

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE CIÊNCIAS DA COMPUTAÇÃO
(Bacharelado)

**PROTÓTIPO DE UM SOFTWARE EDITOR ORIENTADO POR
GABARITOS PARA PROGRAMAS DATAFLEX**

TRABALHO DE ESTÁGIO SUPERVISIONADO SUBMETIDO À UNIVERSIDADE
REGIONAL DE BLUMENAU PARA A OBTENÇÃO DOS CRÉDITOS NA
DISCIPLINA COM NOME EQUIVALENTE NO CURSO DE CIÊNCIAS DA
COMPUTAÇÃO — BACHARELADO

FRANK JUERGEN KNAESEL

BLUMENAU, NOVEMBRO/1999

1999/2-17

PROTÓTIPO DE UM SOFTWARE EDITOR ORIENTADO POR GABARITOS PARA PROGRAMAS DATAFLEX

FRANK JUERGEN KNAESEL

ESTE TRABALHO DE ESTÁGIO SUPERVISIONADO, FOI JULGADO ADEQUADO
PARA OBTENÇÃO DOS CRÉDITOS NA DISCIPLINA DE ESTÁGIO
SUPERVISIONADO OBRIGATÓRIA PARA OBTENÇÃO DO TÍTULO DE:

BACHAREL EM CIÊNCIAS DA COMPUTAÇÃO

Prof. Maurício Capobianco Lopes — Supervisor na FURB

Marcílio Luiz Legal — Orientador na Empresa

Prof. José Roque Voltolini da Silva — Coordenador na
FURB do Estágio Supervisionado

BANCA EXAMINADORA

Prof. Maurício Capobianco Lopes

Prof. Everaldo Artur Grahl

Prof. Wilson Pedro Carli

SUMÁRIO

1	INTRODUÇÃO	10
1.1	OBJETIVOS	11
1.2	DESCRIÇÃO DA EMPRESA	12
1.3	ORGANIZAÇÃO DO TEXTO	13
2	EDIÇÃO ORIENTADA POR GABARITOS	15
2.1	EDITPLUS+ 1.25	15
2.2	ARANEAE 2.78	21
2.3	TEXTPAD 4	22
2.4	FTP EDIT 2.21	25
3	ANÁLISE ESSENCIAL	28
3.1	MODELO ESSENCIAL	28
3.1.1	<i>MODELO AMBIENTAL</i>	29
3.1.2	<i>MODELO COMPORTAMENTAL</i>	30
3.1.3	<i>DICIONÁRIO DE DADOS (DD)</i>	33
3.1.4	<i>MODELO ENTIDADE RELACIONAMENTO (MER)</i>	33
3.2	MODELO DE IMPLEMENTAÇÃO	33
4	DATAFLEX	35
4.1	PROGRAMA-FONTE	35
4.2	UTILITÁRIOS	38
5	ESPECIFICAÇÃO DO PROTÓTIPO	41
5.1	LISTA DE EVENTOS	42
5.2	DESCRIÇÃO DOS EVENTOS	42
5.3	DIAGRAMA DE CONTEXTO	44
5.4	DIAGRAMA DE FLUXO DE DADOS PARTICIONADO	45
5.4.1	<i>DESENVOLVEDOR INFORMA SINTAXE</i>	45
5.4.2	<i>DESENVOLVEDOR INFORMA GABARITOS</i>	45
5.4.3	<i>USUÁRIO EDITA PROGRAMA</i>	46

5.4.4	<i>USUÁRIO ABRE PROGRAMA</i>	47
5.4.5	<i>SERVIDOR TRANSFERE PROGRAMA</i>	48
5.4.6	<i>SISTEMA INFORMA ARQUIVOS ABERTOS</i>	49
5.4.7	<i>SISTEMA SALVA PROGRAMA</i>	49
5.4.8	<i>SISTEMA TRANSFERE PROGRAMA</i>	50
5.4.9	<i>USUÁRIO CONFIGURA SISTEMA</i>	50
5.4.10	<i>SISTEMA INFORMA DEFINIÇÕES DE ARQUIVOS</i>	51
5.4.11	<i>DFD - SISTEMA COMPLETO</i>	52
5.5	DICIONÁRIO DE DADOS (DD)	53
5.5.1	<i>FLUXOS DE DADOS</i>	53
5.5.2	<i>DEPÓSITOS DE DADOS</i>	54
6	IMPLEMENTAÇÃO	55
6.1	ARQUIVO DE DEFINIÇÃO DE SINTAXE	61
6.2	ARQUIVO DE GABARITOS	63
7	CONCLUSÃO	65
7.1	SUGESTÕES	66
	REFERÊNCIAS BIBLIOGRÁFICAS	67

LISTA DE FIGURAS

1: Ligação dos terminais e microcomputadores ao servidor na Prefeitura	13
2: Tela Inicial do EditPlus+	16
3: Menu do botão “New” do Editplus+.	16
4: “Template” do novo arquivo HTML.	17
5: Caixa Preferences do EditPlus+.	18
6: Antes e Depois da inserção do gabarito	19
7: Janela Principal do Editor Araneae	21
8: Tela de Configuração de Cores do Araneae	22
9: Tela inicial do TextPad 4.	23
10: Janela de Configurações do TextPad 4	25
11: Janela do FTP Edit.	26
12: Tela de Abertura de Arquivos do FTP Edit	27
13: Ferramentas do FTP Edit para HTML	27
14: Elementos Gráficos de um DFD.	31
15: Exemplo de um DC.	31
16: DFD do Evento Cliente Solicita Cadastramento	32
17: DFD do Evento Cliente Efetua Pagamento	32
18: DFD do Evento É hora de emitir relatório gerencial	32
19: Diagrama Completo de Fluxo de Dados do sistema Exemplo	33
20: Tela do utilitário DFQUERY	38
21: Diagrama de Contexto do Sistema de Edição de Arquivos Dataflex	44
22: Detalhamento de Processo - Desenvolvedor Informa Sintaxe	45

23: Detalhamento de Processo - Desenvolvedor Informa Gabaritos	45
24: Detalhamento de Processo - Usuário Edita Programa	46
25: Detalhamento de Processo - Usuário Abre Programa	47
26: Detalhamento de Processo - Servidor Transfere Programa	48
27: Detalhamento de Processo - Sistema Informa Arquivos Abertos	49
28: Detalhamento de Processo - Sistema Salva Programa	49
29: Detalhamento de Processo - Sistema Transfere Programa	50
30: Detalhamento de Processo - Usuário ConSistema	50
31: Detalhamento de Processo - Sistema Informa Definições de Arquivos	51
32: Diagrama de Fluxo de Dados completo do Sistema	52
33: Tela Principal do Protótipo do Editor	55
34: Barra de Ferramentas	56
35: Configurações de Edição / Exibição	58
36: Configuração do Servidor FTP	59
37: Configuração de Arquivos	59
38: Configuração de Cores	60
39: Configuração do Dataflex	60

