

Sistema de Gestão de Teletrabalho para Ambiente Windows Suportado em RDIS

Helder Caixinha, Alexandre Escada, Fernando Ramos, José Santos

Resumo - A organização empresarial atravessa um período de enormes transformações. Um dos vértices visíveis dessas alterações é o desenvolvimento do teletrabalho como nova realidade para trabalhadores e empresas. Este artigo descreve a implementação de um conjunto de aplicações destinadas à gestão e utilização do referido teletrabalho, tendo como plataforma de comunicação a Rede Digital com Integração de Serviços. As aplicações implementadas foram desenvolvidas em linguagens Visual Basic e C++.

Abstract - The enterprise organization is passing through a period of enormous transformations. One of its aspects is the development of the telework as a new reality to workers and enterprises. This paper describes the implementation of a group of applications, to perform the management and use of the telework, using as a communication support the Integrated Services Digital Network. The applications were developed in Visual Basic and C++ languages.

I. INTRODUÇÃO*.

Com as sucessivas mudanças do mundo empresarial, as empresas enfrentam desafios decorrentes da globalização dos mercados, que levam à procura de novas soluções organizativas permitindo-lhes assim enfrentar a concorrência em termos mais seguros e rentáveis.

Muitas vezes são gastos períodos significativos de tempo nas deslocações dos trabalhadores para os respectivos locais de trabalho, desperdiçando-se assim importantes períodos de tempo numa actividade não produtiva e normalmente causadora de *stress*.

Para evitar este tipo de situações, surge hoje em dia, uma novo tipo de trabalhador, o qual desempenha as suas funções profissionais sem sair de casa. Obtêm-se assim para as empresas custos de produção mais baixos, aumentos na produtividade e nas oportunidades de emprego, entre outros benefícios. Para o trabalhador, esta nova realidade traz consigo uma melhoria das condições de vida e um aumento dos seus tempos de lazer. É de realçar o efeito catalisador no desenvolvimento de novos recursos na área das telecomunicações, causadas pela apetência do mercado por novos serviços, que ao se desenvolverem, permitem a criação de novas formas de teletrabalho, mais ricas e eficientes.

Com este trabalho realizou-se o desenvolvimento e a especificação de um conjunto de aplicações que realizam a gestão de um ambiente de teletrabalho. Isto é, aplicações que permitem a definição, o controlo e a execução de um conjunto de tarefas que um dado trabalhador pode efectuar remotamente.

No capítulo de distribuição de tarefas, cada trabalhador receberá uma ou mais para executar, sendo essas tarefas não mais do que conjuntos de ficheiros agrupados, que se traduzirão em toda a documentação necessária à execução das mesmas.

O sistema desenvolvido é constituído por duas aplicações complementares:

- **Sistema Central** de gestão, onde é permitido: a definição das tarefas, o agrupar de documentação, a distribuição de trabalho e toda a gestão envolvida na recepção das respostas enviadas pelos trabalhadores.

- **Sistema Terminal** de execução, onde é permitido: a recepção de tarefas, a sua execução de forma integrada e o envio das respectivas respostas para o sistema central.

Ambas as aplicações permitem a comunicação suportada por RDIS¹, de uma forma transparente para o utilizador, usando para tal, os variados serviços implementados com esse fim.

Este sistema foi desenvolvido para o ambiente Windows e no capítulo das comunicações é suportado por placas de comunicação RDIS do tipo PCBIT voz/dados, que dispõem de telefone integrado.

II. REDE DIGITAL COM INTEGRAÇÃO DE SERVIÇOS.

As crescentes necessidades de comunicações nos mais diversos sectores da sociedade moderna, veio trazer novas soluções, de entre as quais a chamada Rede Digital com Integração de Serviços [1]. Esta rede foi projectada como um padrão mundial, que serve uma larga variedade de serviços digitais.

A RDIS apresenta uma largura de banda bastante superior relativamente às linhas telefónicas convencionais (que permitiam o seu uso por sistemas computacionais, através dos vulgarmente chamados modem's²). Podem-se definir, segundo o ponto de vista do CCITT, os seguintes princípios para a RDIS: suporte de voz e aplicações

* Trabalho realizado no âmbito da disciplina de projecto.

¹ Rede Digital com Integração de Serviços.

² Modulador / Desmodulador.

diversas, utilizando um conjunto limitado de potencialidades; suporte de comunicações, quer por comutação a pacotes, quer por comutação a circuito; ligações a 64 Kbps; capacidade de suporte de serviços sofisticados, muito para além dos normais em ligações comutadas a circuitos; arquitectura de protocolos em camadas seguindo de muito perto o modelo OSI³; configurações físicas variáveis.

Foi utilizado o denominado *acesso básico* à referida rede, permitindo assim dispor de dois canais tipo B (64 Kbps) e um de tipo D (16 Kbps), o que se traduzirá numa resposta satisfatória às necessidades das aplicações no domínio da comunicação.

III. HARDWARE E SOFTWARE DE SUPORTE.

A. Placa PCBIT.

A placa PCBIT [2] permite o interface entre a RDIS e o sistema computacional utilizado. É constituída na sua essência por dois processadores, um responsável pelas ligações de voz e outro pelas ligações de dados.

A placa possui dois conectores externos, um para ligação de um telefone e outro para a ligação à rede de comunicações.

B. Application Program Interface - API.

Com a divulgação em larga escala dos computadores pessoais, a oferta de uma enorme variedade de *packages* de *software* tem sido uma constante. Com a sua diversidade, aparecem problemas de incompatibilidade entre aplicações de teleserviços. Para resolver parte dos problemas foram usadas diversas plataformas no desenvolvimento do *software* [3], caminhando-se de um nível inferior, para um nível superior que terminará com o interface gráfico da aplicação final (fig. 1).

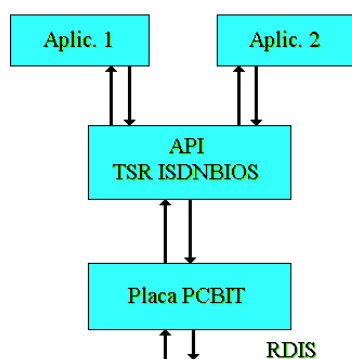


Fig. 1 - Plataformas de *software*.

