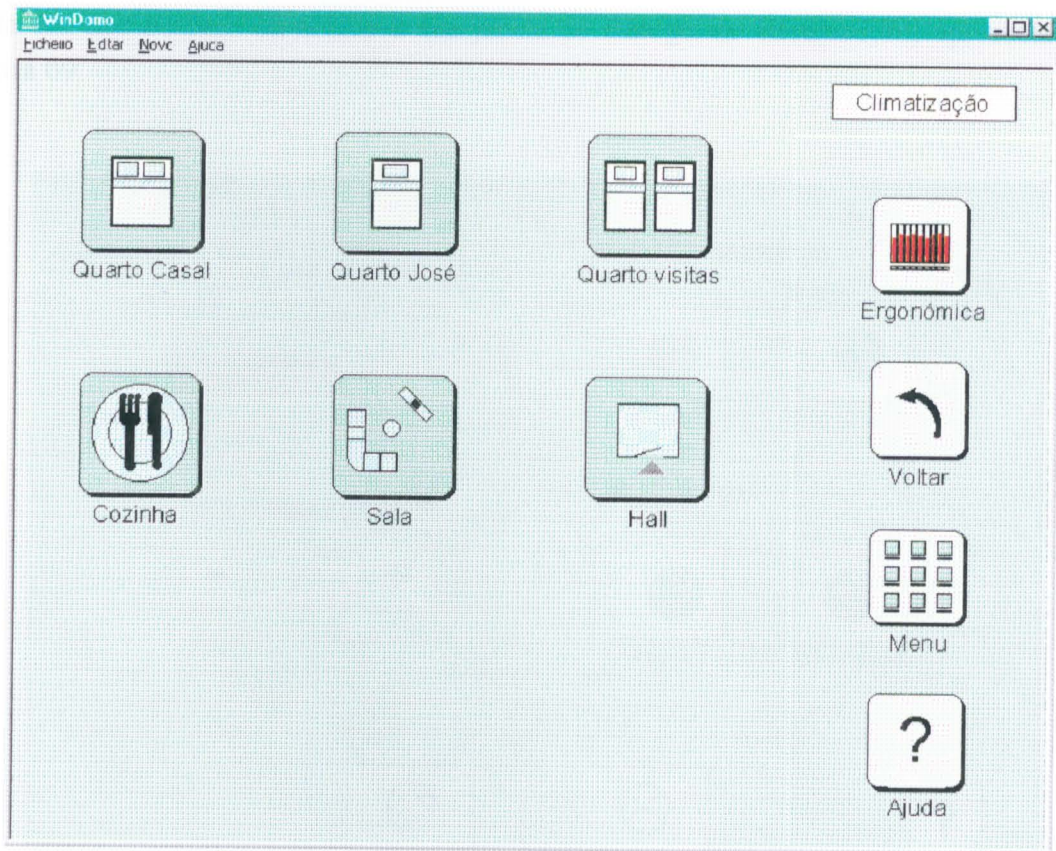


Figura 67 - O painel de "Climatização".

O botão “Ergonómica” situado na zona direita do painel permite ao utilizador, através da janela de diálogo apresentada na Figura 68, configurar temperaturas pretendidas diferentes ao longo do dia.

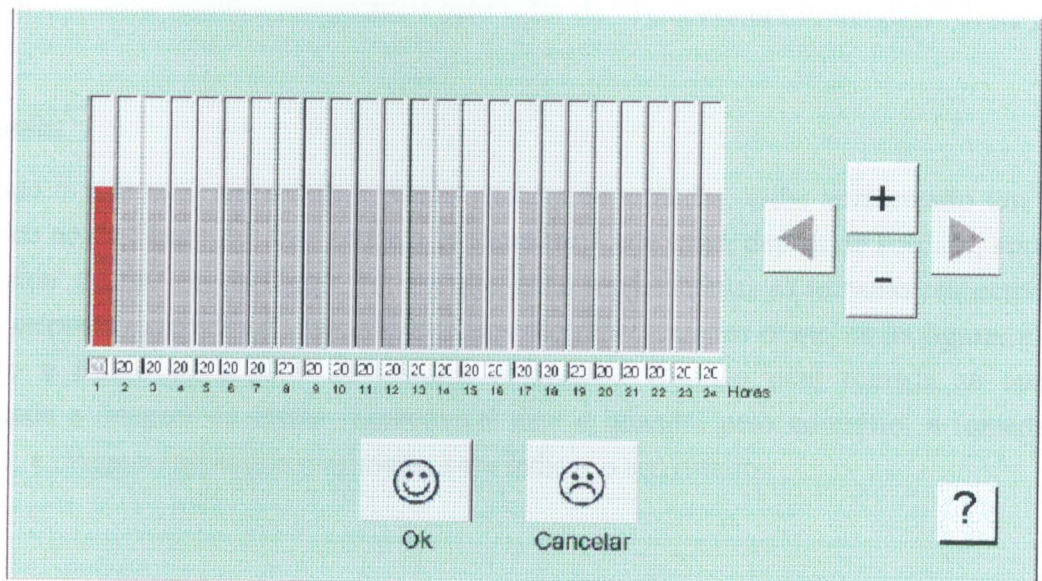


Figura 68 - A janela de diálogo para definir as temperaturas ao longo do dia.

Cada um dos botões dá acesso à sua janela de diálogo de climatização, apresentada na Figura 69.

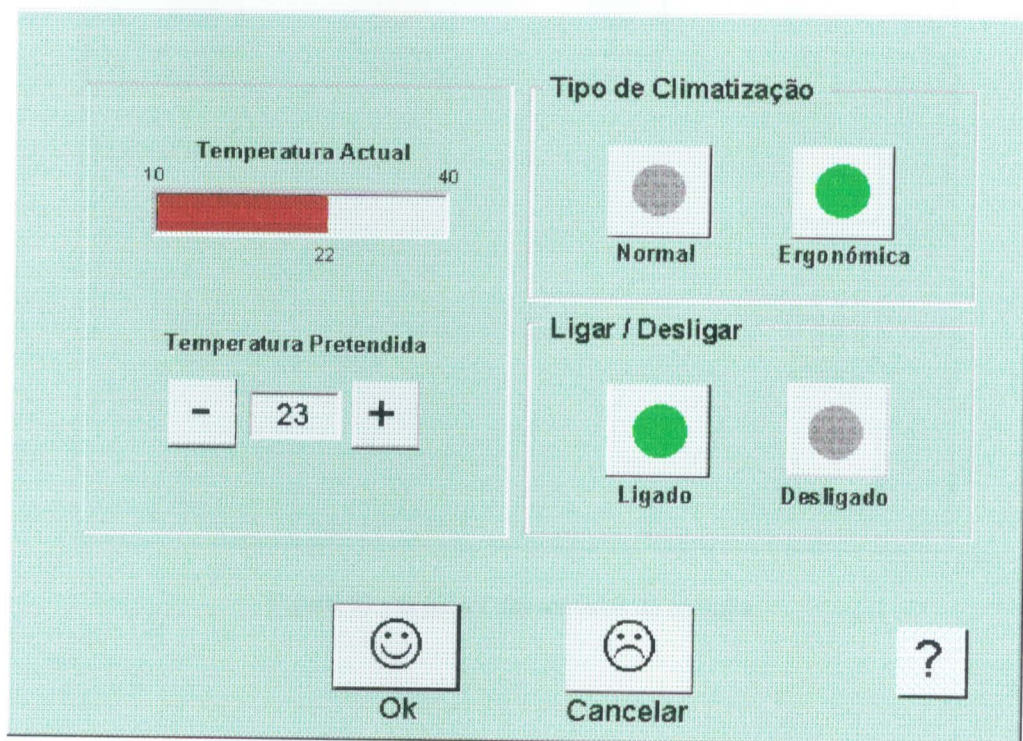


Figura 69 - Janela de diálogo para climatização de cada local.

Através desta janela de diálogo o utilizador pode seleccionar entre climatização ergonómica e normal. Se a climatização ergonómica for a escolhida o sistema regula a temperatura pelos valores escolhidos na janela de diálogo da Figura 68. Se a climatização escolhida for a normal o sistema regula a temperatura pelo valor de “Temperatura Pretendida” desta janela.

Tomadas

Quando o utilizador selecciona o botão relativo às tomadas, é-lhe apresentado um painel idêntico ao da Figura 63, mas relativo às tomadas. Cada local, depois de acedido, apresenta um painel do tipo do da Figura 70, em que estão representadas as tomadas respectivas. Se estes dispositivos forem do tipo simples eles são representados como os da figura, e o seu estado é fornecido através da imagem respectiva. Quando uma tomada se encontra desligada a imagem do botão respectivo é azul e quando, pelo contrário, a tomada está ligada, a imagem respectiva é vermelha para indicar o seu estado.