LISTA DE QUADROS

1: Arquivo de “Auto-Completion” do EditPlus+ para C, C++ e Java. _____	10
2: Arquivo de Definição de Sintaxe do EditPlus+ para linguagem C e C++ _____	11
3: Exemplo de uma Lista de Eventos _____	20
4: Programa fonte exemplo do Dataflex. _____	26
5: Exemplo de uma imagem num programa fonte Dataflex _____	27
6: Exemplo de um Arquivo “.FD” _____	29
7: Exemplo de um Arquivo “.DEF” _____	30
8: Comandos da Linguagem Dataflex _____	61
9: Operadores da Linguagem Dataflex _____	62
10: Variáveis e Indicadores Pré-Definidos da Linguagem Dataflex _____	62
11: Formato do Arquivo de Definição de Sintaxe _____	62
12: Exemplo de Arquivo de Gabarito _____	64

RESUMO

Este trabalho ilustra por meio de um protótipo a edição de programas fonte da linguagem Dataflex com inserção de gabaritos, realce colorido para a sintaxe e transferência de arquivos via FTP. Atualmente, não foi encontrado nenhum editor para programas fonte da linguagem Dataflex com as características supra citadas. O objetivo principal deste trabalho foi desenvolver tal software. Para a especificação do protótipo foi utilizada a Análise Essencial, auxiliada pela ferramenta CASE Power Designer e para a implementação foi usado o ambiente de programação Delphi versão 4. O protótipo deste software foi colocado em uso prático na Prefeitura Municipal de Jaraguá do Sul, que utiliza a linguagem Dataflex e o sistema operacional Windows. Os primeiros resultados de sua utilização foram bons, porém, existem melhorias a serem realizadas.

ABSTRACT

This work illustrates by means of a prototype, the edition of programs source of the Dataflex language with insertion of templates, syntax color highlighting and transference of archives by FTP. Currently, there wasn't any software for edition of programs source of the Dataflex language with the features above. The main objective of this work was to develop such software. For the specification of the prototype, the Essential Analysis was used assisted by Power Designer CASE tool, and for the implementation, the Delphi programming environment version 4 was used. The prototype of this software was placed in practical use in the Jaraguá do Sul City Hall, that uses the Dataflex language and the operational system Windows. The firsts results of the utilization of this software were good, however, there are improvements to be carried through.

1 INTRODUÇÃO

A linguagem Dataflex existe desde 1981 e está sendo usada na Prefeitura Municipal de Jaraguá do Sul desde 1989.

Atualmente, os programas fonte do Dataflex são editados usando-se um Software chamado N`Word, um editor de textos simples (semelhante ao antigo WordStar para MS-DOS), que é executado através de emulação de terminal. Existe ainda, um editor de programas fonte Dataflex escrito pela DataAccess (fabricante do Dataflex), chamado DFPDS, que também é executado no servidor através de emulação de terminal.

A edição dos programas fica bastante complicada (no caso acima) para um usuário que não conhece nenhum dos editores supra citados. No caso do N`Word, existem muitas combinações de teclas, tornando-o complexo. Este editor que está sendo usado atualmente, é um editor de textos muito simples e sem recursos que em nada ajudam o programador a escrever seus programas. Através de menus, cores, botões e teclas de atalho simples e auxiliado por gabaritos, a edição dos programas fonte ficaria muito mais fácil.

O trabalho proposto terá enfoque na edição de programas fonte da linguagem Dataflex orientada por gabaritos(sintaxe) utilizando-se dos recursos visuais do Sistema Operacional Windows e transferência de arquivos via FTP.

Sendo assim, este estágio propõe a implementação de um protótipo de um software para ser executado sob o Sistema Operacional Windows, usando o ambiente de programação Delphi, aproveitando principalmente os recursos visuais e algumas facilidades deste ambiente, a fim de deixar a tarefa de edição destes programas menos árdua e mais produtiva.

Além disso, este editor será um editor orientado por gabaritos (arquivo de auto conclusão dos comandos mais utilizados), a fim de poupar esforço do programador, onde depois de digitado um comando (como por exemplo o comando "IF") e pressionando uma tecla especial (a qual poderia ser F1) este editor coloca o gabarito para o comando na tela e o programador que está usando o protótipo deste software só necessitaria preencher as lacunas (no caso do comando "IF", condição e comandos a serem executados).

Para a especificação do protótipo, será utilizada a Análise Essencial, auxiliada pela ferramenta Power Designer – Process Analyst.

A implementação consistirá na criação de um software escrito no ambiente de programação Delphi que fará a transferência destes programas-fonte através da rede TCP/IP para a máquina local. Após isto feito, acontecerá a edição do programa-fonte propriamente dita, auxiliada por gabaritos (para os comandos mais usados apenas). No momento em que é salvo o programa-fonte, acontece então uma nova transferência com sentido inverso, ou seja, da máquina local para a máquina Servidora. Será possível também, utilizar este software para abrir arquivos locais.

Como a transferência de arquivos utiliza-se do protocolo TCP/IP, este editor poderá servir para fazer a edição de programas fonte localizados em qualquer máquina conectada diretamente (com um endereço IP verdadeiro) à Internet.

Foram encontrados em [TUC99], aproximadamente 25 editores. Nenhum dos ambientes que foram pesquisados possuem todas as características desejadas e nenhum deles possui realce colorido da sintaxe para a linguagem Dataflex. Porém, foram encontrados editores que possuem, isoladamente, as características almejadas.

Foi feita uma análise dos seguintes editores: FTP Edit, EditPlus+, TextPad e Aranae. Uma análise mais detalhada sobre eles será encontrada no capítulo 2.

Algumas características da Linguagem Dataflex também deverão ser estudadas para se obter o que este documento propõe, como comandos e estruturação dos programas.

1.1 OBJETIVOS

O objetivo principal do trabalho é implementar um protótipo de um software para facilitar a edição de programas fonte do Dataflex orientado por gabaritos.

Os objetivos secundários do trabalho são:

- a) conhecer o funcionamento de gabaritos;
- b) facilitar a tarefa de edição deste programas pelos programadores, usando uma interface simples, intuitiva e com muita operacionalidade;

- c) aprofundar o estudo do ambiente de programação Delphi, bem como da transferência de arquivos através da rede TCP/IP (FTP).

1.2 DESCRIÇÃO DA EMPRESA

A empresa na qual será realizado este estágio é a Prefeitura Municipal de Jaraguá do Sul, que é uma empresa de administração pública municipal.

Na Divisão de Processamento de Dados, daqui por diante denominada simplesmente DPD, trabalham 9 funcionários:

- a) o Diretor de Divisão, que desempenha as funções de Direção e Análise de Sistemas;
- b) dois Operadores de Computador, responsáveis pela instalação e manutenção dos computadores da prefeitura, e pela administração do servidor HP(Unix);
- c) um Webmaster, que é o responsável pelo Web Site da prefeitura;
- d) um Auxiliar de Informática, responsável pela digitação de textos, elaboração de cartazes, processamento de disquetes de relógios ponto, etc.;
- e) quatro programadores, responsáveis pela implementação dos sistemas existentes (IPTU, Folha de Pagamento, Ouvidoria, Patrimônio, Alvarás, Fiscalização, etc.)