A API, é um programa residente em memória (TSR⁴), que estabelece a interligação entre a placa RDIS e a aplicação final, através de interrupções por *hardware* (notificando a chegada de uma mensagem), pela sua memória e de interrupções geradas por *software*.

O mecanismo de funcionamento da API baseia-se na criação de filas de espera do tipo FIFO⁵, uma por cada aplicação existente no sistema computacional. No caso presente o número de filas de espera, está limitado a quatro, limitando a igual número a quantidade de aplicações que façam uso dos recursos proporcionados pela RDIS.

A API contém as primitivas essenciais para o estabelecimento, gestão e finalização de uma ligação:

- **API_Register(...)**, responsável pelo registo da API, criando e especificando *buffers* de memória, onde serão armazenadas as mensagens de sinalização e de dados, assim como os dados em si.

- **API_Release(...)**, responsável pela libertação da API, libertando os *buffers* criados aquando o registo da aplicação.

- **API_PutMessage(...)**, responsável pelo envio de mensagens da aplicação para a API.

- **API_GetMessage(...)**, responsável pela recepção de mensagens (enviadas pela API), por parte da aplicação.

IV. SOFTWARE DESENVOLVIDO EM C++.

A. Organização e explicação das classes utilizadas.

No desenvolvimento do *software*, foi sempre tido como objectivo escrever código estruturado e de fácil: uso, leitura, evolução e compreensão futura. Para tal foram construídos vários módulos. Todas as funções construídas são usadas pela aplicação final, na forma de DLL⁶.

Neste desenvolvimento foram aproveitadas as vantagens e qualidades da linguagem orientada por objectos C++ [4], linguagem essa que permite a definição de classes, sua derivação, criação de funções virtuais⁷ e virtuais puras⁸, para além de um maior encapsulamento de dados e procedimentos.

O uso de funções virtuais, permite-nos definir e usar em classes bases, procedimentos, que poderão ser redefinidos de modo adequado em classes derivadas. Esta característica revela-se fundamental na escrita de um código genérico e flexível. As características atrás referidas permitem assim a construção de um hierarquia de classes [5] (fig. 2)

⁴ *Terminate and Stay Resident*.

⁵ *First In First Out*.

⁶ *Dynamic - Link Libraries*.

⁷ Funções virtuais permitem, que classes derivadas, tenham diferentes versões em relação à classe base.

⁸ Função virtual onde não existe corpo na classe base, sendo obrigatoriamente redefinida nas classes derivadas.

³ Open Systems Interconnection.

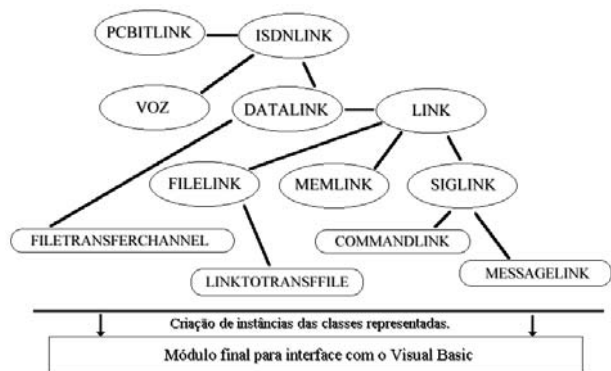


Fig. 2 - Hierarquia de classes.

O módulo de nível mais baixo na hierarquia (excluindo o TSR de interface com a placa PCBIT), PCBITLINK, é responsável pelo registo das aplicações, libertação das mesmas, distribuição das mensagens (vindas da API) através da implementação de uma máquina de estados e distribuição do tempo de CPU por meio do uso de um *timer*⁹, permitindo assim que diferentes aplicações tenham possibilidades de execução.

A classe PCBITLINK, usa referências da classe ISDNLINK, que sendo abstracta, contém essencialmente os métodos virtuais puros, que nas classes dela derivadas serão redefinidos.

A classe DATALINK, é responsável pelo estabelecimento e terminação, de ligações de dados sobre os canais B. É ainda responsável pela implementação dos protocolos das camadas dois e três do modelo OSI e pela identificação do número remoto quando na presença de uma ligação. Contém obrigatoriamente as redefinições dos métodos virtuais puros da classe de que deriva, ou seja ISDNLINK.

A classe DATALINK é ainda detentora de referências a objectos do tipo LINK, que se comportam como ligações virtuais, podendo existir vários sobre o mesmo canal. Sendo por isso possível a implementação de vários serviços de transferência de dados, ficheiros e mensagens em simultâneo.

As classes FILELINK, MEMLINK e SIGLINK, fornecem serviços concretos para a transferência de ficheiros e mensagens.

A classe FILELINK, é responsável pela transmissão e recepção de ficheiros, permitindo executar de diversos modos, tal procedimento. Esta classe contém diversas funções que são redefinidas, em classes derivadas.

A classe SIGLINK, permite o envio de pequenas mensagens, que correspondem simplesmente a *buffers* de memória. Torna-se indicada para o envio de mensagens de controlo, como por exemplo mensagens de erro.

A classe MEMLINK, que permite transferir *buffers* de memória de grande dimensão, funciona de modo semelhante à anterior.

As classes referidas até agora, não permitem o seu uso directo, pois na sua maioria são abstractas¹⁰, devendo ser criadas novas classes derivadas, onde as funções virtuais puras, serão definidas de acordo com as necessidades da aplicação.

A classe FILETRANSFERCHANNEL, representa o canal de comunicação físico, ou seja um canal B da RDIS. As suas funções e estruturas de dados permitem o estabelecimento, terminação e controlo das comunicações de dados. Não esquecendo a sua derivação da classe DATALINK, que inclui as funções virtuais e que agora são redefinidas.