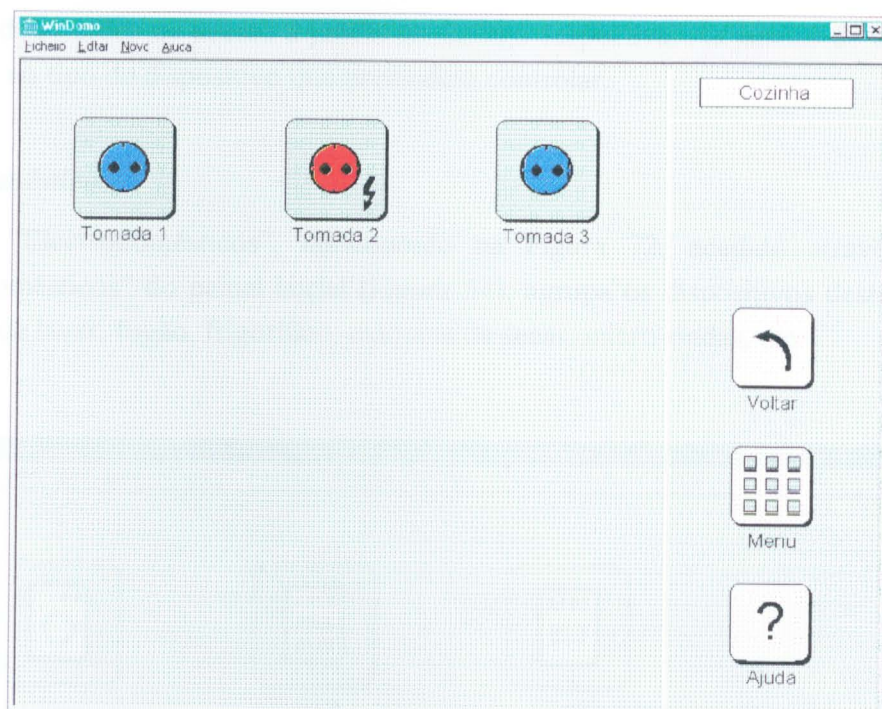


Figura 70 - O painel das tomadas da cozinha.

Lazer

O painel “Lazer” agrupa os dispositivos desta área, do tipo televisão, HiFi, VCR, etc. Quando acedido é apresentado um painel do tipo do da Figura 71.

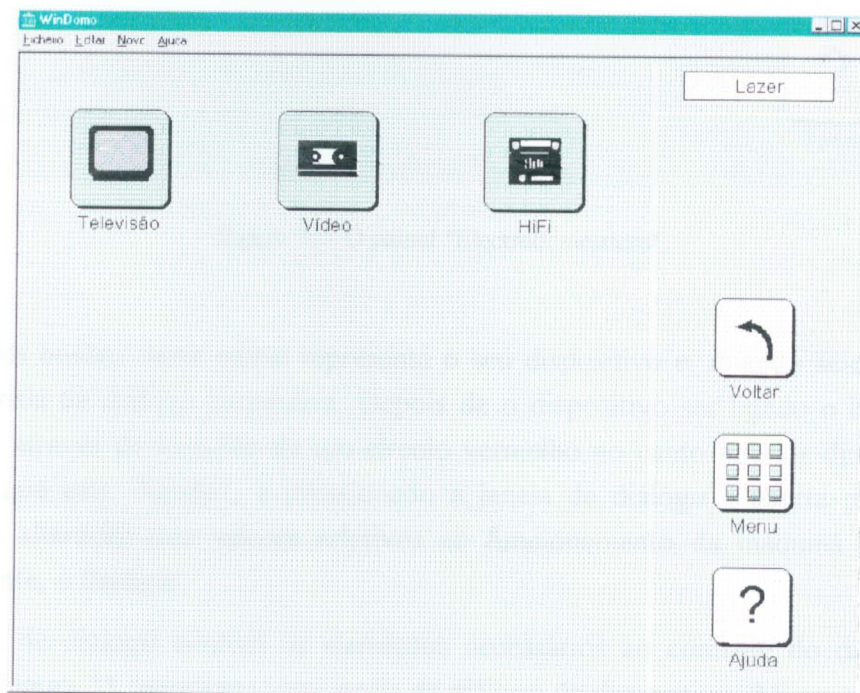


Figura 71 - O painel de "Lazer".

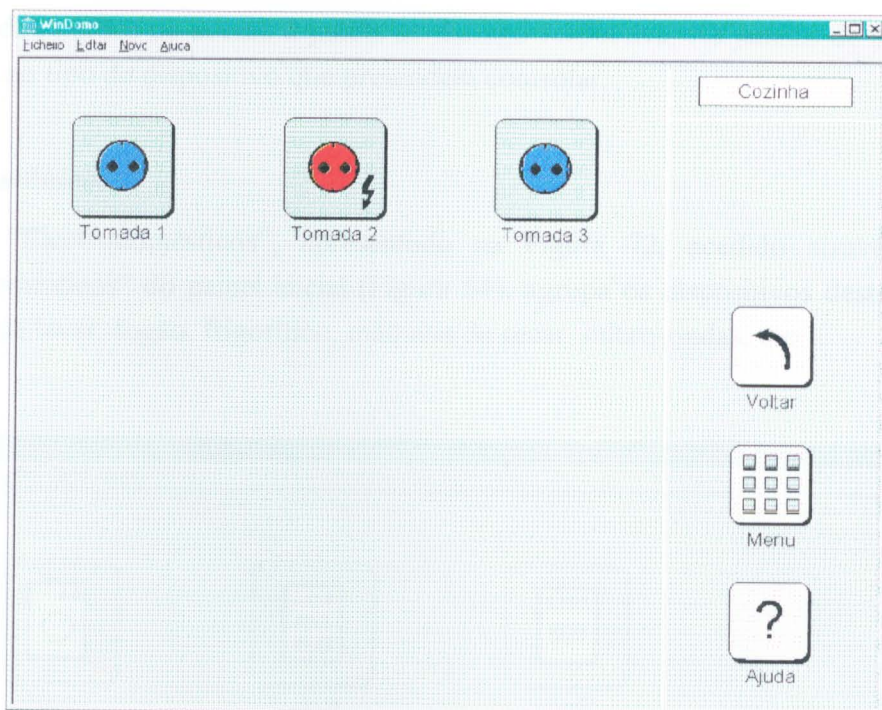


Figura 70 - O painel das tomadas da cozinha.

Lazer

O painel “Lazer” agrupa os dispositivos desta área, do tipo televisão, HiFi, VCR, etc. Quando acedido é apresentado um painel do tipo do da Figura 71.

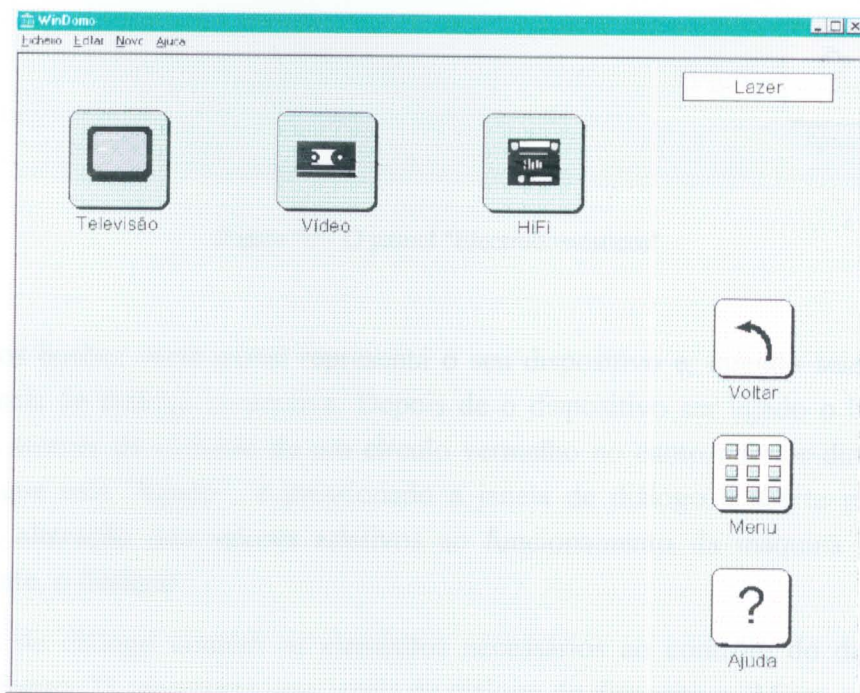


Figura 71 - O painel de "Lazer".

Cada um dos botões dá acesso à janela de diálogo respectiva. As janelas de diálogo serão adaptadas ao tipo de dispositivo que pretendem controlar.

Electrodomésticos

O painel “Electrodomésticos”, apresentado na Figura 72, acessado através do botão “Electrodomésticos” do painel inicial (Figura 54), agrupa os dispositivos deste tipo, como máquinas de lavar, fogão, frigorífico, máquina de secar, micro-ondas, etc.