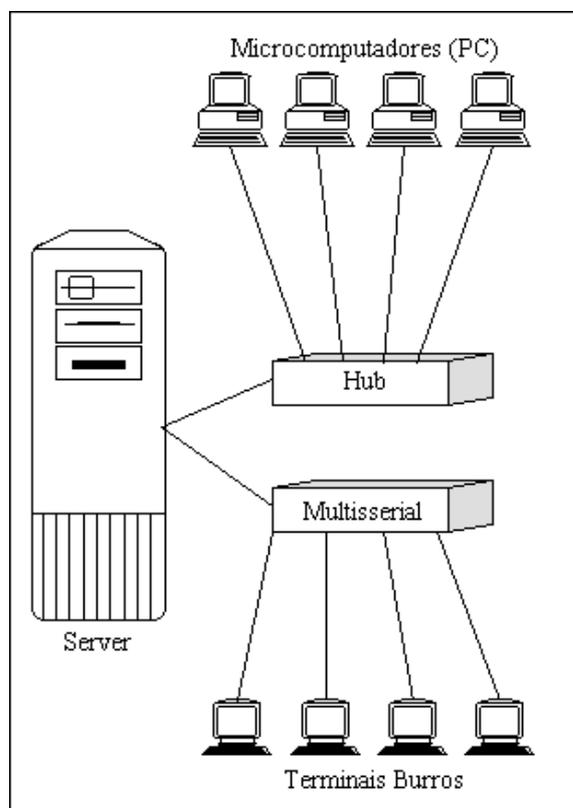
Quanto aos equipamentos, existem basicamente 3 tipos:

- a) microcomputadores;
- b) terminais seriais não processados (burros);
- c) servidor.

Deles, o mais relevante a ser descrito aqui, é o servidor. É uma máquina RISC, cujo fabricante é a Hewlett-Packard (HP). Tem 2 processadores RISC de 64bits, 512 Mb de memória RAM, dois discos rígidos SCSI de 9Gb cada, unidade de fita DAT, placas MUX (para a conexão com os terminais) e placa de rede para conexão aos microcomputadores através do protocolo TCP/IP (Figura 1).

Os programas fonte da linguagem Dataflex, bem como as bases de dados e utilitários desta linguagem localizam-se neste servidor.

Figura 1: Ligação dos terminais e microcomputadores ao servidor na Prefeitura



1.3 ORGANIZAÇÃO DO TEXTO

No capítulo 2 estarão descritas as informações de funcionamento de editores orientados por gabaritos / sintaxe.

No capítulo 3 estarão descritas as informações necessárias sobre Análise Essencial, que será necessária para a especificação do protótipo.

No capítulo 4 estarão as informações referentes a linguagem Dataflex como: estruturação, tipos de comandos, utilitários, etc...

No capítulo 5 está a especificação do protótipo deste sistema de edição de programas dataflex através de gabaritos / sintaxe.

No capítulo 6 estarão descritos alguns detalhes referentes a implementação deste sistema utilizando o ambiente de programação Delphi.

No capítulo 7 estão as conclusões e sugestões de continuação deste trabalho.

2 EDIÇÃO ORIENTADA POR GABARITOS

Este capítulo trata sobre o material encontrado sobre edição orientada por gabaritos. Como não foram encontrados textos que falassem explicitamente sobre o assunto, o que será descrito aqui, são as conclusões que foram obtidas analisando-se o software EditPlus+1.25, TextPad4, Araneae e FTP Edit 2.21.

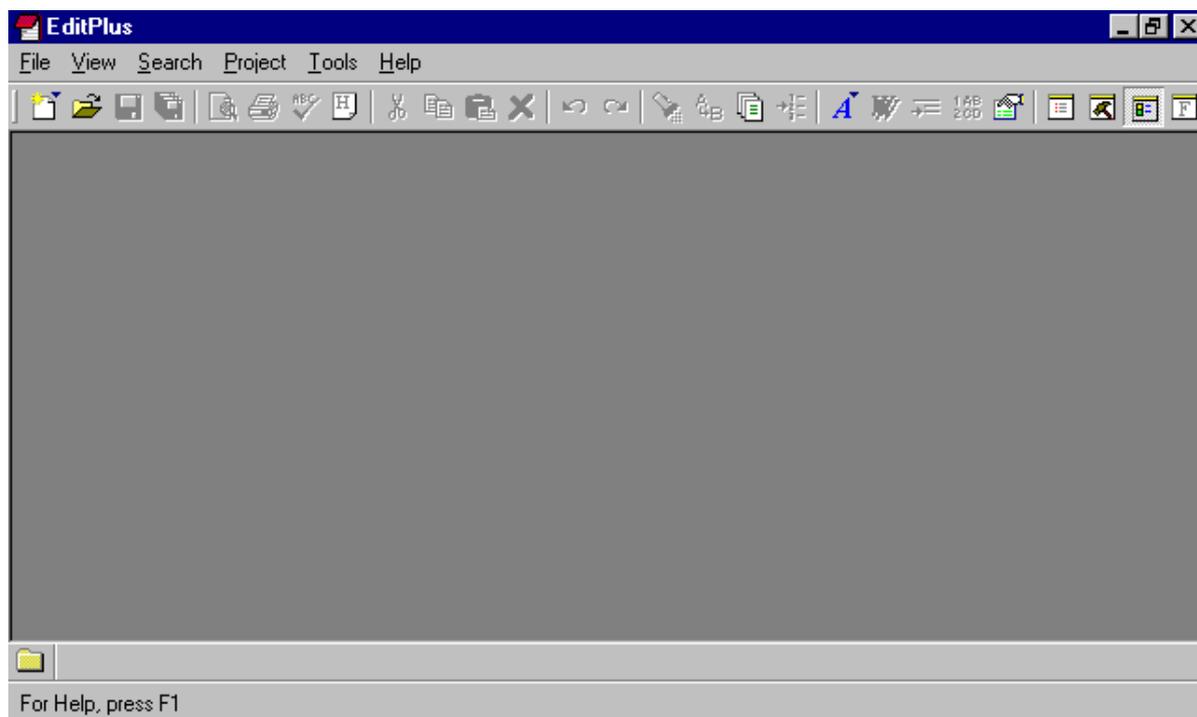
2.1 EDITPLUS+ 1.25

Segundo [EDI99], a empresa fabricante deste software (EditPlus+), é a ES-Computing, e este é seu único produto. Com base nas sugestões dos programadores da Prefeitura Municipal de Jaraguá do Sul, e a facilidade de uso do EditPlus+, foi que surgiu a idéia de implementar um editor de programas específico para a linguagem Dataflex e para Prefeitura Municipal de Jaraguá do Sul.

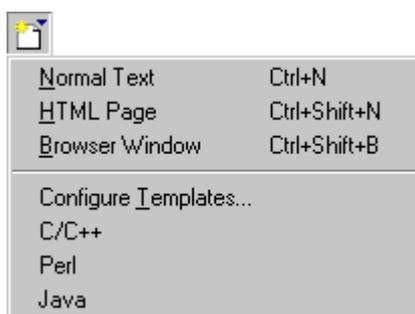
As principais características deste software são:

- a) *Syntax Color Highlighting*: significa que os comandos, strings, operadores, etc... serão coloridos com cores diferentes para facilitar a visualização pelos programadores;
- b) *Auto Completion*: para alguns comandos apenas, são inseridos gabaritos para auto-completar sua sintaxe e reduzir o tempo de digitação;
- c) *Mostrar número das linhas*: mostra no lado esquerdo o número das linhas;
- d) *Aumentar ou Reduzir indentação*: move o texto selecionado para a esquerda ou para a direita, através da inserção de espaços na frente da linha. A quantidade de espaços a ser inserida ou removida pode ser configurada pelo usuário;
- e) *Inserção*: é possível inserir data e hora atuais em diversos formatos; inserir outros arquivos dentro do arquivo atual; inserir quebras de página.