A classe LINKTOTRANSFFILE, é responsável pelo envio e recepção de ficheiros, que mais uma vez, incluem essencialmente as redefinições das funções virtuais da classe de que deriva, FILELINK.

A classe COMMANDLINK, é responsável pelo envio de comandos (que poderão representar mensagens de erro, etc...), entre as aplicações intervenientes na ligação. Estas mensagens permitem o controlo do protocolo de alto nível implementado, o qual é necessário para o bom funcionamento das aplicações.

A classe MESSAGELINK, é muito semelhante à classe COMMANDLINK, permitindo o envio e recepção de mensagens, tendo sido criada para a implementação de um *talk on-line*¹¹.

B. Organização das DLL's, seu uso e funcionamento.

Todo o código desenvolvido em C++, foi compilado na forma final de DLL. No final obtivemos três DLL's:

- **ISDNBIOS.DLL**, que constitui a base da hierarquia, sendo fornecida com a placa PCBIT.
- **ISDN.DLL**, onde se encontra a definição das classes de nível intermédio, DATALINK, SIGLINK, MEMLINK, FILELINK e VOICELINK.
- **ISDNVB.DLL**, que inclui os procedimentos de mais alto nível, podendo estes ser usados directamente pelo interface gráfico. Fazem dela parte as classes FILETRANSFERCHANNEL, LINKTOTRANSFFILE, COMMANDLINK e MESSAGELINK.

No final obtivemos um conjunto de funções de fácil utilização em Visual Basic que passamos a descrever:

- **VBLoad(...)**, esta função deve ser chamada no arranque da aplicação, o argumento passado corresponde aos *handles* de comunicação entre a DLL e a aplicação (de C para Visual Basic).
- **VBUnload(...)**, função para o fim da aplicação.
- **LigacaoDePara(...)**, tenta uma ligação de um determinado ponto para outro.
- **Disconnect(...)**, tenta desligar uma ligação existente.

¹⁰ Uma classe abstracta, é uma classe contendo pelo menos uma função virtual pura.

¹¹ Sistema que permite o intercâmbio de mensagens, escritas através de teclado, de forma simultânea.

⁹ Temporizador.

•**Remoto(...)**, permite reconhecer o endereço remoto, numa ligação.

•**SubRemoto(...)**, permite reconhecer o sub-endereço remoto, numa ligação.

•**VBGetEstado(...)**, consulta o estado da aplicação (desligada, ligada, etc).

•**EnviaFile(...)**, envio de ficheiros, uma vez estabelecida a ligação.

•**GetMyPath(...)**, consulta o directório de depósito dos ficheiros recebidos.

•**SetMyPath(...)**, especificação do directório de depósito dos ficheiros recebidos.

•**GetRecvFileNam (...)**, consulta do nome especificado para o ficheiro recebido.

•**SetRecvFileName (...)**, especificação de um nome para o ficheiro recebido, diferente do seu original.

•**ResetRecvFileName(...)**, os ficheiros recebidos terão de novo o nome de origem.

•**SendMesgCommand(...)**, envio de uma mensagem tipo comando.

•**SendMesg(...)**, envio de uma mensagem tipo dados.

•**VBCSetCommunicationType(...)**, permite seleccionar o depósito de uma mensagem tipo comando.

•**EscreveTextoinHandle(...)**, escreve texto num dado *handle*, a partir de um *array* de *handles*.

• **EscreveTextoFromC(...)**, semelhante à anterior, mas o *handle* é perfeitamente especificado.

C. Comunicação DLL's - Visual Basic.

A passagem de parâmetros de Visual Basic para C, é bastante simples, não oferecendo problemas de maior. Mas já o mesmo não acontece na passagem de parâmetros em sentido contrário, principalmente quando pretendemos efectuar-la por referência. Outro problema na interacção referida, prende-se com a notificação da aplicação, o que deverá acontecer face à ocorrência de diversos acontecimentos (estabelecimento de uma chamada, início e finalização no envio de um ficheiro, recepção de uma mensagem, etc...).

Esta comunicação é feita através de janelas de texto (*text boxes*) definidas na aplicação, cujos *handles*¹², deverão ser passados à DLL através de uma estrutura¹³ de *handles*, que deverá ser incluída nos parâmetros da função *VBLoad()*. Deste modo, apenas é necessário interpretar a ocorrência de um *Text_Change*¹⁴ na janela de texto. É deste modo que se gerem os protocolos entre as aplicações; a indicação do progresso, quer na transmissão, quer na recepção de ficheiros; a troca de mensagens de

dados e é ainda deste modo que a aplicação recebe as mensagens de erro vindas das DLL's.

V. FUNCIONAMENTO DA APLICAÇÃO EM MULTITASK COOPERATIVO.

As aplicações funcionando no sistema operativo Windows, têm a possibilidade de existirem em simultâneo, realizando o sistema operativo, para tal, um ciclo de *spooling*, onde se verifica a fila de mensagens associada à aplicação em questão. Alguma mensagem aí existente será processada, devolvendo-se posteriormente o controlo ao sistema operativo, para que outra aplicação possa disfrutar de igual tratamento.

Deste modo uma aplicação como as desenvolvidas, onde a frequência de chegada de mensagens por parte da API, é muito mais lenta do que o seu processamento por parte do CPU; pode sem problemas, estar em *background*, afectando, apenas ligeiramente, o desempenho do sistema.

Esta característica é bastante importante neste tipo de aplicações, pois na maioria do tempo, o teletrabalhador, encontra-se a realizar as suas tarefas, em aplicações diversas, e raramente estará a usar directamente a aplicação de teletrabalho.

Em todas as situações o teletrabalhador, estará em potencial contacto com todos os postos associados, nomeadamente a central; sendo notificado da presença de ligações RDIS sempre que uma ocorra.