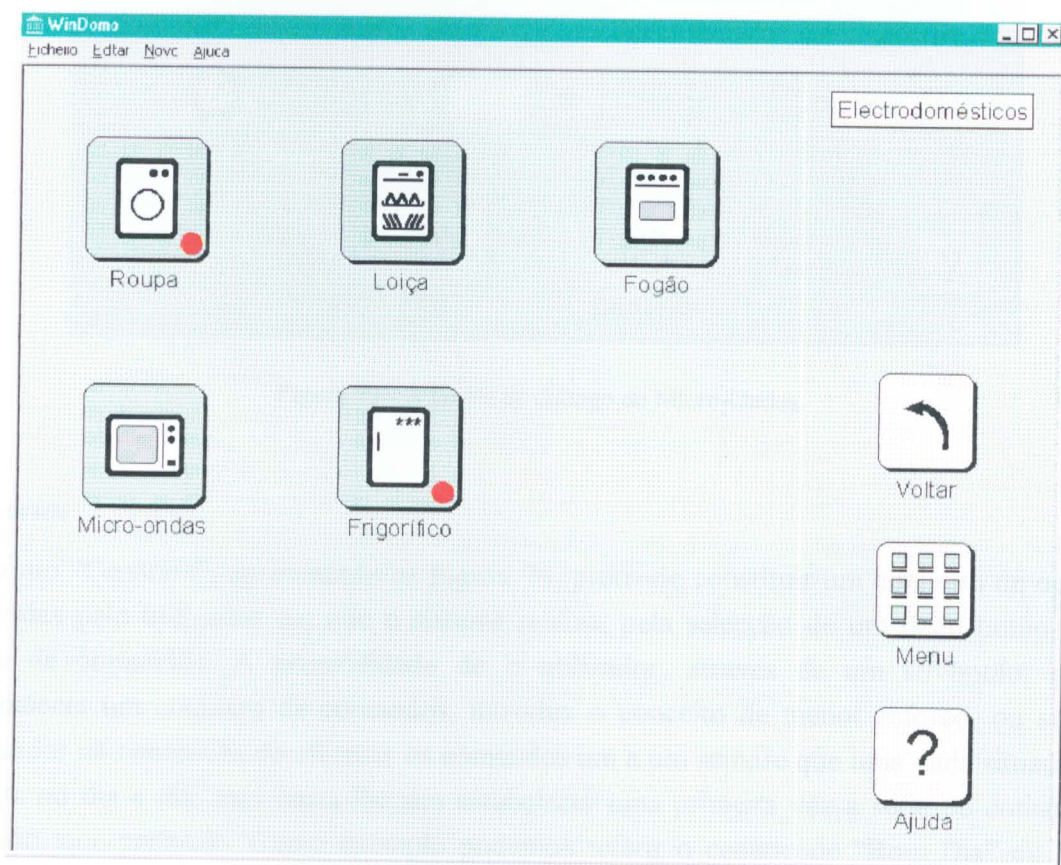


Figura 72 - O painel "Electrodomésticos".

Cada um dos botões deste painel representa o seu dispositivo e, quando seleccionado, dá acesso à janela de diálogo respectiva. Depois de o dispositivo ser ligado o botão reflecte esse estado através da inclusão de um círculo vermelho no canto inferior direito. Quando um botão, que está “ligado”, é pressionado a janela de diálogo é aberta permitindo ao utilizador a alteração dos valores relativos ao funcionamento da máquina respectiva e, inclusivamente, o desligar.

Cada janela de diálogo contém os elementos necessários ao controlo do dispositivo em questão. A Figura 73 representa uma janela de diálogo do dispositivo “Micro-ondas”.

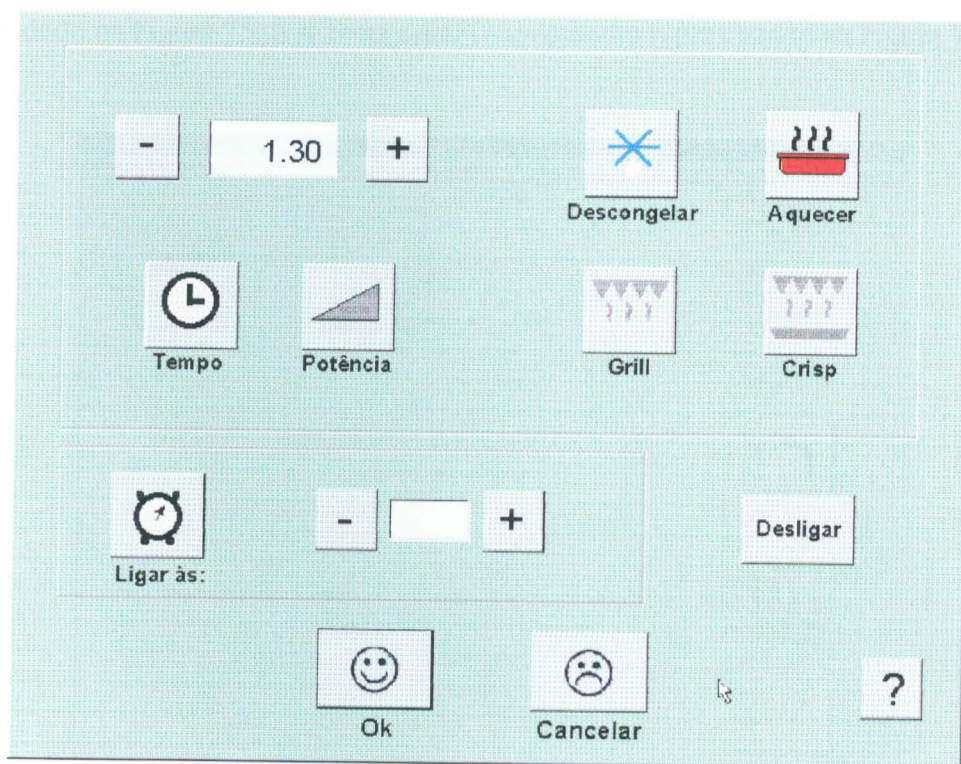


Figura 73 - A janela de diálogo do Micro-Ondas.

Cenários

O grupo “Cenários”, apresentado na Figura 74, pretende constituir um conjunto de opções definidas pelo utilizador em que o sistema realiza, pela selecção de um dado botão, uma série de comandos. A possibilidade de o utilizador, através de um só botão, poder estabelecer um conjunto de comandos, introduz o conceito de menor esforço, ou seja, o utilizador não necessita de efectuar os comandos um a um sempre que uma dada situação se repete no dia a dia, mas basta-lhe sim estabelecer uma primeira vez a série de comandos, cenário, que pretende. Como exemplo podemos referir o cenário de “Bom Dia” ou “Boa Noite”. Nestes dois cenários o utilizador estabelece o conjunto de comandos que pretende que o sistema efectue quando se levanta ou quando se deita. No primeiro podemos pensar em acções do tipo: abrir os estores, ligar a máquina do café, acender a luz da casa de banho, etc. No segundo podem ser incluídos eventos do tipo: fechar estores, desligar as luzes, ligar o alarme contra intrusos, fechar o portão, etc.

Esta série de comandos é estabelecida através da janela de diálogo da Figura 75, em que o utilizador escolhe os eventos a realizar e o tempo de espera até que eles se realizem. Este tempo de espera é necessário, em certos casos, para permitir ao utilizador sair de casa ou deitar-se. Se, por exemplo, o utilizador pressiona o botão referente ao cenário de férias, em que o sistema liga o alarme, a vontade dele será que este só esteja activo quando ele sair de casa.

Depois de o utilizador definir um novo cenário, o sistema apresenta o botão respectivo. No

exemplo dado na Figura 75, o sistema criou o novo botão representado na Figura 76.

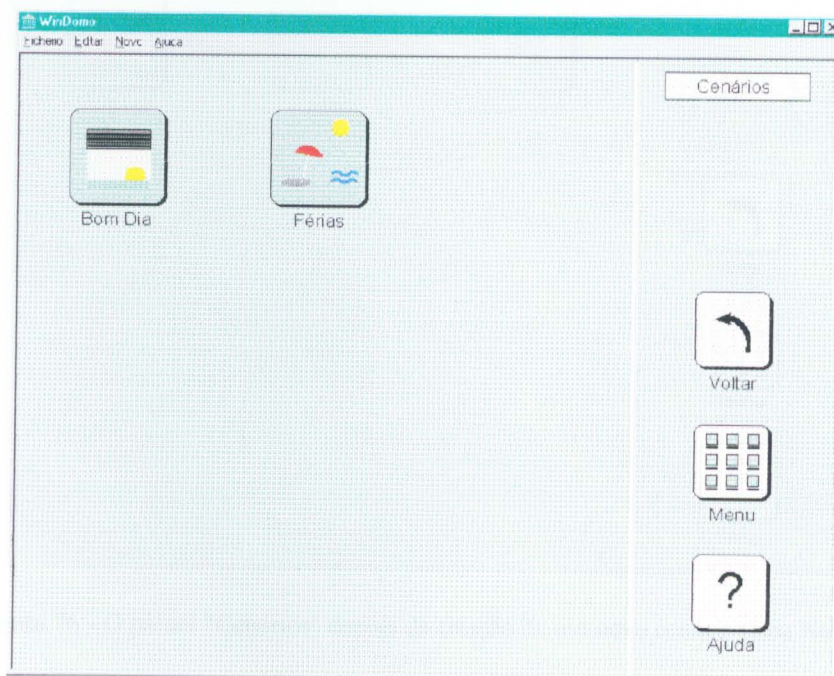


Figura 74 - O painel "Cenários".