Na figura 2 pode ser vista uma tela inicial do EditPlus+.

Figura 2: Tela Inicial do EditPlus+

No EditPlus+ 1.25 [EDI99], ao clicar no botão New, da barra de ferramentas (ver Figura 3), é aberto um menu com opções de criação de novos arquivos. Entre os tipos pré definidos estão textos normais, páginas HTML, programas fonte em C e/ou C++, Perl e Java. A opção “Browser Window” abre uma sub-janela como se fosse um browser, para poder visualizar os arquivos HTML, sem ser necessário sair do programa.

Figura 3: Menu do botão “New” do Editplus+.

A opção “Configure Templates”, mostra uma caixa de diálogo na qual se pode configurar novos “Templates”. Neste software (EditPlus), um “Template” é um arquivo de exemplo muito simples, que aparece numa nova janela de edição, quando um arquivo novo é criado. Um exemplo é mostrado na Figura 4. Este “Template” pode ser modificado ao gosto do usuário.

Figura 4: “Template” do novo arquivo HTML.

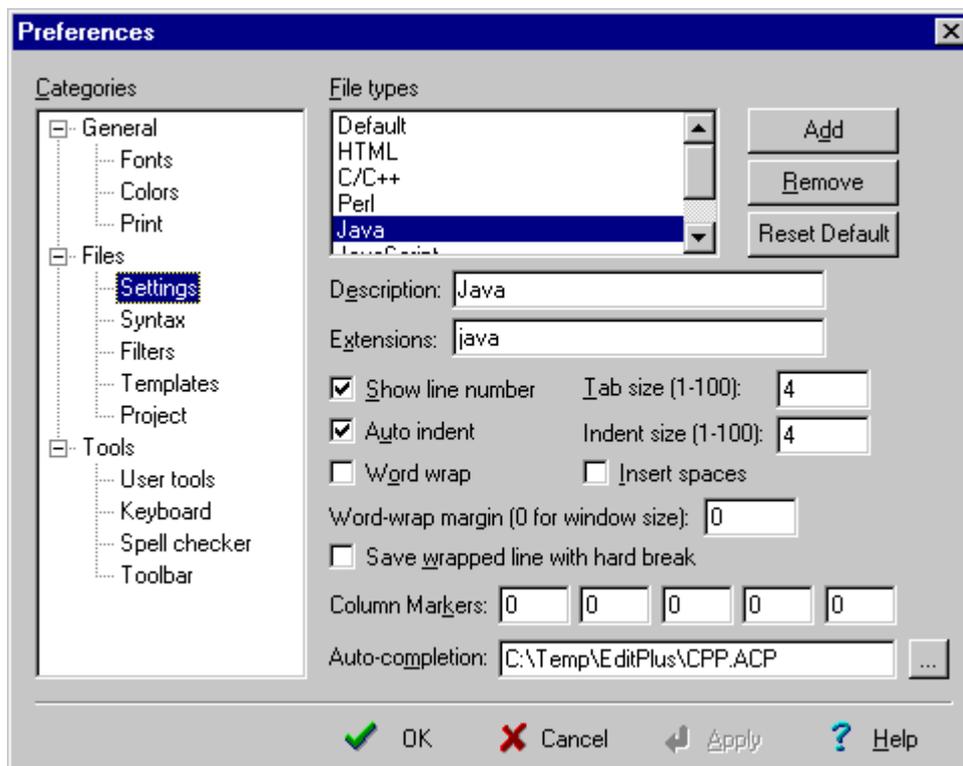
```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<HTML>
<HEAD>
<TITLE> New Document </TITLE>
<META NAME="Generator" CONTENT="EditPlus 1.2">
<META NAME="Author" CONTENT="">
<META NAME="Keywords" CONTENT="">
<META NAME="Description" CONTENT="">
</HEAD>

<BODY BGCOLOR="#FFFFFF">

</BODY>
</HTML>
```

Ao clicar-se em “Configure Templates” através do botão “New”, ou no menu em Tools – Preferences, é aberta uma caixa de diálogo de configuração do programa. A caixa de diálogo na opção “Settings” é mostrada na Figura 5. Na parte superior aparece File types, onde pode-se definir novos tipos de arquivos (atribuir extensões a determinados tipos de arquivos) e o que é mais importante, na parte inferior desta caixa, aparece um campo chamado Auto-Completion, que é o arquivo com os gabaritos de comandos mais utilizados.

Figura 5: Caixa Preferences do EditPlus+.



No Quadro 1 está um exemplo do arquivo de “Auto-Completion” do EditPlus para programas fonte da linguagem C, C++ e Java. Nesse arquivo estão os gabaritos para os principais comandos da linguagem. Daqui por diante esse arquivo será chamado de Arquivo de Gabaritos.

Gabarito será aquele pedaço de texto que será inserido assim que o usuário digitar uma das palavras chaves (comandos principais), que é a conclusão automática do comando (*auto completion*), deixando para o usuário apenas o trabalho de preencher as lacunas com os comandos que desejar. Neste software, o gabarito inserido, já aparece nas cores corretas, baseado no arquivo de definição de sintaxe e indentado na posição correta.

Quadro 1: Arquivo de “Auto-Completion” do EditPlus+ para C, C++ e Java.

```

1 #TITLE=C/C++
2 #T=if
3 if (^!)
4 {
5 }
6 #T=while
7 while (^!)
8 {
9 }
10 #T=for
11 for (^!; ; )
12 {
13 }
14 #T=do
15 do
16 {
17 }
18 while (^!);
19 #

```

Analisando o conteúdo do Arquivo de Gabaritos (Quadro 1), pode-se identificar na primeira linha uma cláusula que indica o título. A partir da linha n.º 2 inicia-se a construção dos gabaritos. As linhas iniciadas pelos caracteres "#T=" são linhas de declaração dos gabaritos. Na linha de n.º 2 está escrito o gabarito do comando "IF". Então, se o programador que estiver editando um arquivo da linguagem C, C++ ou Java, escrever um comando "IF" e pressionar a barra de espaços (no caso do EditPlus+), na posição onde estava o cursor, serão inseridas as linhas que seguem a linha de n.º 2, até encontrar uma nova linha iniciada por "#T=", portanto as linhas de n.º 3, 4 e 5. Um exemplo de como isso acontece pode ser visto na Figura 6.