VI. INTERFACE GRÁFICO.

A. Especificação da funcionalidade.

Foi implementada a seguinte funcionalidade para o sistema central:

- Construção de tarefas a enviar.
- Gestão de uma base de dados contendo informação relevante dos trabalhadores.
- Envio de tarefas aos trabalhadores.
- Análise das respostas, às tarefas enviadas.
- Apoio à decisão, no capítulo do envio de tarefas.
- Manutenção do historial do trabalhador.
- Gestão de todas as comunicações, de uma forma transparente ao utilizador.
- Associação entre ficheiros de um determinado tipo e as aplicações que os originam, de forma a conseguir-se um ambiente integrado de trabalho.
- Criação de serviços complementares ao nível da comunicação.

Do mesmo modo para o sistema terminal:

- Construção das respostas a enviar.
- Envio de respostas para o sistema central.

¹² Valor inteiro único utilizado para a identificação e acesso a uma janela, controlo, etc...

¹³ Esta estrutura deve ser construída de modo semelhante em C++ e Visual Basic.

¹⁴ Função do Visual Basic que é chamada sempre que existe alteração do conteúdo do texto numa *text box*.

- Gestão de todas as comunicações, de uma forma transparente ao utilizador.

- Associação entre ficheiros de um determinado tipo e as aplicações que os originam, de forma a conseguir-se um ambiente integrado de trabalho.

- Criação de serviços complementares ao nível da comunicação.

B. Perfil do utilizador.

O desenho de um interface deve ter como guia no seu desenvolvimento um dado tipo de utilizador, o qual foi caracterizado por: psicologia e atitude perante o sistema, conhecimentos e experiência e características do trabalho a ser realizado.

C. Documentação.

Qualquer aplicação deve conter um conjunto de documentos auxiliares que forneçam um correcto apoio à sua aprendizagem e utilização.

Neste projecto dotámos as aplicações de um *help on-line*¹⁵, bem como de um *context sensitive hint system*¹⁶. Esta documentação atrás referida é ainda complementada com os habituais manuais do utilizador.

D. Estilos de Diálogo.

De um forma genérica todo o interface deve respeitar os seguintes princípios [6]:

- Compatibilidade com o utilizador.
- Compatibilidade com o trabalho a desenvolver.
- Simplicidade.
- Familiaridade.
- Minimização dos erros de interacção.
- Fácil recuperação dos referidos erros.

Com base nos princípios apresentados e no perfil do utilizador, os estilos de diálogo escolhidos foram os mais adequados e os que melhor servem a funcionalidade pretendida.

Foram implementadas nos interfaces, várias características, tendo em vista a obtenção de um ambiente amigável para o utilizador, que passamos a enumerar:

i) Organização e disposição da interface.

- Suporte adequado do trabalho a desenvolver.
- Informação organizada por grupos relacionados.
- Divisão lógica da informação, mantendo itens relacionados e interdependentes na mesma *form*¹⁷.

Construção cuidada do aspecto visual das *forms*, de modo a aumentar a clareza da interface.

ii) *Captions*¹⁸, campos e entradas de dados.

- Maximização da legibilidade de cada *form*.
- Atribuição a cada grupo de informação de um título claro para o utilizador.
- Distinção clara entre *captions* e campos.
- Indicação, quando necessária da formatação a que o utilizador tem que obedecer na introdução dos dados.
- Regras simples para a entrada de dados.
- Fornecimento de *defaults*¹⁹, sempre que tal seja conveniente.

iii) Instruções e ajudas ao utilizador.

- Fornecimento de sistemas de ajuda ao utilizador, que facilitem a sua interacção.
- Fácil navegação, dentro de uma dada *form*.
- Consistência²⁰ ao longo de todo o interface.

iv) Navegação.

- Posicionamento correcto do cursor no início de cada *form*.
- Total liberdade de movimentos dentro de cada campo, entre campos e mesmo entre diferentes *forms*.

v) Erros.

- Fornecimento de mensagens de erro, com uma informação precisa e clara sobre a natureza dos erros ocorridos e a forma de os solucionar.

E. Descrição do interface da aplicação central.

O utilizador ao inicializar a aplicação, encontrará após a fase de apresentação, a janela de acesso à aplicação. Nesta terá de introduzir o seu *login* e *password* de forma a poder prosseguir.

A informação respeitante a esta janela, encontra-se armazenada de uma forma codificada, no ficheiro PASSWORD.INI.

O utilizador dispõe de três tentativas para uma introdução correcta, ao fim das quais a aplicação é compulsivamente terminada.

Realizada a fase de acesso, entramos na aplicação propriamente dita. É assim apresentada ao utilizador a janela de gestão do sistema central (fig. 3).

¹⁵ Sistema integrado na aplicação de ajuda ao utilizador na sua interacção.

¹⁶ Sistema integrado na aplicação que fornece uma informação contextual à interação que o utilizador realiza.

¹⁷ Janela.

¹⁸ Títulos dos campos onde o utilizador realiza a introdução ou visualização de dados.

¹⁹ Valores por defeito.

²⁰ Manutenção de um conjunto de aspectos, quer visuais, quer estruturais, ao longo de todo o interface.



Fig. 3 - Janela de gestão.

Aqui o utilizador tem acesso aos serviços fornecidos, os quais passamos a explicar:

- **Build Task**, permite a construção de tarefas.
- **Worker DB**, acesso à base de dados dos trabalhadores.
- **Worker Link**, acesso aos serviços de comunicação implementados.
- **Options**, configuração da aplicação.
- **Send Task**, permite o envio de tarefas, previamente contruídas.
- **Worker Stat...**, acesso à informação que visa apoiar a decisão de envio de tarefas e acesso ao capítulo de classificação e avaliação das respostas enviadas pelos trabalhadores.
- **Responses**, permite a consulta e análise das respostas enviadas, pelos trabalhadores.
- **Exit**, saída da aplicação.

Cada tarefa a enviar a um dado trabalhador é no fundo um conjunto de ficheiros, que serão agrupados e enviados em bloco (fig. 4). Resulta assim a seguinte estrutura para a tarefa:

- ****.TSK, ficheiro contendo o nome dos diversos ficheiros agrupados à tarefa, bem como um pequeno comentário descritivo de cada um deles.
- ****.PTH, ficheiro contendo a *path* completa de cada ficheiro agrupado. Este ficheiro é utilizado pelo sistema na fase de envio das tarefas.
- ****.DAT, ficheiro de texto que opcionalmente poderá ser criado pelo utilizador para fornecer uma descrição mais detalhada da tarefa em questão.