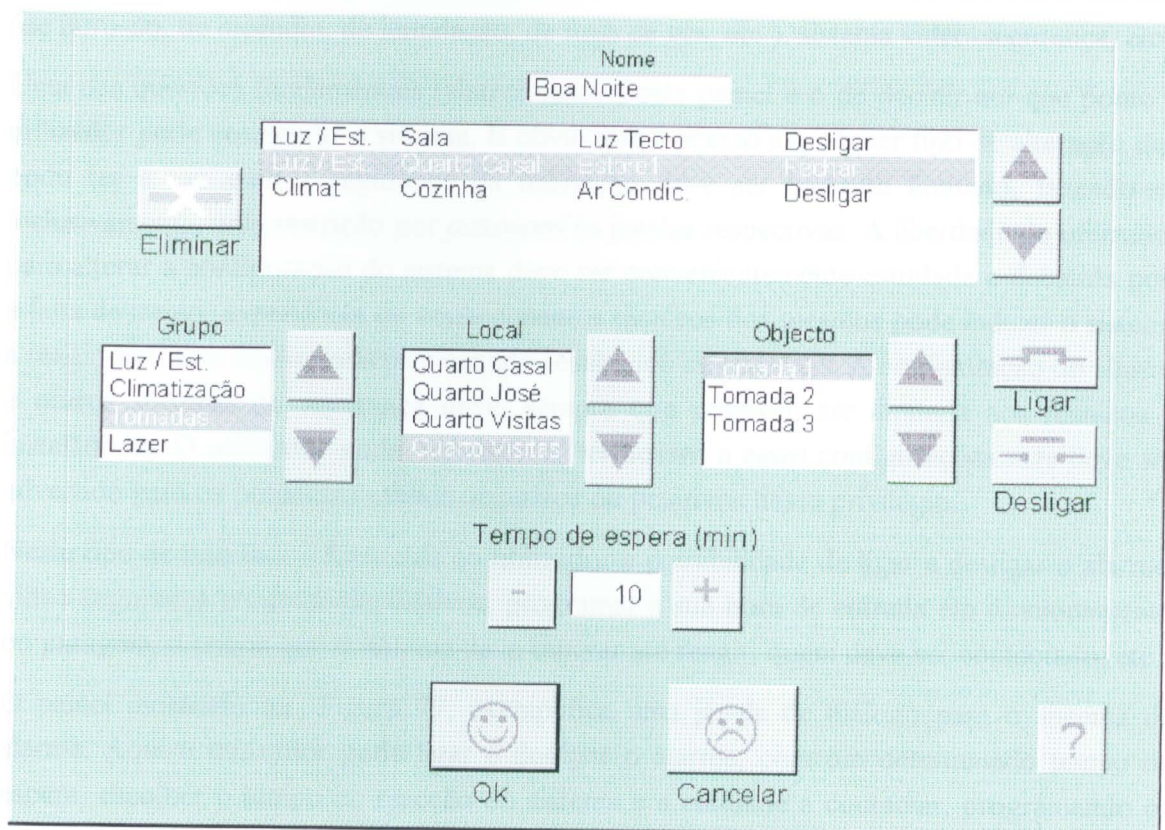


Figura 75 - A janela de diálogo para criar um novo cenário.

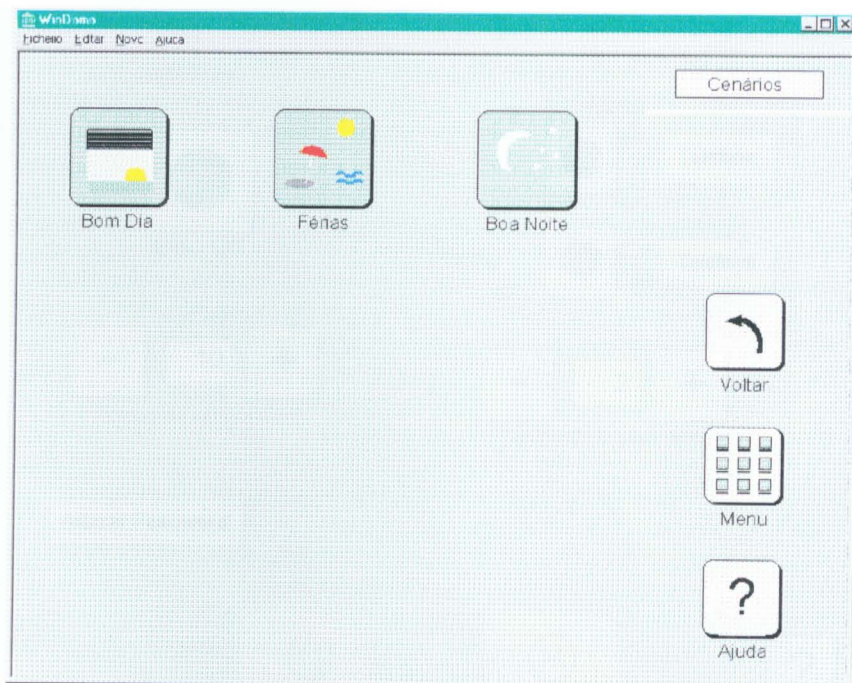


Figura 76 - O painel "Cenários" depois da criação de um novo cenário (Boa Noite).

Segurança

O painel de segurança englobará todos os elementos referentes a esta área, como alarmes (de intrusão, de incêndio, de inundação, de fuga de gás, etc.), sistema video-segurança, etc.

Uma das questões fundamentais relativamente a este painel é o de decidir até que ponto o utilizador pode configurar o sistema. É óbvio que o acesso a qualquer tipo de alteração que pode ter a ver com a segurança da habitação deve ser bastante limitado, fazendo-se, inclusivamente, uma restrição por *password* às janelas respectivas. A liberdade do utilizador para alterar a configuração do sistema deve ser convenientemente estudada e debatida pois a falta de treino, experiência ou conhecimentos técnicos dos mesmos pode induzir o sistema a reagir de forma inadequada provocando resultados indesejáveis. A configuração do alarme e outros sistemas de segurança deve, sempre que possível, ser limitada aos técnicos e instaladores. O utilizador pode, no entanto, ter acesso a essas configurações mas deve ser advertido para os potenciais efeitos negativos ou positivos desse privilégio.

Neste tipo de interface é fornecida ao utilizador a possibilidade de ligar e desligar o alarme, video-segurança e outros dispositivos, programar a sua hora de entrada em funcionamento ou paragem, o tempo que o sistema deve esperar até reagir, quem deve ser contactado, etc.

O painel mostrado na Figura 77, exemplifica uma janela de diálogo para o sistema de alarme. Aqui o utilizador pode ligar e desligar o alarme, com um determinado tempo de espera, escolher o tempo de reacção do sistema e a entidade a contactar, programando os respectivos números de telefone, e, ainda, alterar a *password* para acesso ao alarme e para desactivar o alarme.

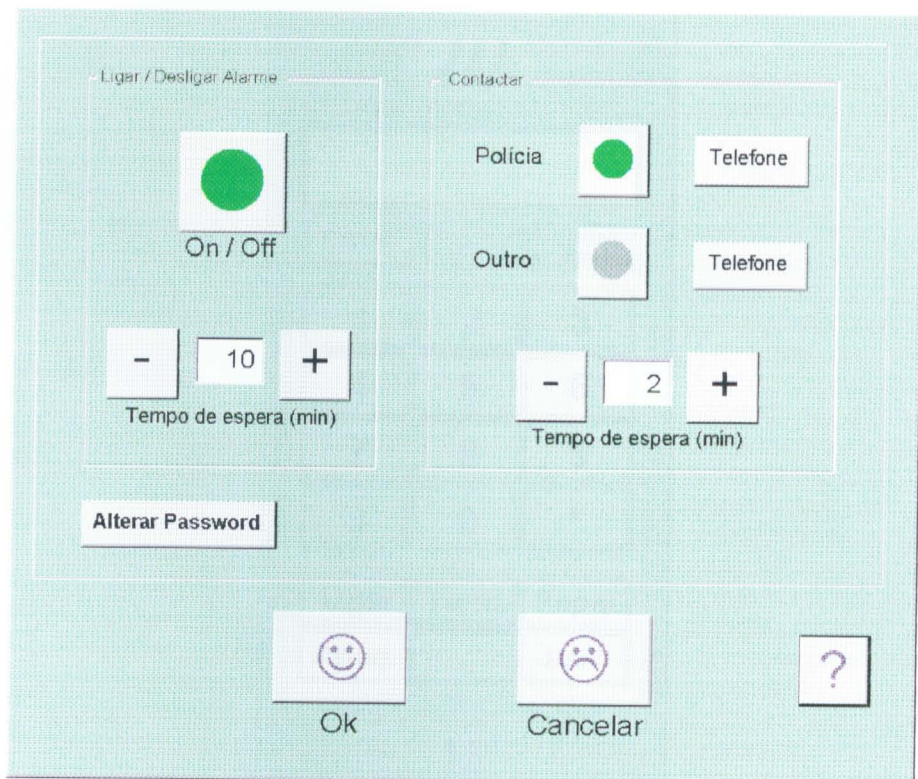


Figura 77 - A janela de diálogo de alarme contra intrusos.

Comunicações

O painel de comunicações engloba todos os objectos dessa área. Desde o telefone, passando pelo fax, modem, até a sistemas mais avançados de video-porteiro, internet, etc.

No painel, representado na Figura 78, apresentamos um exemplo de um botão representativo do telefone. A janela de diálogo respectiva está representada na Figura 79.

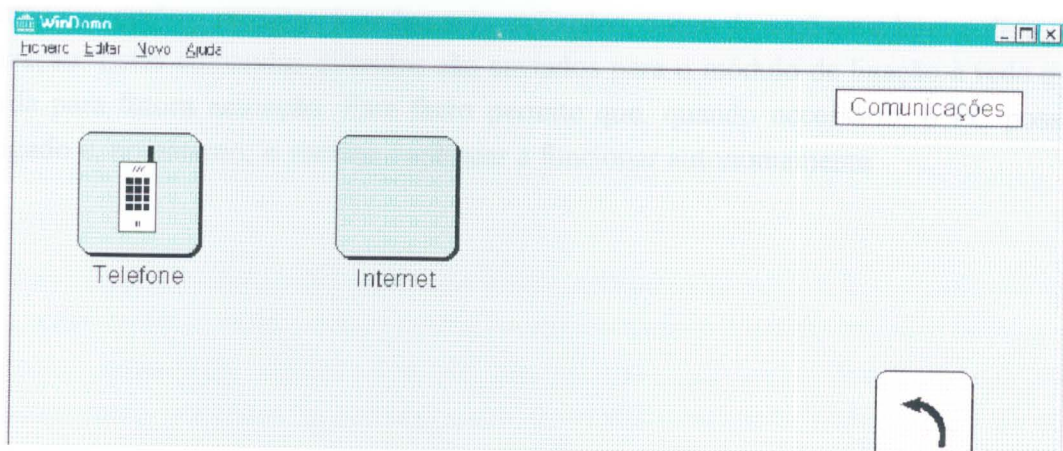


Figura 78 - Parte do painel de comunicações.