Figura 6: Antes e Depois da inserção do gabarito

<pre> 1 class 2 { 3 public static void main(String[] args) 4 { 5 System.out.println("Hello World!"); 6 if 7 } 8 } 9 </pre>	<pre> 1 class 2 { 3 public static void main(String[] args) 4 { 5 System.out.println("Hello World!"); 6 if {} 7 { 8 } 9 } 10 } 11 </pre>
---	---

As diferentes cores nas quais aparecem os comandos, são baseadas em um arquivo denominado Syntax Definition File (arquivo de definição de sintaxe). Neste caso, é possível ter arquivos de definição de sintaxe separados para cada um dos três tipos de arquivo acima citados (C, C++ e Java). Neste arquivo, devem estar todas as palavras reservadas da linguagem na qual se queira programar posteriormente. As palavras podem estar separadas em classes para deixar, por exemplo, os comandos em uma cor e os operadores em outra. O arquivo (resumido) de definição de sintaxe da linguagem C / C++ para o EditPlus+ (existe um arquivo separado para a linguagem Java) pode ser observado no Quadro 2.

Quadro 2: Arquivo de Definição de Sintaxe do EditPlus+ para linguagem C e C++

```
#TITLE=C/C++
; EditPlus Syntax File v1.2 written by ES-Computing.
; This file is provided as a default syntax file for C/C++.

#DELIMITER=,(){}[]-+*%/=" '~!&|<>?:;.#
#QUOTATION1='
#QUOTATION2="
#LINECOMMENT=//
#LINECOMMENT2=
#COMMENTON=/*
#COMMENTOFF=*/

; Reserved words
#KEYWORD
break
case
char
do
double
else
float
for
getchar
goto
if
int
long
printf
public
while
; Compiler directives
#KEYWORD
define
error
include
```

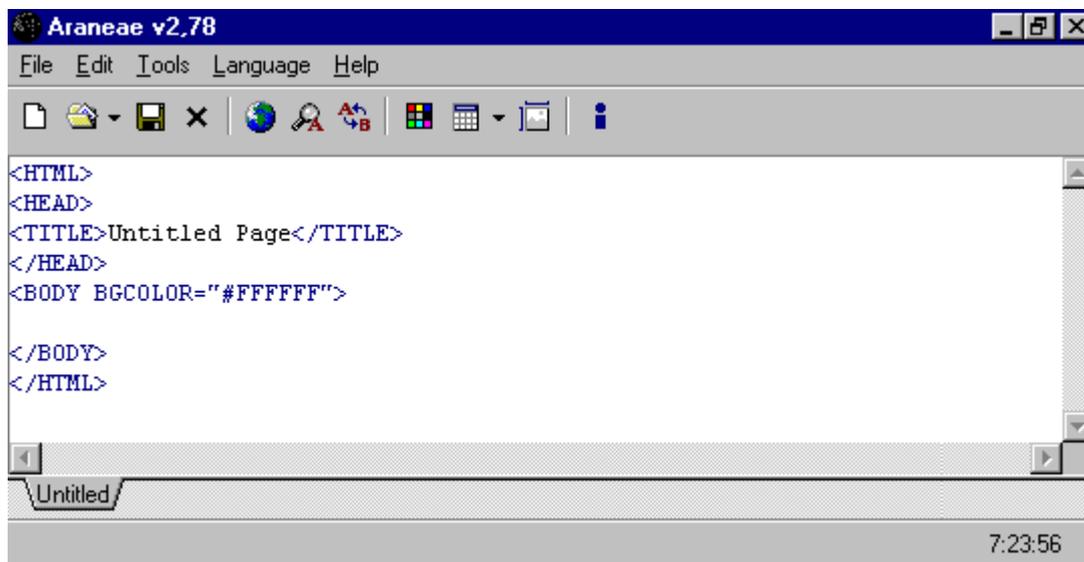
Na parte superior, pode-se observar as definições do que são os delimitadores (e outros símbolos especiais), os tipos de aspas (quotation), os caracteres que definem comentário de linha, caracteres de início e fim de comentário, e outros.

Na parte inferior, aparecem algumas das palavras reservadas (comandos / instruções) da linguagem, e diretivas de compilação. A cada linha que aparece #KEYWORD, a cor muda e a configuração das cores pode ser alterada através do menu Tools - Preferences.

2.2 ARANEA 2.78

Este é um editor criado por um estudante universitário americano [ARA99] e possui algumas funcionalidades específicas para edição de arquivos HTML. Ao executar o software, já aparece um arquivo HTML base. Isto pode ser visto na Figura 7.

Figura 7: Janela Principal do Editor Araneae

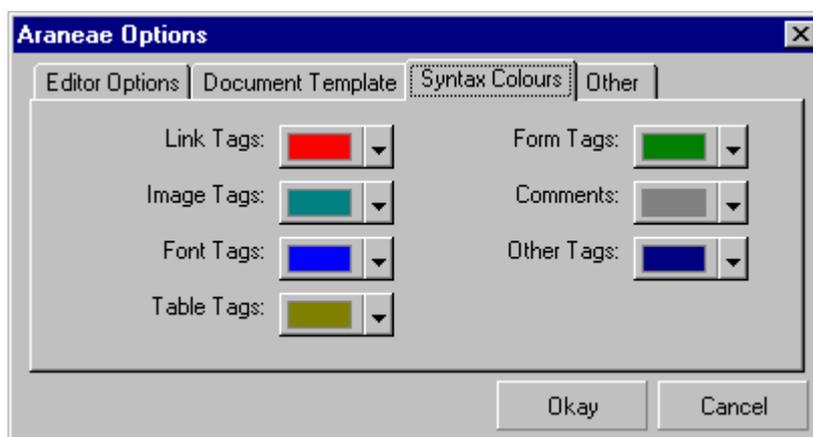


Assim, como o EditPlus+, este editor também possui aquela característica chamada *syntax color highlighting*, que modifica somente as cores das tags HTML e textos entre aspas.

É possível editar outros arquivos, só que a alteração de cores não funciona para estes tipos de arquivos.

Este editor possui algumas ferramentas específicas para edição de arquivos HTML como seleção de cores, seleção de tamanho da imagem, visualizar arquivo em um browser e inserção de data/hora. Neste editor, é possível configurar as cores para determinadas tags HTML. A tela de configuração de cores pode ser vista na Figura 8.