Os ficheiros ****.TSK, ****.PTH, ****.DAT, são armazenados no directório TASKS (sub-directório da aplicação), de modo a obter-se em permanência um arquivo de todas as tarefas já construídas.

Cada ficheiro agrupado encontra-se associado de uma forma integrada a uma dada aplicação. O utilizador inicialmente apenas deve construir o seu conjunto de associações que serão armazenadas no ficheiro ASSOCIACIAC.ASC. Ao seleccionar um ficheiro para visualização o sistema recorre às associações existentes para determinar a aplicação a inicializar, caso ainda não exista essa associação o utilizador será convidado a efectuar-las.

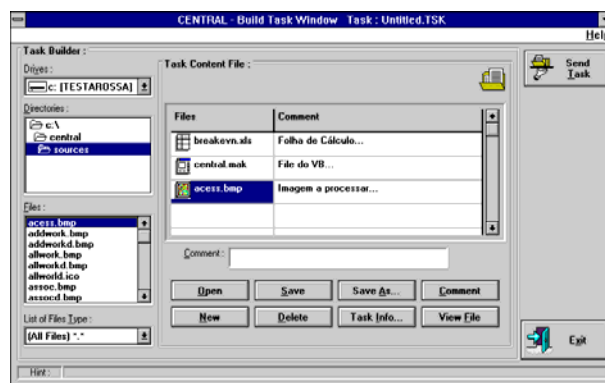


Fig. 4 - Janela de construção de tarefas.

Para esta aplicação foi construída uma base de dados, WORKER.MDB, utilizando o *MS Access*, onde são armazenados os dados respeitantes a cada trabalhador. A sua gestão é efectuada numa janela própria (fig. 5).

Nesta base de dados são realizadas, buscas, utilizando a linguagem SQL²¹, o que permite a visualização dos dados correspondentes às intenções do utilizador (fig. 6).

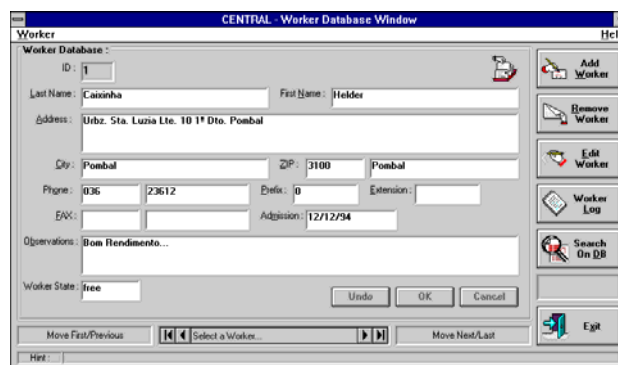


Fig. 5 - Janela de gestão da base de dados.

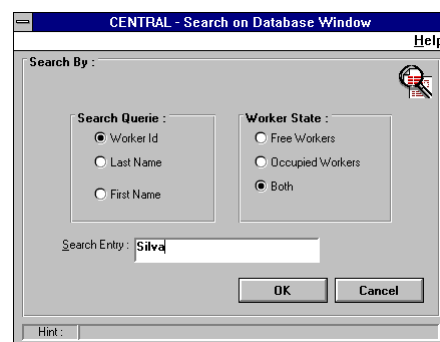


Fig. 6 - Janela de busca na base de dados.

O utilizador da aplicação central dispõe de um mecanismo automático de envio de tarefas para um dado trabalhador, bastando-lhe para tal seleccionar a tarefa e o seu receptor (fig. 7).

O envio pressupõe as seguintes fases:

- Actualização do historial, do receptor seleccionado com os dados referentes à tarefa enviada, isto é, nome da tarefa, data de envio e prazo de entrega.

²¹ Search Query Language.

- Abertura do ficheiro ****.PTH, contendo a *path* completa dos ficheiros a enviar.
- Inicialização da janela de gestão das comunicações.
- Estabelecimento da ligação.
- Transferência dos ficheiros indicados por ****.PTH.
- Transferência dos ficheiros ****.TSK e ****.DAT (este último, caso exista).
- Terminação da ligação.

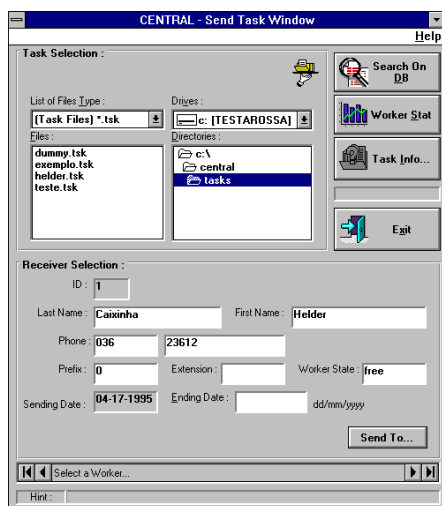


Fig. 7 - Janela de envio de tarefas.

No sistema terminal (caso não exista), é criado um directório, com o nome da tarefa, dentro do sub-directório TASKS, destinado a albergar todos os ficheiros transmitidos.

Uma resposta a uma dada tarefa recebida no sistema central é constituída pelos seguintes ficheiros:

- ****.RSP, ficheiro contendo o nome dos diversos ficheiros agrupados à resposta, bem como um pequeno comentário descritivo destes últimos.
- ****.DAT, ficheiro de texto que opcionalmente poderá ter sido enviado pelo trabalhador, para fornecer uma descrição mais detalhada da resposta em questão.
- Diversos ficheiros agrupados.