Figura 79 - A janela de diálogo do telefone.

Programação

O botão da programação, representado no painel inicial da Figura 54, permite aceder a uma janela onde o utilizador pode programar eventos diferidos, ou seja, comandos que o utilizador pretende que sejam efectuados num determinado instante que não o actual.

A respectiva janela de diálogo encontra-se representada na Figura 80, e mostra os vários eventos que podem ser seleccionados pelo utilizador, o tipo de acção e a hora e data para a realização. Estes comandos diferidos são enviados para o módulo de ligação à rede que os guarda para futura actuação. Este facto permite que, quando necessário, o PC possa ser desligado e, no entanto, o sistema continuar a funcionar autonomamente.

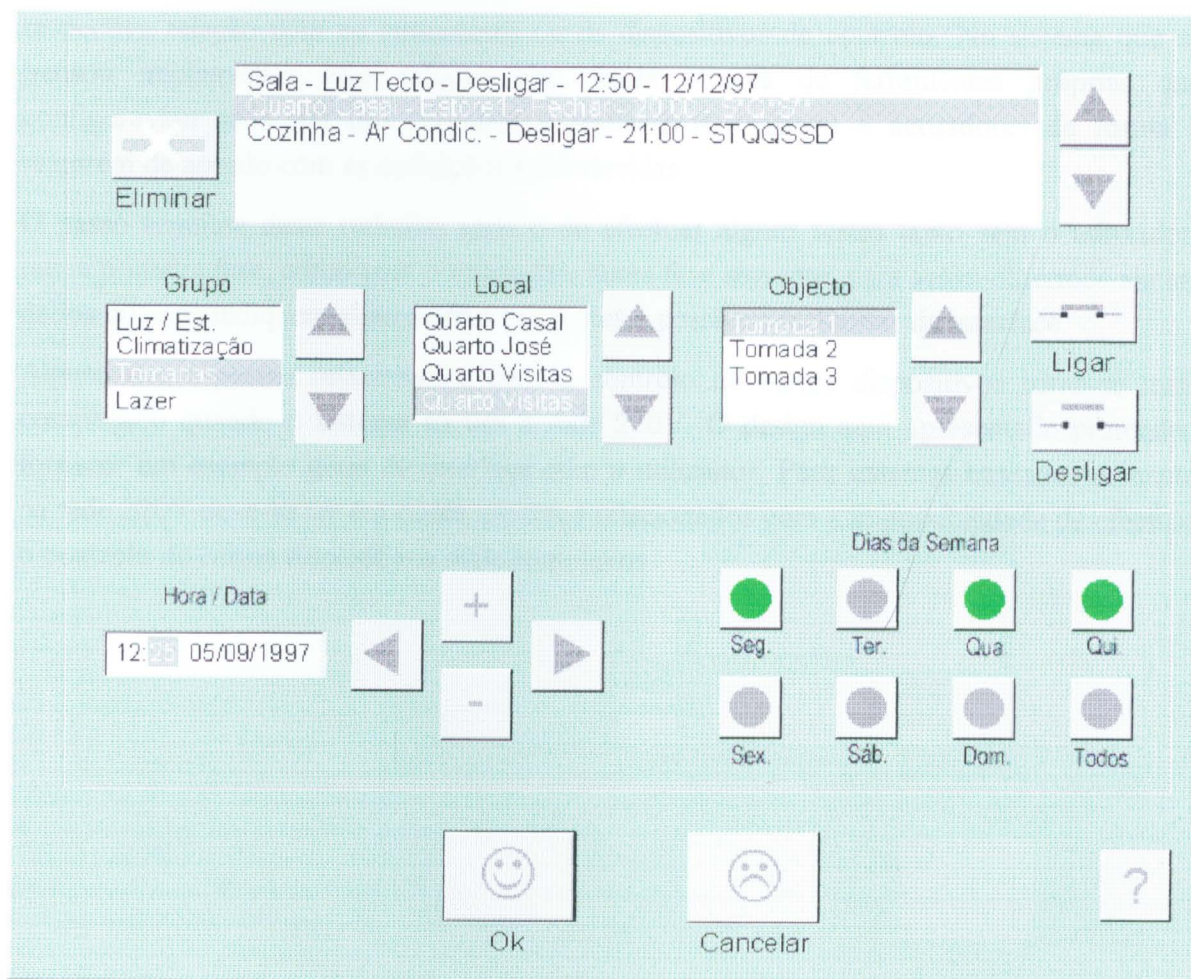


Figura 80 - A janela de diálogo de "Programação".

6.4 Considerações finais sobre a interface

Este protótipo pretende ser uma primeira abordagem ao *design* de uma interface com o utilizador para um sistema domótico, ele próprio com uma determinada autonomia. O sistema pode ser programado para funcionar sem o PC com a aplicação de controlo, podendo ser simplesmente programado para reagir a um determinado conjunto de acções derivadas dos parâmetros pré-estabelecidos e comparados com aqueles fornecidos pelos sensores.

Existem ainda algumas janelas que não foram apresentadas aqui e que dizem respeito às mensagens de informação, aviso e erro já mencionadas num capítulo anterior. Sempre que o utilizador estabelece valores fora do normal ou que sejam indutores de estados potencialmente perigosos o sistema reage enviando mensagens convenientes. O mesmo sucede quando o sistema detecta alguma situação anómala e pretende informar o utilizador desse facto.

Não foram abordadas também as características relativas à configuração dos sensores e a

respectiva relação com os actuadores ou os dispositivos de controlo. No sistema que se pensou implementar essa relação era obtida através de ferramentas próprias que configuravam cada um dos módulos e respectivos sensores e actuadores de forma a reagirem de acordo com as definições estabelecidas.

O passo seguinte deste trabalho seria o de efectuar alguns testes tanto sem o utilizador, numa primeira fase, como com o utilizador, numa fase seguinte, para assim obtermos alguns dados que nos indiquem alguns erros existentes e possíveis melhorias da interface.

Alguns dos aspectos relacionados com o controlo de certos dispositivos torna-se mais complicado quando o sistema só usa a rede 220V. O design aqui apresentado pretendeu fornecer um exemplo geral de interface com o utilizador. Para sistemas baseados somente na rede 220V deve-se ter em conta aspectos relacionados com a impossibilidade de efectuar o controlo de certos dispositivos mais complexos.

Capítulo 7

Conclusão

A Domótica é um domínio recente e em fase de expansão. Problemas que anteriormente foram identificados noutras situações vêm-se agora reencontrados nesta área. As interfaces com o utilizador, especialmente as interfaces gráficas, apresentam características próprias, e regras válidas para outros sistemas não se mostram as mais adequadas nesta área.

As disciplinas que abrangem a interação homem-máquina são bastante complexas e apresentam, ainda, muitas áreas de estudo e aperfeiçoamento. Os problemas referentes à forma como o utilizador reage a uma determinada interface pode determinar o seu sucesso ou fracasso pelo que as grandes empresas apostam cada vez mais no desenvolvimento suportado de aplicações, isto é, através de uma equipa pluridisciplinar, onde esteja integrado pelo menos um elemento da área do *design* de interfaces.

A interface com o utilizador, em termos gerais, entra numa nova era: a era da informação. As futuras aplicações vão, com certeza, demonstrar esta tendência e vamos assistir à introdução de novas interfaces com o utilizador que facilitam o trabalho de pesquisa, utilização e armazenamento da informação.

Com este trabalho pretendeu-se, acima de tudo, mostrar o que se pode fazer em termos de interfaces gráficas com o utilizador, de uma forma geral, e focalizou-se esse estudo para o

design de uma interface gráfica com o utilizador para um sistema domótico.

Apresentou-se uma breve descrição dos sistemas domóticos existentes, sua implantação ao nível do utilizador comum e desenvolvimento. Os Estados Unidos da América continuam a seguir alguns passos mais à frente dos seus rivais europeus e asiáticos e a tentativa da definição de um verdadeiro *standard* tem encontrado vários obstáculos, especialmente pelo individualismo e rivalidade crescente em actividades deste tipo. O HES, sistema da Siemens, a ser difundido brevemente, apresenta características de interface que nos parecem poder constituir um avanço real neste campo.

A forma como se processou, em termos históricos, a adaptação das interfaces com o utilizador foi apresentada neste trabalho, e um dos factores principais, a apresentação da informação, constituiu um dos subcapítulos mais produtivos e frutuosos desta dissertação, talvez ao mesmo nível dos princípios de concepção das interfaces gráficas, que apresentaram as regras fundamentais no desenvolvimento das interfaces e efectuaram uma primeira intervenção na área da domótica.