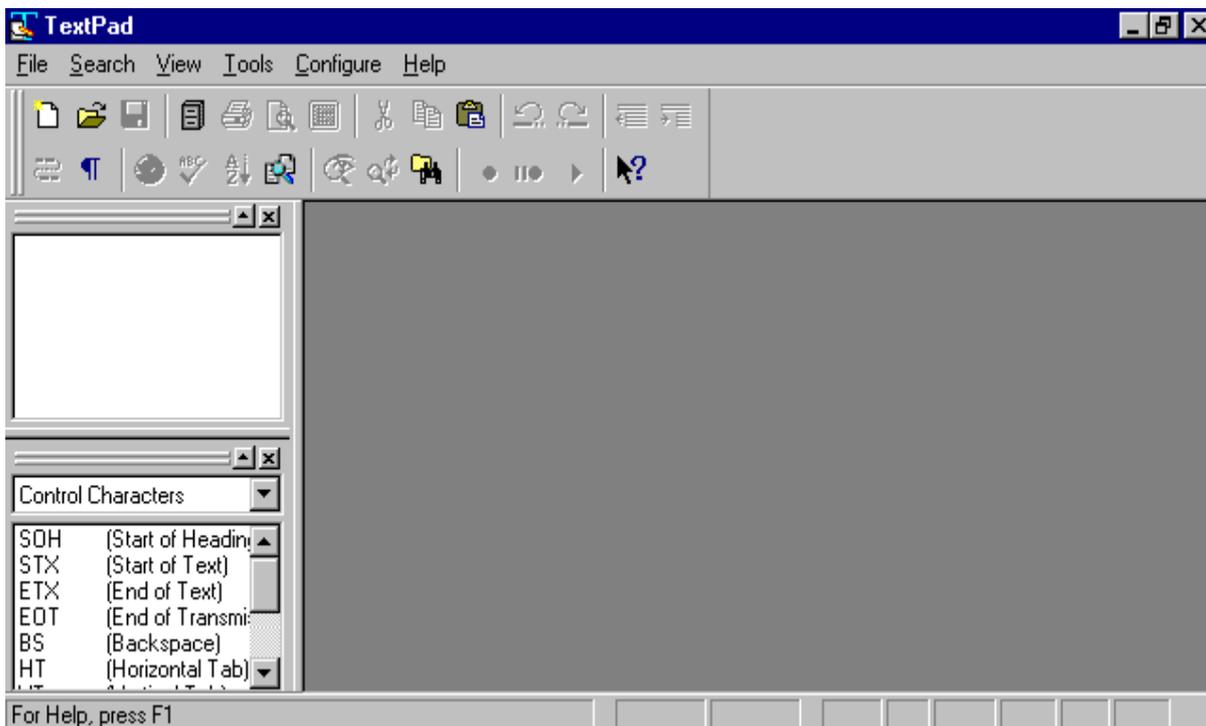
Figura 8: Tela de Configuração de Cores do Araneae



Assim como o Araneae, o software EditPlus+, também possui uma tela de configuração de cores. Ao lado de cada uma das caixinhas de cores, aparece o tipo da tag HTML que será colorida. No caso EditPlus, como ele não é um editor específico para HTML, aparecem as palavras “Keyword 1”, “Keyword 2”, ... “Keyword n”.

2.3 TEXTPAD 4

O TextPad 4 é desenvolvido por uma empresa chamada Helios Software Solutions Inc. e dentre estes quatro editores analisados, o TextPad 4 foi o editor com a maior quantidade de recursos. A tela inicial do TextPad 4 pode ser vista na Figura 9.

Figura 9: Tela inicial do TextPad 4.

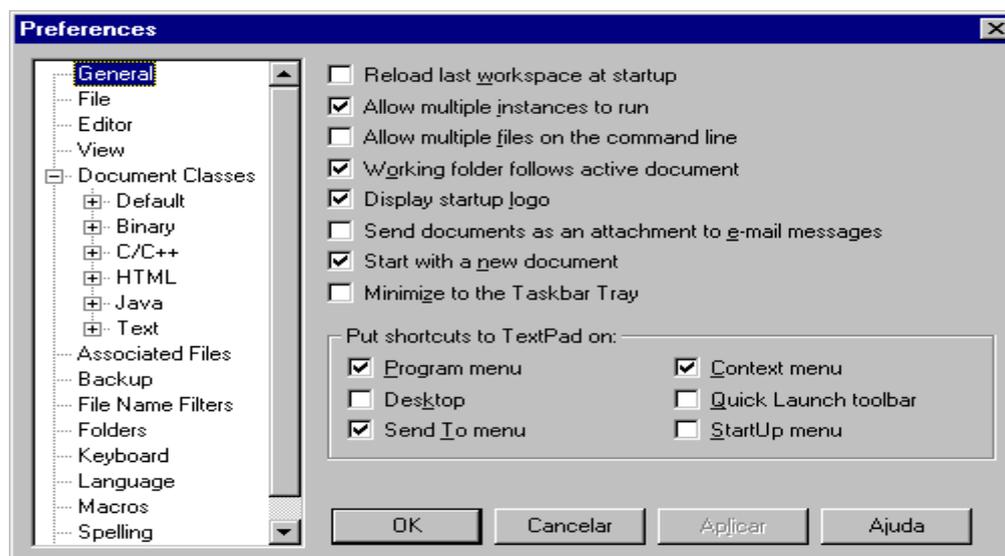
Este editor possui as ferramentas que o EditPlus+ possui e mais algumas. As principais ferramentas são:

- a) inserção de caracteres especiais: no canto esquerdo inferior, é possível observar uma caixa com caracteres de controle. Ali é possível selecionar também, caracteres ANSI, caracteres HTML e tags HTML;
- b) permite selecionar texto em colunas (EditPlus+ também possui esta característica);
- c) selecionador de Documentos: no canto esquerdo superior aparece um quadro com o nome dos arquivos abertos (os editores anteriores também possuem esta característica, porém em outra posição);
- d) comparador de Arquivos: compara dois arquivos mostrando suas diferenças;
- e) ordenação: permite classificar o arquivo em ordem ascendente, descendente, etc...

- f) maiúsculas/Minúsculas: possibilita o usuário alterar um texto para letras maiúsculas, minúsculas, por sentença e palavra (EditPlus+, também possui esta característica);
- g) caracteres Ocultos: mostra ou esconde caracteres ocultos como Enter, Tab e Espaços;
- h) ferramentas: pode-se configurar ferramentas personalizadas como por exemplo, configurar uma tecla especial para ativar um compilador java a partir da tela do editor (EditPlus+, também possui esta característica);
- i) convert to DOS: se o arquivo que está sendo editado foi primeiramente editado num sistema operacional UNIX, ele pode converter caracteres incompatíveis com o DOS para caracteres compatíveis com o DOS.
- j) minimizar para a bandeja: quando minimizado, não aparece na barra de tarefas do Windows, e sim na bandeja (lugar onde geralmente aparecem ícones de programas que ficam residentes na memória como tradutores, anti-vírus, etc.)