A recepção na aplicação central de uma resposta implica:

- Estabelecimento da ligação.
- Inicialização da janela de gestão das comunicações.
- Preenchimento de uma ficha de recepção da tarefa com os dados respeitantes ao seu remetente (número identificativo), data de chegada e nome da resposta.
- Actualização do ficheiro que contém o historial do trabalhador (remetente da resposta), nomeadamente a introdução da data em que foi recebida a resposta.
- Criação de um directório, com o nome da resposta, dentro do sub-directório RESPONSE, destinado a albergar todos os ficheiros de uma resposta.
- Transferência de todos os ficheiros agrupados.
- Transferência dos ficheiros ****.RSP e ****.DAT (este último, caso exista).
- Terminação da ligação.

- Introdução da resposta na lista das últimas recebidas.

As respostas podem ser analisadas e classificadas, bastando para tal a selecção da pretendida de entre as que foram recebidas (fig. 8).

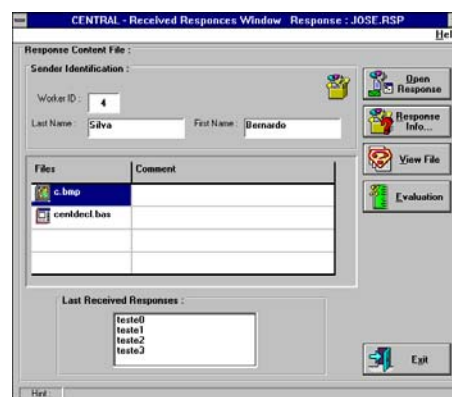


Fig. 8 - Janela de recepção de respostas.

A escolha do receptor de uma dada tarefa, é uma decisão importante e por vezes difícil, para tal foi concebido um sistema de apoio, que com base num critério, ainda que simples, pode ajudar tal decisão.

O trabalhador quando é integrado na empresa, recebe um índice de desempenho de 0%, o qual será actualizado aquando da classificação das respostas por ele enviadas. Ao classificar uma resposta, atribuímos um dado valor em percentagem, que reflecte o grau de satisfação pelo seu desempenho. Esse valor atribuído entra em conta para o cálculo de um índice médio a que chamamos *índice de desempenho do trabalhador*. Com o conjunto dos índices de todos os trabalhadores é calculado um valor que designaremos por *desempenho médio da empresa*. Os trabalhadores encontrar-se-ão assim classificados numa escala de desempenho.

Os trabalhadores são então comparados quer isoladamente, com o desempenho médio da empresa (fig. 9), quer de uma forma global entre eles (fig. 10).

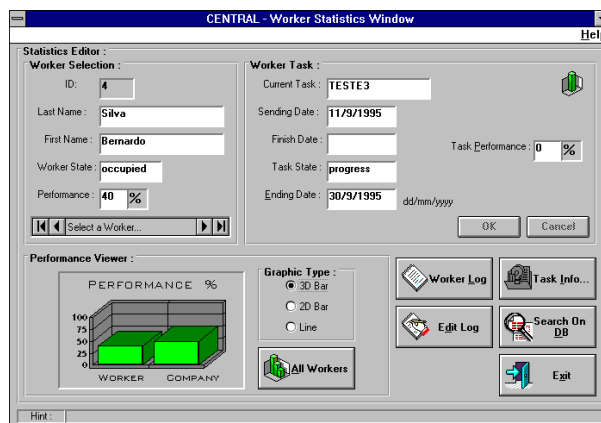


Fig. 9 - Janela dos dados estatísticos do trabalhador.

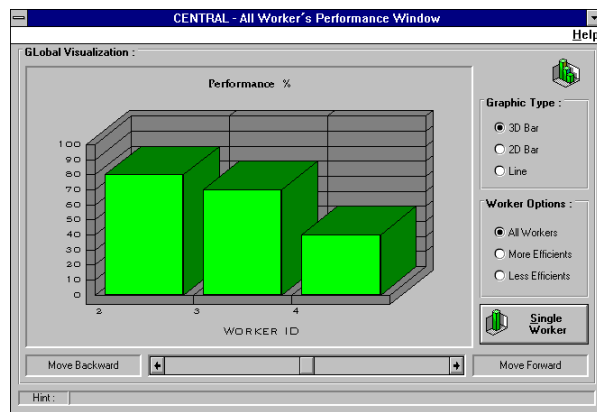


Fig. 10 - Janela de visualização global do desempenho.

Para a classificação de respostas ou alterações nos dados de tarefas enviadas faz-se uso da janela representada na figura 9.

Na aplicação central existe um ficheiro para cada trabalhador, que contém todo o seu historial (fig. 11), ou seja todos os dados referentes às tarefas que lhe foram atribuídas e às respostas que ele produziu, nomeadamente:

- Nome da tarefa enviada, que implica à partida uma resposta com igual nome, de modo a facilitar a identificação lógica de um *par tarefa/resposta*.
- Data de envio.
- Data limite de entrega da resposta.
- Estado da tarefa (concluída ou em execução).
- Data de recepção da resposta à tarefa.
- Índice de desempenho.

Este ficheiro é actualizado sempre que existe um envio de uma tarefa, ou a recepção de uma resposta, reflectindo deste modo os seus últimos campos, o estado actual do trabalhador. São todos estes dados que o utilizador tem à sua disposição sempre que efectua uma classificação de uma resposta ou uma decisão de envio de tarefa.

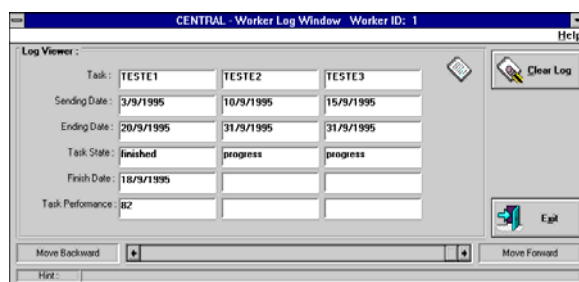


Fig. 11 - Janela de visualização do historial do trabalhador.