Com o capítulo referente aos requisitos da interface com o utilizador para sistemas domóticos orientou-se, finalmente, o leitor para a área de especialização deste trabalho. Os aspectos relacionados com a familiaridade do utilizador, com os objectos utilizados e o seu significado, constitui um dos principais factores na especificação de uma interface com o utilizador em sistemas domóticos.

Fundamentou-se, com o capítulo referido anteriormente, o *design* da interface gráfica com o utilizador apresentada no capítulo final. Este *design* foi efectuado tendo em vista a sua apresentação para controlo do sistema através de um *touch-screen* e os objectos de interacção utilizados reflectem esse princípio: estes devem ser pensados de forma a permitir ao utilizador, de preferência sem o uso do rato ou teclado, efectuar o controlo do sistema. Convém referir que a primeira parte de construção da interface, pelo utilizador ou instalador, não obedeceu propositadamente a este princípio, por constituir, por si só, uma pequena parte no conjunto da utilização da aplicação.

Trabalho futuro

O passo seguinte, resultado do estudo efectuado por este trabalho, será o de realizar os primeiros testes da interface, tanto sem a intervenção do utilizador como com a participação deste. Este tipo de testes e a forma como devem ser realizados foram apresentados neste trabalho e devem ser conduzidos com extremo cuidado e sensibilidade para que os resultados obtidos possam ser proveitosos para a melhoria e aperfeiçoamento da interface final. Para testar de forma eficiente a interface, além de seguir as regras apresentadas, introduziram-se, no mesmo protótipo, diferentes soluções de maneira a acelerar o processo iterativo. Exemplo deste procedimento são as formas diversas como na Figura 69 e Figura 77 se efectua o ligar e desligar de um dispositivo: no primeiro caso com dois botões e no segundo somente com um botão.

Depois de termos aferido as características da interface, passamos à implementação da funcionalidade da aplicação sobre a qual aquela vai funcionar. A aplicação suportada pela interface realizada deve tentar seguir as especificações indicadas, ou seja, possíveis limitações ao nível da linguagem de programação não devem coarctar a forma como a informação é apresentada ao utilizador.

Os aspectos relacionados com a definição do tipo de protocolo, trama de comunicações, serviços fornecidos pelos dispositivos do sistema e outros relativos à interligação com o Domo220 devem ser amplamente discutidos e especificados para que o sistema final constitua um verdadeiro sistema domótico. Relativamente a estes aspectos podemos adiantar que, conforme referimos no capítulo anterior, o facto de implementarmos o sistema na rede 220V limita substancialmente o poder de controlo sobre dispositivos mais complexos. Este será um dos aspectos que constitui maior factor de reflexão pois a possibilidade da colocação de um barramento complementar para um controlo mais específico deve ser bem pensada.

Algumas outras características como a utilização de elementos Plug and Play que facilitam a instalação de novos dispositivos, a possível utilização de sistemas já existentes para efectuar a interface com o utilizador (caso da televisão e telecomando) ou ainda a adaptação do sistema a utilizadores discapacitados são, também, indicativas de que, nesta área, os factores são tão diversos que nos podem remeter para a constituição de sistemas menos abrangentes.

Penso que o trabalho futuro, de qualquer forma, se encontra facilitado, pois este estudo efectuado sobre as interfaces gráficas com o utilizador tentou adiantar e esclarecer alguns dos pontos fundamentais da interacção homem-sistema.

Referências Bibliográficas

- [Alben96] Alben, Lauralee
Defining the Criteria for Effective Interaction Design
Interactions, pp.11-15, May-June 1996.
- [Anderson97] Anderson, Tim
Ferramentas de Programação Visual
Personal Computer World, pp.58-76, Maio 1997.
- [Anthony96a] Anthony, Christopher R.
Color Guide,
NCR Corporation, Retail Systems Group, Atlanta, 1996.
- [Anthony96b] Anthony, Christopher R.
Color: To Use or Not To Use,
NCR Corporation, Retail Systems Group, Atlanta, 1996.

- [Apple92] Apple Computer, Inc.
Macintosh Human Interface Guidelines,
Addison-Wesley, 1992.
- [Laurel90] Laurel, Brenda
The Art of Human-Computer Interface Design,
Addison-Wesley, 1990.
- [Baptista97] Baptista, Ivan C.
TeleDomus - Um Sistema Domótico Programável com Acesso Remoto,
Tese de Mestrado, Instituto Superior Técnico, Lisboa, 1997.
- [Brock96] Brock, John F.
Whose Metaphor?,
Interactions, pp.25-29, July-August 1996.
- [Brown97] Brown, Judy
Exploring Human-Computer Interaction and Software Engineering Methodologies for the Creation of Interactive Software,
SIGCHI Bulletin, 29 (1), pp. 32-35, January 1997.
- [Cesta96] Cesta, Amedeo e D'Aloisi, Daniela
Building Interfaces as Personal Agents - A Case Study,
SIGCHI Bulletin, 28 (3), pp. 108-113, July 1996.
- [Christ75] Christ, R. E.
Review and Analysis of Color Coding Research for Visual Displays
Human Factors, 17 (6), 542-570, 1975.
- [Dubrau96] Dubrau, Gunter
The Attributistical Understanding of Information
SIGCHI Bulletin, 28 (4), pp. 79-84, October 1996.
- [DSTL-95-003] Data Systems Technology Division / Code 520
User-Interface Guidelines,
NASA, Goddard Space Flight Centre, Greenbelt, Maryland, 1997.
- [Erickson90] Erickson, Thomas D.
Working with Interface Metaphors,
Em "The Art of Human-Computer Interface Design", edit. Brenda Laurel,
Addison-Wesley, 1990.

- [FACE94] ESPRIT Project 6994
Guidelines and Rules for Design of User Interfaces for Electronic Home Devices,
1994.
- [Ford97] Ford, Shannon e Marchak, Frank M.
The Future of Visual Interaction Design,
SIGCHI Bulletin, 29 (1), pp. 10-11, January 1997.
- [Gomoll90] Gomoll, Kathleen
Some Techniques for Observing Users,
Em “The Art of Human-Computer Interface *Design*”, edit. Brenda Laurel,
Addison-Wesley, 1990.
- [Gray96] Gray, Phil e Johnson, Chris
Requirements for the Next Generation of User Interface Specification Languages,
Electronic Paper, 1996.
- [Gulliksen95a] Gulliksen, Jan e Sandblad, Bengt
Domain-specific Design of User Interfaces,
International Journal of Human-Computer Interaction, 7 (2), pp. 135-151,
Ablex Publishing Corporation, Norwood, New Jersey, 1995.
- [Gulliksen95b] Gulliksen, Jan e Sandblad, Bengt e Lind, Mats
The Nature of User Interface Design - The Role of Domain Knowledge,
Electronic paper, 1995.
- [Gulliksen95c] Gulliksen, Jan e Lif, Magnus e Sandblad, Bengt
Work Spaces and Documents in User Interface Design
Electronic paper, 1995.
- [Gunton92] Gunton, Tony
Dictionary of Information Technology and Computer Science
Penguin Books, 1992.
- [Halfhill97] Halfhill, Tom R.
Good-Bye, GUI. Hello, NUI.
Byte, pp. 60-72, July 1997.
- [Heimer95] Heimer, Thomas
A Comparison of Diferent User-Interfaces for Intelligent Homes,
Resumo de palestra apresentada no INESC, Abril 1995.

- [EHSA92] European Home Systems Association
Home Systems Specification, Release 1.1,
ESPRIT Project 5448, Home Systems Consortium
- [Kay90] Kay, Alan
User Interface: A Personal View
Em “The Art of Human-Computer Interface *Design*”, edit. Brenda Laurel,
Addison-Wesley, 1990.
- [Kobara91] Kobara, Shiz
Visual Design with OSF/Motif,
Addison-Wesley, 1991.
- [Lewis93] Lewis, Clayton e Rieman, John
Task-Centred User Interface Design, A Practical Introduction,
Authors Electronic Edition, 1993.
- [Mahajan95] Mahajan, Rohit e Shneiderman, Ben
A Family of User Interface Consistency Checking Tools,
Human-Computer Interaction Laboratory, University of Maryland, 1995.
- [Marcos95] Marcos, Maria da Graça
*O Processo de Desenvolvimento de Interfaces Gráficas de Utilizador -
Uma Aplicação,*
Tese de Mestrado, Faculdade de Engenharia, Porto, Setembro de 1995.
- [May95] May, Jon e Scott, Sophie e Barnard, Phil
Structuring Displays: a Psychological Guide,
AMODEUS Project, ESPRIT Basic Research Action 7040, 1995.
- [Mikes92] Mikes, Steven
X Window System Program Design and Development,
Addison-Wesley, 1992.
- [MIL-STD-1472C] Department of Defense
*Military Standard: Human Engineering Design Criteria for Military
Systems, Equipments and Facilities,*
Washington D.C., 1983.
- [Myers95] Myers, Brad A.
User Interface Software Tools,
ACM Transactions on Computer-Human Interaction, 2(1), pp. 64-103,
March 1995

- [Myers96] Myers, Brad A.
A Brief History of Human Computer Interaction Technology
Technical Report CMU-CS-96-103, Carnegie Mellon University, 1996.
- [Myers97] Myers, Brad A.
List of User Interface Software Tools,
<<http://www.cs.cmu.edu/afs/cs.cmu.edu/users/bam/www/toolnames.html>>,
1997
- [Negroponte90] Negroponte, Nicholas
Hospital Corners,
Em “The Art of Human-Computer Interface *Design*”, edit. Brenda Laurel,
Addison-Wesley, 1990.
- [Nelson90] Nelson, Theodor H.
The Right Way to Think About Software Design,
Em “The Art of Human-Computer Interface *Design*”, edit. Brenda Laurel,
Addison-Wesley, 1990.
- [Nielsen93] Nielsen, Janni e Aboulaflia, Annette
Designing User Interfaces - the Role of Intuition and Imagination,
Interchi'93, Conference on Human Factors in Computing Systems,
Amsterdam, The Netherlands, pp. 209-210, April 1993.
- [Nonnen97] Nonnen, Klaus
Let's Go - The first Introductory Packages for HES are now Available,
Product Profile, Siemens, April 1997.
- [Norman90] Norman, Donald A.
Why Interfaces Don't Work
Em “The Art of Human-Computer Interface *Design*”, edit. Brenda Laurel,
Addison-Wesley, 1990.
- [NP3003] Norma Portuguesa NP 3003
Computadores e Tratamento da Informação: Vocabulário,
Novembro de 1993.
- [Nunes95] Nunes, Renato J.C.
Integração de Serviços para Edifícios Inteligentes,
Tese de Doutorado, Instituto Superior Técnico, Julho de 1995.
- [OPEN LOOK90] Sun Microsystems, Inc.
Graphical User Interface Application Style Guidelines,
Addison-Wesley, 1990.