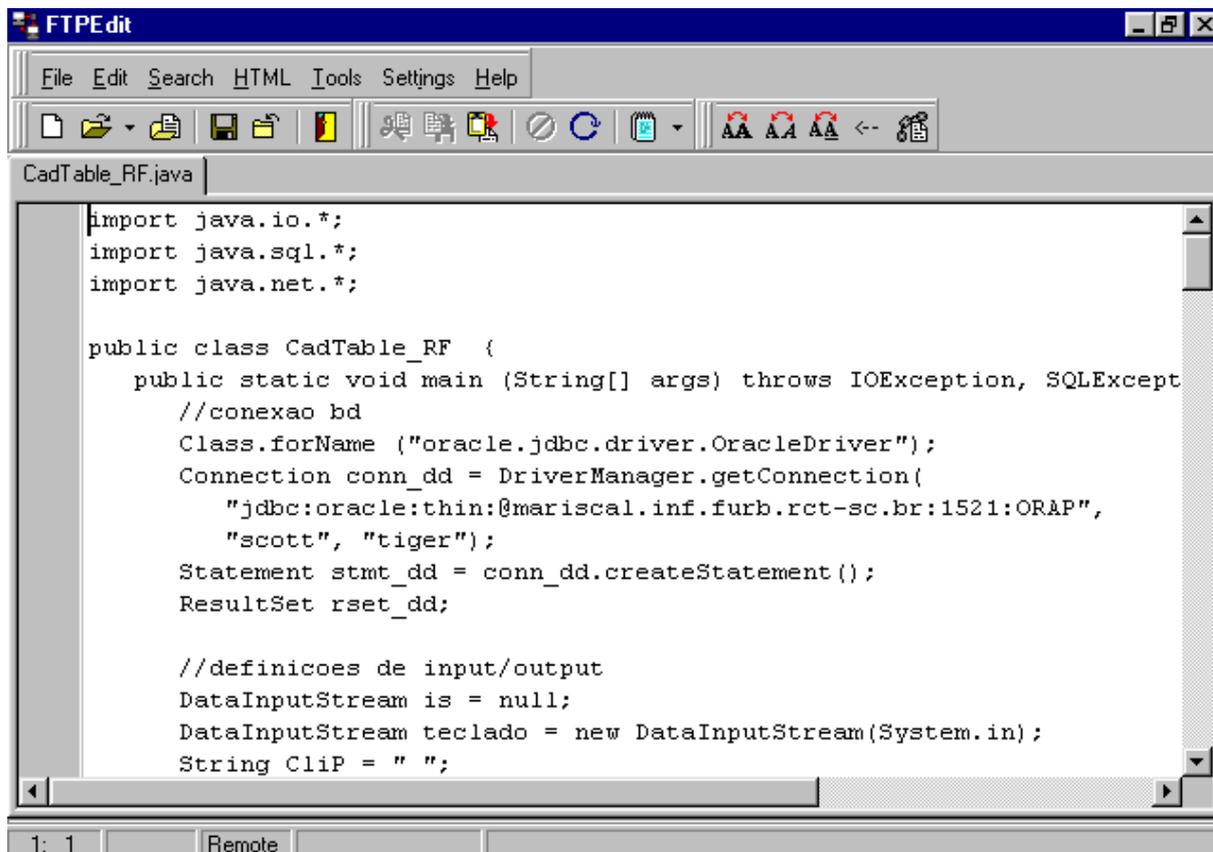
Este editor já vem com arquivos de definição de sintaxe para C, C++, Java, Cobol, HTML, LaTeX, Lisp, Pascal e Basic. Uma deficiência deste software é que ele não possui a característica de “Auto-Completion” que o EditPlus+ possui. E também, nenhum deles abre arquivos de servidores FTP. A janela de configurações pode ser vista na figura 10.

Figura 10: Janela de Configurações do TextPad 4



2.4 FTP EDIT 2.21

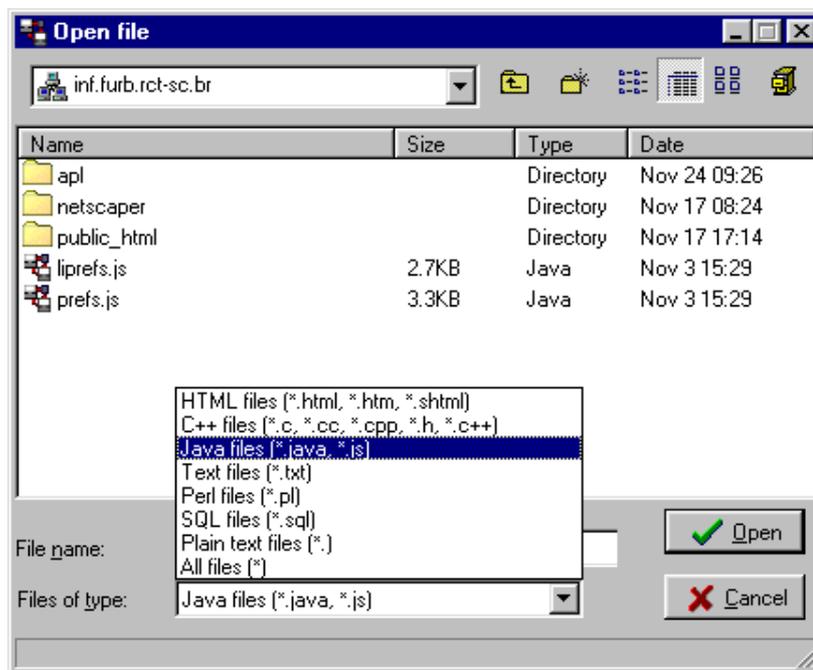
O FTP Edit, é um software desenvolvido por Kristoffer Henriksson [FTP99], e foi o único software da categoria que possibilita editar arquivos de um servidor FTP. Um programa fonte em java aberto de um servidor FTP pelo FTP Edit pode ser observada na figura 11.

Figura 11: Janela do FTP Edit.

Ele possui realce colorido de sintaxe para arquivos SQL, PERL e HTML. Apesar de na janela de abertura de arquivos (ver figura 12), existirem opções para abrir arquivos do tipo JAVA e C/C++, o realce colorido de sintaxe para estes arquivos não funcionou.

Ele possui uma tela de configuração do servidor FTP. Podem ser incluídos vários servidores, cada qual com seu nome de usuário e senha, e estes servidores FTP podem ser acessados facilmente (figura 12), através da tela de abertura de arquivos.

Figura 12: Tela de Abertura de Arquivos do FTP Edit



Na pasta onde este software foi instalado, não foi encontrado nenhum arquivo de definição de sintaxe ou gabarito. Ele também não possui suporte a auto conclusão de comandos, como o EditPlus+ possui.

Com relação à edição de arquivos HTML, possui algumas (poucas, comparadas ao EditPlus+) ferramentas específicas para inclusão de tags HTML. Na figura 13, podem ser vistas quais são estas ferramentas.

Figura 13: Ferramentas do FTP Edit para HTML



3 ANÁLISE ESSENCIAL

Durante muito tempo houve debates entre os profissionais de desenvolvimento de sistemas sobre por qual perspectiva se deveria começar a especificação de um sistema: pelos dados ou pelas funções? Segundo [POM95], hoje, com uma visão mais abrangente do problema, pode-se responder a esta indagação: “devemos começar a especificação de um sistema pela identificação dos eventos que o afetam”. A rigor, o evento é a pedra fundamental de um sistema [POM95].

A análise essencial, propõe o particionamento do sistema em eventos. A rigor, o valor de um sistema está na sua capacidade de responder com eficácia a todos os estímulos a que for submetido. Assim, um sistema é construído para responder a estímulos. A cada estímulo, o sistema deve reagir produzindo uma resposta predeterminada.

A análise essencial divide-se em duas partes principais: o modelo essencial e o modelo de implementação, que serão detalhados abaixo.

3.1 MODELO ESSENCIAL

Uma das primeiras atividades na análise, envolve o desenvolvimento de um modelo essencial do sistema. Conforme [GRA99], o Modelo Essencial é um modelo da essência do sistema do usuário, o que é necessário para que o sistema satisfaça os requisitos propostos, ou seja, um modelo do que o sistema deveria fazer, não importando a metodologia selecionada para sua implementação.

McMenamim [McM91], revela que o modelo essencial deve conter todos os requisitos válidos e nada além destes. Caso isto não ocorra, o sistema provavelmente apresentará duas deficiências: não conterà todos os requisitos válidos e conterà requisitos inválidos.

O modelo essencial consiste em dois modelos principais: o modelo ambiental e o modelo comportamental. O primeiro mostra como o sistema interage com o ambiente externo e o outro, o que deve ser feito para interagir de maneira satisfatória a esse ambiente.