Existem quatro serviços que fazem uso da janela de gestão de comunicações (fig. 12): transferência de tarefas, transferências de respostas, transferência de um ficheiro isolado e troca de correio. Sempre que um destes serviços é utilizado, o sistema provoca o aparecimento desta janela em ambas as aplicações intervenientes no processo de comunicação. Consegue-se assim uma informação dos

progressos de cada um dos serviços referidos; para tal existe aí:

- Informação do serviço prestado.
- Estado actual do serviço utilizado, isto é: estado da ligação, destinatário da ligação, remetente da ligação, etc...
- Barra de progresso na transferência de uma tarefa ou resposta, indicando a percentagem de ficheiros agrupados, já transmitidos.
- Barra de progresso do ficheiro corrente, indicando a percentagem já transferida.

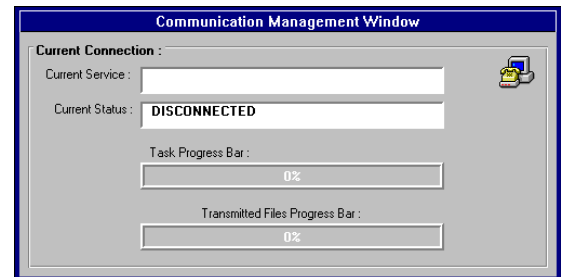


Fig. 12 - Janela de gestão de comunicações.

De notar que o sistema, automaticamente, assume o papel de emissor ou receptor consoante se trate de um envio, ou de uma recepção, adequando assim as informações prestadas ao utilizador, em termos de quantidades enviadas ou recebidas.

Cada um dos serviços referidos recorre a protocolos, que transformam as aplicações em máquinas de estado, onde se assumem determinadas sequências de funcionamento. Cada um destes protocolos, tem por motivos de segurança uma *password* interna que é trocada entre as aplicações intervenientes de modo à correcta execução de um dado serviço.

Nos referidos protocolos foram implementados alguns mecanismos, de recuperações de erros, de entre os quais:

- Auto-suspensão das ligações em caso de erros no seu decurso.
- Mecanismo de auto-suspensão da tentativa de ligação em caso de dificuldades.
- Repetição do envio de um ficheiro mal transmitindo. O sistema repete, um determinado número de vezes, o envio do ficheiro.
- Notificação aos utilizadores da impossibilidade de transferir um determinado ficheiro. Exibem-se janelas de aviso, onde se alerta para a falha existente, tal situação é também gravada em ficheiros próprios que indicam a ocorrência da situação, prevenindo-se assim o caso dos utilizadores não se encontrarem no posto de trabalho, nesse momento.

No respeitante à comunicação, a juntar aos serviços de envio e recepção de tarefas ou respostas, o utilizador tem adicionalmente ao seu dispor a possibilidade de seleccionar um de três serviços (fig. 13):

- Envio de um ficheiro isolado. Neste serviço o utilizador escolhe um ficheiro e um destinatário, quer através da

selecção de um trabalhador existente na base de dados, quer através da marcação do número de telefone desejado. Consideram-se assim dois tipos de ligações, a interna e a externa, respectivamente.

•Envio de correio. O utilizador constrói uma mensagem, a partir de três campos: o remetente, o assunto e a mensagem propriamente dita; esta é automaticamente gravada no sub-directório MAILBOX e é em seguida enviada para a máquina destinatária. Aí, é-lhe atribuído um nome único e sequencial de modo a inseri-la na lista de mensagens do receptor. O receptor tem assim uma noção correcta da ordem de chegada, bem como uma acrescida segurança, no respeitante ao envio accidental de mensagens com nomes iguais.

•Conversa interactiva (*talk on-line*). Aqui o utilizador após selecção do interlocutor, tenta o estabelecimento da conversação. O que conduz ao surgimento de uma janela semelhante na aplicação do interlocutor. Aí este último tomará conhecimento de quem o procura contactar, tendo então a liberdade de aceitar ou recusar tal contacto, sem que com isto o utilizador se aperceba. Caso não seja tomada nenhuma acção a ligação será rejeitada automaticamente, ao fim de algum tempo.

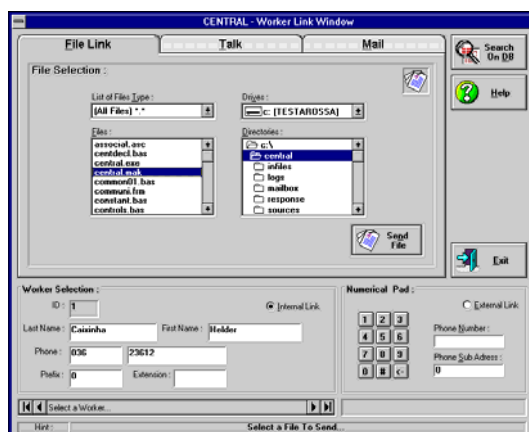


Fig. 13 - Janela de comunicação.

F. Descrição da interface da aplicação terminal.

A aplicação terminal, é no fundo, uma aplicação semelhante à central, apenas desprovida de algumas capacidades desta última. Como tal serão feitas referências apenas às diferenças significativas.

Após um processo de inicialização e acesso à aplicação o utilizador, será confrontado com a janela de gestão da aplicação (fig. 14).



Fig. 14 - Janela de gestão.

Aqui o utilizador tem acesso a todos os serviços fornecidos, os quais passamos a explicar:

- Build Response**, permite a construção de respostas.
- Worker Link**, acesso aos serviços de comunicação implementados.
- Options**, configuração da aplicação.
- Send Response**, permite o envio de respostas, previamente contruídas, para a central.
- Tasks**, permite ao utilizador consultar as tarefas enviadas pela central.
- Exit**, saída da aplicação.

Cada resposta a enviar para a central é no fundo um conjunto de ficheiros, que serão agrupados e enviados em bloco. Resulta assim a seguinte estrutura:

- ****.RSP, ficheiro contendo o nome dos diversos ficheiros agrupados à resposta, bem como um pequeno comentário descritivo destes últimos.
- ****.PTH, ficheiro contendo a *path* completa de cada ficheiro agrupado. Este ficheiro é utilizado pelo sistema na fase de envio das respostas.
- ****.DAT, ficheiro de texto que opcionalmente poderá ser criado pelo utilizador para fornecer uma descrição mais detalhada da resposta realizada.