- [Rauch96] Rauch, Thyra, Kaller, Susan e Flanagan, George
Usability Management Maturity, Part 2: Usability Techniques - What Can We Do?
SIGCHI Bulletin, 28 (4), pp. 63-66, October 1996.
- [Rubinstein93] Rubinstein, Michel
L'Impact de la Domotique sur les Fonctions Urbaines,
Fondation Européenne pour l'Amélioration des Conditions de Vie et de Travail, 1993.
- [Rouff96] Rouff, Christopher
Formal Specification of User Interfaces
SIGCHI Bulletin, 28 (3), pp. 27-33, July 1996.
- [Salomon90] Salomon, Gitta
New Uses for Color,
Em "The Art of Human-Computer Interface Design", edit. Brenda Laurel,
Addison-Wesley, 1990.
- [Santos94] Santos, Jorge
Desenvolvimento de uma Aplicação para Gestão de Habitação,
Relatório de Projecto de Fim de Curso, Instituto Superior de Engenharia do Porto, 1994.
- [Santos95] Santos, Cristina Russo
Interfaces Gráficas com o Utilizador - Contribuição para a Concepção em Ambientes de Fabrico.
Tese de Mestrado, Faculdade de Engenharia, Porto, 1995.
- [Sastry93] Sastry, Lakshmi
Graphical User Interface Development Tools
Visualization Group, Informatics Department, DRAL, 1993.
- [Sellen90] Sellen, Abigail e Nicol, Anne
Building User-centred On-line Help,
Em "The Art of Human-Computer Interface Design", edit. Brenda Laurel,
Addison-Wesley, 1990.
- [Shneiderman92] Shneiderman, Ben
Designing de User Interface: Strategies for Effective Human-Computer Interaction,
Addison-Wesley, 1992.

- [Shubin96] Shubin, Falk e Johansen
Exploring Color in Interface Design,
Interactions, pp.37-47, July-August 1996.
- [Smith86] Smith, Sidney L. e Mosier, Jane N.
Guidelines for Designing User Interface Software,
Report ESD-TR-86-278, MITRE Corporation, Massachusetts, 1986.
- [Swaminathan97] Swaminathan e Sato
Interaction Design for Large Displays,
Interactions, pp.15-24, January-February 1997.
- [TEAM97] HOME TEAM
Home Networks: Control Standards,
<<http://www.hometeam.com/homenet/home.htm>>, 1997.
- [Tognazzini90] Tognazzini, Bruce
Consistency,
Em “The Art of Human-Computer Interface *Design*”, edit. Brenda Laurel,
Addison-Wesley, 1990.
- [Vanderdonckt93] Vanderdonckt, Jean
A Corpus of Selection Rules for Choosing Interaction Objects,
Technical Report 93/3, Project Trident, Intitut d’Informatique, 1993.
- [Vanderdonckt94] Vanderdonckt, Jean e Gillo, Xavier
Visual Techniques for Traditional and Multimedia Layouts,
Proceedings of 2nd Workshop on Advanced Visual Interfaces, AVI’94,
ACM Press, June 1994.
- [Wagner90] Wagner, Annette
Prototyping: A Day in the Life of the Interface Designer,
Em “The Art of Human-Computer Interface *Design*”, edit. Brenda Laurel,
Addison-Wesley, 1990.
- [Wilson95] Wilson, Stephanie e Johnson, Peter e Johnson, Hilary
User Participation in Task-based Design,
Position Paper for CHI’95 Research Symposium, Denver, Colorado, 1995.
- [Wertheimer58] Michael Wertheimer
Principles of Perceptual Organization,
Em “Studies in Perception”, edit. Gerald M. Murch, The Bobbs-Merrill
Company, Inc, 1976.

- [Yankelovich96] Yankelovich, Nicole
Know What to Say?,
Interactions, pp.33-43, November-December 1996.
- [Yeo96] Yeo, Alvin
Cultural User Interfaces, A Silver Lining in Cultural Diversity,
SIGCHI Bulletin, 28 (3), pp. 4-8, July 1996.

Anexos

Anexo A

O protocolo de comunicações na rede está a ser implementado no âmbito de um projecto de mestrado desenvolvido por outro mestrando desta mesma edição do curso e os dois trabalhos, serão interligados por forma a constituir um sistema domótico completo.

Este anexo apresenta a trama de comunicação entre o PC e o módulo de rede.

O sistema domótico idealizado, já referido no capítulo 6, terá a estrutura representada na Figura 81.

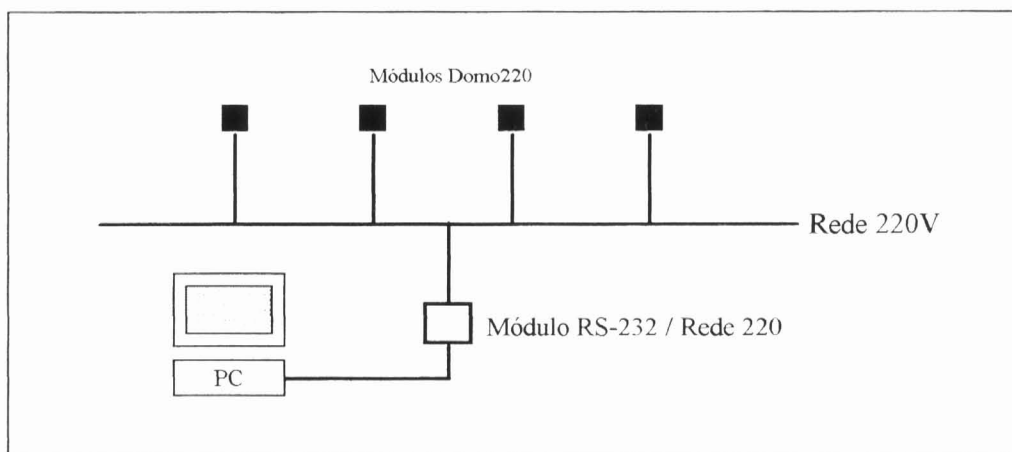


Figura 81 - A estrutura do sistema domótico.

As comunicações a realizar entre o PC e o módulo de rede seguem a filosofia do EHS em que uma dada acção de um determinado serviço é requisitada a um determinado dispositivo.

A trama da comunicação do PC para o módulo é constituída por vários campos, totalizando 25 Bytes. Os campos estão representados na Figura 82.

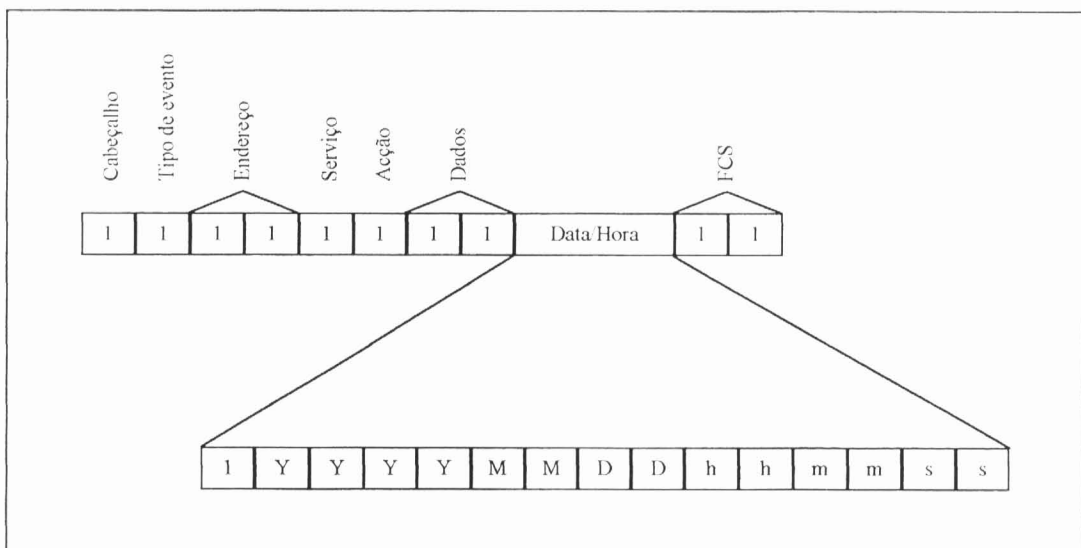


Figura 82 - O formato da trama da comunicação PC-Módulo

Os campos da trama de envio de comando têm as seguintes funções:

Cabeçalho

Campo que inicia a comunicação.

Tipo de evento

O campo “Tipo de evento” definirá o tipo de evento a realizar: Imediato, se a acção tiver que ser realizada na altura do pedido; Diferido, se a acção for realizada num determinado instante, definido pela data e hora, posterior ao pedido; Programação do relógio do módulo, este evento serve para programar o relógio de tempo real do módulo; Periódico, se o evento tiver que ser realizado em determinados dias da semana (por exemplo, todas as Quartas, Quintas e Sextas-feiras a uma determinada hora).

Endereço

Este campo conterà o endereço do módulo onde se encontra o serviço a que se destina o comando enviado.

Serviço

O campo de “Serviço” especifica que tipo de serviço de um determinado objecto é requisitado (por exemplo, o serviço de zoom de uma *camcorder*).

Acção

O campo de “Acção” especifica que tipo de acção se pretende efectuar sobre o serviço indicado no campo anterior (por exemplo, aumentar).

Dados

Este campo de dados é utilizado quando a acção pretendida é acompanhada de um determinado valor (por exemplo, 2 vezes).

Data/Hora

O campo de “Data/Hora” especificará a data e a hora em que a acção é requisitada. Este campo só é lido quando o evento é do tipo temporizado, programação ou periódico. Quando o evento é do tipo imediato este campo não é lido.

FCS

Campo para controlo de erros.

A trama de resposta do módulo para o PC será extremamente mais simples. Optou-se por efectuar uma resposta curta que informasse a aplicação se o comando enviado foi ou não recebido e se foi, quando é esse o caso, efectuado.

A trama de resposta encontra-se representada na Figura 83.

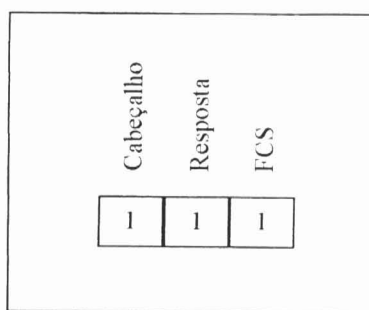


Figura 83 - Formato da trama de resposta.

Anexo B

Este anexo apresenta uma lista obtida através do agente “The Informant”, abordado no capítulo 2, e que pesquisa endereços na Web de acordo com uma série de palavras chaves introduzidas pelo utilizador. Este serviço é gratuito e qualquer pessoa pode aceder a ele.



The Informant

Your personal search agent, the Informant

- [Home](#)
- [Questions?](#)
- [Advertising](#)
- [Licensing](#)
- [Reviews](#)
- [Contact Us](#)
- [Auto login](#)
- [Edit account](#)
- [Remove account](#)
- [Change password](#)

WHAT IS THE INFORMANT?

The Informant is a **FREE** service that will save your favorite search engine queries and web sites, check them periodically, and send you email whenever there are new or updated web pages.

If you do not have an account, sign up today!

[Click here to sign up!](#)

WELCOME BACK! LOG IN HERE:

Username:

Password:

Remember your password

Log in automatically

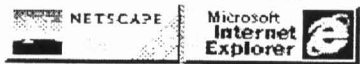
Powered by:

- [Lycos](#)
- [AltaVista](#)
- [Excite](#)
- [Infoseek](#)

[Link to us from your page!](#)



Best viewed with:



© 1995-1997 : Push Technologies

Now serving 13824 customers.



The Informant

Your personal search agent on the internet

Results summary

Your searches were last run on Saturday, September 20, 1997.

[New Results](#)

[All Results](#)

[Edit account](#)

Details	Query	Summary
Query 1	intelligent buildings (OR)	6 new, 0 updated, 2 unreachable
Query 2	building automation systems (OR)	8 new, 0 updated, 0 unreachable
Query 3	home automation (OR)	10 new, 0 updated, 0 unreachable

- [Main page](#)
- [Questions?](#)
- [Contact us](#)
- [Autologin](#)
- [Change password](#)
- [Remove account](#)
- [Advertising](#)
- [Licensing](#)
- [Reviews](#)

Thank you for using the *Informant*. Please come again!

©1995-1997 Push Technologies

[Questions?](#) [Comments?](#)

Write us!



The Informant

Your personal search agent on the internet

Results - Query 1

Query

Query text: *intelligent buildings*

Summary:

Query logic	Search engine	Last searched date
match some terms (OR)	Excite	Saturday, September 20, 1997

Summary

Do you need *an explanation of the results table?*

Edit account

- [Main page](#)
- [Questions?](#)
- [Contact us](#)
- [Autologin](#)
- [Change password](#)
- [Remove account](#)

Rank	Status	Site
1/10	new	Intelligent Buildings http://www.public.asu.edu/~jyr705/it_int.html
2/10	new	The Computer Information Centre (CompInfo) - the definitive directory ... http://www.compinfo.co.uk/index.htm
3/10	new	HOME TEAM: Advanced Search Site http://www.hometeam.com/index.htm
6/10	new	Intelligent Buildings http://www.fmbooks.com/library/intellbldg/intellbldg
7/10	new	INTELLIGENT BUILDINGS http://www.np.ac.sg/~tcc1/ib.htm
9/10	new	Designed and constructed Intelligent Architecture http://www.tue.nl/lava/people/rob/ia-world.htm
5/10	old	Unit for Architectural Studies Home Page http://doric.bart.ucl.ac.uk/web/UAS/UASHOME.HTM
8/10	old	Intelligent Buildings http://www.emrinc.com/ib.htm
4/10	⊗	ProPlan Home Page http://www.proplan.co.uk/index.htm
10/10	⊗	Untitled http://www.en.polyu.edu.hk/~hkie/pinkpage/jw1_1.htm

Push Technologies is not responsible for the contents of third-party web sites

Thank you for using the *Informant*. Please come again!

©1995-1997 Push Technologies

[Questions?](#) [Comments?](#)

Write us! - [mailto:...](#)



The Informant

Your personal search agent
on the internet

Results - Query 2

Query

Summary:

Query text: <i>building automation systems</i>		
Query logic	Search engine	Last search date
match some terms (OR)	Excite	Saturday, September 20, 1997

Summary

Do you need *an explanation of the results table?*

Edit account

- [Main page](#)
- [Questions?](#)
- [Contact us](#)
- [Autologin](#)
- [Change password](#)
- [Remove account](#)

Rank	Status	Site
1/10	new	VSE OLE Automation Development http://www.icon.co.za/~vse/Welcome.html
2/10	new	EH Online: Enhancing Your Lifestyle with Home Electron http://www.electronichouse.com/
3/10	new	Yahoo! - Business and Economy:Companies:Construction http://www.yahoo.com/text/Business_and_Economy/Com
4/10	new	VTT Technical Research Centre of Finland - Homepage http://www.vtt.fi/
5/10	new	BUILDING AUTOMATION SYSTEMS http://www.np.ac.sg/~tcc1/bas.htm
6/10	new	HIT Automation http://www.hit.co.kr/eauto.html
8/10	new	HVAC Concepts, Inc. http://www.nacgroup.com/hvac/
10/10	new	The World-Wide Web Virtual Library: Mechanical Engine http://cdr.stanford.edu/html/WWW-ME/home.html
7/10	old	VSTU -- Department of Automation and Robotization of I http://www.vorstu.ac.ru/vstu/f5/main-en.html
9/10	old	Document Automation Systems, LLC - building tour http://www.documentautomation.com/pictures.htm

Push Technologies is not responsible for the con

Thank you for using the *Informant*.

©1995-1997 Push Techn

[Questions?](#) [Comm](#)
[Write us!](#)



The Informant

Your personal search agent
on the internet

Results - Query 3

Query Summary:

Query text: <i>home automation</i>		
Query logic	Search engine	Last search date
match some terms (OR)	Excite	Saturday, September 20, 1997

- [Summary](#)
- [Edit account](#)

Do you need [an explanation of the results table?](#)

- [Main page](#)
- [Questions?](#)
- [Contact us](#)
- [Autologin](#)
- [Change password](#)
- [Remove account](#)

Rank	Status	Site
1/10	new	Institute of Hydraulics and Automation http://www.tut.fi/~ihatut/Index.html
2/10	new	PLC PLCs automation Transtronics transducers sensors Lightning protect... http://www.xtronics.com/index.htm
3/10	new	TUT Automation and Control Institute homepage http://ae.tut.fi/
4/10	new	Alfa Laval Automation Homepage http://www.automation.alfalaval.se/
5/10	new	Azure's AutomationNET Home Page http://www.automationnet.com/
6/10	new	PHD, Inc. Industrial Automation Home Page http://www.phdinc.com/
7/10	new	Precision Industrial Automation Inc. http://www.precisionia.com/
8/10	new	Allied Automation, Inc. - Main http://www.a1.com/allied/
9/10	new	Welcome to ZIP Mail Services http://www.pacific.net/~zipmail/
10/10	new	Sales Force Automation Web Site Menu http://www.pentech.net/pt1menu.shtml

Push Technologies is not responsible for the contents of third-party web sites.

Thank you for using the Informant. Please come again!

©1995-1997 Push Technologies

[Questions?](#) [Comments?](#)

Write us!