Os ficheiros ****.RSP, ****.PTH, ****.DAT, são armazenados no directório RESPONSE (sub-directório da aplicação).

O utilizador do sistema terminal, dispõe de um mecanismo automático de envio de respostas para a central, bastando-lhe para tal seleccionar a resposta a enviar (fig. 15).

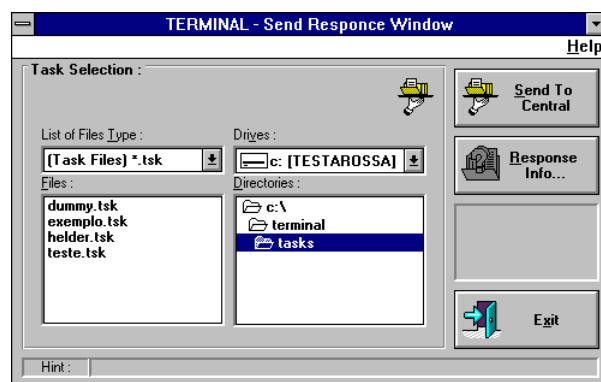


Fig. 15 - Janela de envio de respostas.

Aqui o destinatário das respostas é obviamente a aplicação central.

O envio de uma resposta pressupõe as seguintes fases:

- Abertura do ficheiro ****.PTH, contendo a *path* completa dos ficheiros a enviar.
- Inicialização da janela de gestão das comunicações.
- Estabelecimento da ligação.
- Envio do número de identificação do trabalhador, para efeitos de construção de uma ficha de recepção da resposta, na aplicação central.

- Transferência dos ficheiros indicados por ****.PTH.
- Transferência dos ficheiros ****.RSP e ****.DAT (este último, caso exista).
- Terminação da ligação.

Na aplicação central, caso não exista, será criado um directório, com o nome da resposta, dentro do sub-directório RESPONSE, destinado a albergar todos os ficheiros transmitidos.

A recepção na aplicação terminal de uma tarefa implica:

- Estabelecimento da ligação.
- Inicialização da janela de gestão das comunicações.
- Criação de um directório, com o nome da tarefa, dentro do sub-directório TASKS, destinado a albergar todos os ficheiros de uma tarefa.
- Transferência de todos os ficheiros associados.
- Transferência dos ficheiros ****.TSK e ****.DAT (este último, caso exista).
- Terminação da ligação.

Os ficheiros de uma tarefa podem então ser manipulados através da janela específica para esse efeito (fig. 16).

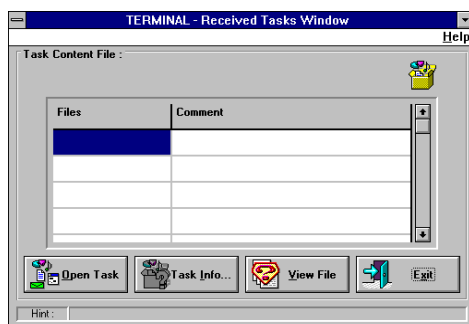


Fig . 16 - Janela de recepção de tarefas.

REFERÊNCIAS.

- [1] William Stallings Ph. D., "Data and Computer Communications", Fourth Edition, MacMillan.
- [2] Telecom Portugal, "Common - ISDN - API, Perfil Português, (1991-1993)", ver. 2.1.
- [3] Projecto PCBIT, "Desenvolvimento de Aplicações Windows para PCBIT", INESC 1993.
- [4] Bryan Flaming, "Turbo C++, A Self-Teaching Guide", Jonh Wiley & Sons Inc., 1991.
- [5] Osvaldo A. Santos, Fernando M. S. Ramos, "ISDNLINK - Uma Biblioteca de Classes para RDIS", Revista do DETUA
- [6] Deborah J. Mayhew, "Principles and Guidelines in Software User Interface Design", Prentice Hall, 1992.

VII. CONCLUSÕES.

Os objectivos propostos com este projecto foram atingidos, embora se pudesse ter avançado mais, principalmente no capítulo da integração de comunicações com voz. Ainda assim foram implementados serviços de comunicação flexíveis e abrangentes das necessidades dos utilizadores típicos destes sistemas.

Sentiram-se algumas dificuldades iniciais, ao nível do *software* de comunicação, o que adicionado a outros factores vieram condicionar enormemente o progresso deste projecto.

Apresentam-se assim um conjunto de aplicações, que foram um bom ponto de partida para as versões actualmente em desenvolvimento com vista à sua futura utilização por determinadas empresas.

Any comments to enhance this format can be send to jlo@inesca.pt.

This document was written in Word for Windows 2.0.

94.6.13 JLO

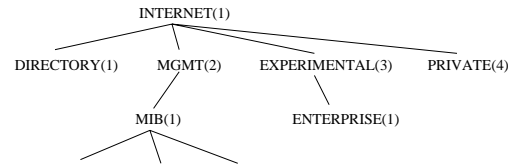


Fig. 1 - This is Style “legenda”.

A. This sub-title is of Style Heading 2

To construct a item enumeration use the format “pontos”.

- bla-bla-bla;

To insert program code or similar text just use the format “código” :

```

OBJECT-TYPE MACRO ::=
BEGIN
    TYPE NOTATION ::=
        ReferPart
        IndexPart
        DefValPart
    VALUE NOTATION ::= value (VALUE ObjectName)

```

When you want to create a figure, you must align it to the center, insert a Frame in the selection figure+legend and select Format-Frame (top or bottom) in order to avoid figures in the middle of the page. Finally remove the Border.

REFERENCES

Use the Style “referencia” to the references. Example:

- [1] M. T. Rose and K. McCloghrie, "Structure and Identification of Management Information for TCP/IP-based Internets", RFC 1155, DDN Network Information Center, SRI International, May 1990.

This file is available by anonymous FTP in:
greco.inesca.pt: ~/pub/docs/dos/revdetua.doc

```

$ftp greco.inesca.pt
user: anonymous
passwd: <none>
cd pub/docs/dos
bin
get revdetua.doc
bye
$

```