3.1.1 MODELO AMBIENTAL

Conforme [GRA99], a principal função deste modelo é documentar e esclarecer os dados que devem ser processados, os dados que devem ser produzidos e as entidades que fornecerão dados para o sistema ou consumirão dados do sistema.

Os sistemas são racionais e objetivos, produzindo saídas como resposta a um evento. A essas interações entre ambiente e sistema dá-se o nome de evento e resposta. Como descrito anteriormente, um evento pode ser definido como um acontecimento do mundo exterior que requer uma resposta, e uma resposta é um conjunto de ações executadas pelo sistema sempre que um determinado evento ocorre. Enfim, pretende-se neste modelo, identificar os eventos ocorridos no ambiente externo e que exigem uma resposta do sistema.

Segundo [GRA99], a finalidade do sistema e as políticas do usuário são entradas do Modelo Ambiental que conseqüentemente possui duas saídas: a Lista de Eventos e o Diagrama de Contexto (DC). Este último, é um Diagrama de Fluxo de Dados (DFD), que possui apenas um processo representando todo o sistema. Os fluxos de dados mostrados no DC representam os fluxos de dados entre o sistema e com quem ele se comunica (outros sistemas, pessoas, etc...), que também são chamados de Entidades Externas. Um Evento pode ser definido informalmente como um acontecimento do mundo exterior que requer do sistema uma resposta. A Lista de Eventos é uma ferramenta que ajuda a examinar o que acontece no ambiente (sobre qual o sistema não tem controle). É possível obter a definição de como o sistema vai reagir aos eventos, desde que a Lista de Eventos tenha sido elaborada.

O primeiro passo na análise de um sistema é descobrir quais os eventos que vão provocar os estímulo aos quais o sistema deve responder. Normalmente, só se consideram eventos externos ao sistema, ou seja, aqueles sobre cuja ocorrência não se pode ter qualquer interferência; mais precisamente, são completamente independentes do sistema. No Quadro 3 é exemplificada uma Lista de Eventos.

Quadro 3: Exemplo de uma Lista de Eventos

- | |
|--|
| <ol style="list-style-type: none">1. Cliente solicita cadastramento.2. Cliente efetua pagamento.3. É hora de emitir relatório gerencial para controle. |
|--|

O segundo passo na análise de um sistema é elaborar o Diagrama de Contexto. Como tem-se a Lista de Eventos, baseia-se nela.

O diagrama de contexto é um diagrama que representa apenas um sistema (processo) e suas interfaces (entidades externas), e é criado utilizando-se apenas dois dos símbolos usados para a construção de diagramas de fluxo de dados (DFD), que será visto mais abaixo, na descrição de cada um dos componentes do DFD.

3.1.2 MODELO COMPORTAMENTAL

Neste modelo é feito o trabalho de construção dos diagramas de fluxo de dados (DFD's), modelos de entidade-relacionamento (MER's), especificação de processos e dicionário de dados(DD). O modelo comportamental descreve o comportamento do interior do sistema necessário para interagir adequadamente com o ambiente.

No modelo comportamental, cada evento identificado será apresentado por um DFD particionado. Posteriormente, no refinamento do modelo, o evento poderá ser decomposto, ou ainda, vários eventos afins podem ser agregados.

Confome [POM95], um DFD é uma forma gráfica de mostrar a interdependência das funções que compõem um sistema, apresentando fluxo de dados entre elas. Mostra ainda os arquivos lógicos de dados, que são denominados depósitos de dados, como será visto mais adiante, bem como as entidades externas, denominação dada tanto à origem dos fluxos de dados que chegam ao sistema, como ao destino dos fluxos que dele partem.

A notação gráfica utilizada é a que é apresentada por Gane e Sarson [GAN83], que será mostrada abaixo na Figura 14. O DFD é composto de quatro elementos gráficos:

- a) fluxo de dados: mostra como os processos são interligados. No DFD são representados por setas que indicam a direção do tráfego dos dados e que são nomeados de acordo com o conteúdo dos dados transportados;
- b) processos: representa um componente procedural do sistema, onde os dados sofrem alguma alteração. São representados por retângulos arredondados ou elipses, e também recebem um nome de acordo com a sua função;

- c) entidades externas: são as origens / destinos dos dados processados no sistema. Podem representar uma pessoa, um departamento ou outro sistema. No DFD são representados por um retângulo ou um quadrado.
- d) depósitos de dados: representação de um arquivo lógico ou um dispositivo de armazenamento de arquivos. São representados por linhas paralelas e horizontais.

Na figura 15 está representado um Diagrama de Contexto (DC) de exemplo, utilizando alguns dos elementos gráficos mostrados na figura 14.

Figura 14: Elementos Gráficos de um DFD.

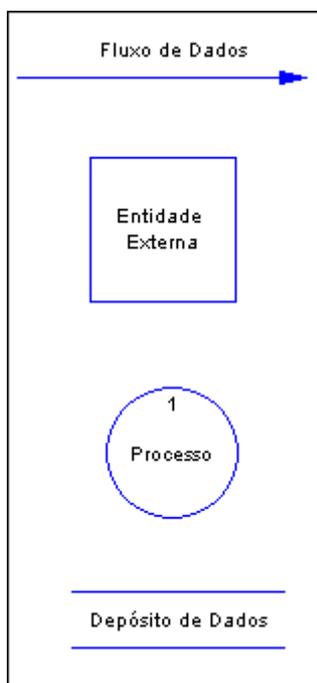
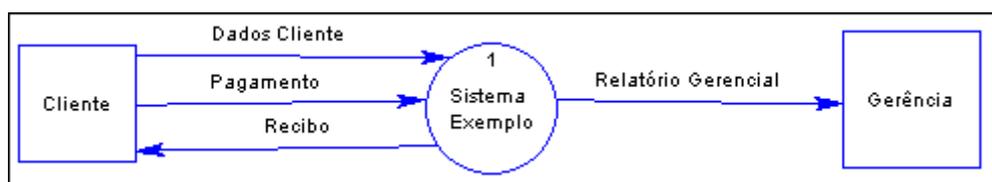


Figura 15: Exemplo de um DC.



O próximo passo é fazer baseando-se nos eventos da Lista de Eventos os DFD's (Diagramas de Fluxo de Dados) particionados. Isto pode ser visto nas figuras 16, 17 e 18. E na Figura 19, pode-ser observado o Diagrama de Fluxo de Dados Completo do Sistema Exemplo.

Figura 16: DFD do Evento Cliente Solicita Cadastramento

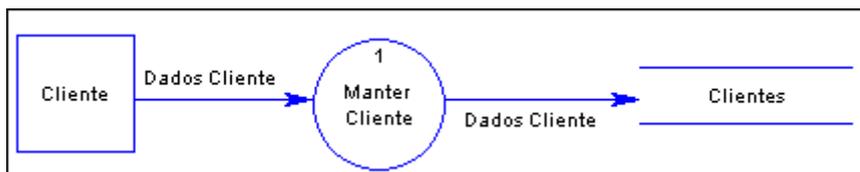


Figura 17: DFD do Evento Cliente Efetua Pagamento

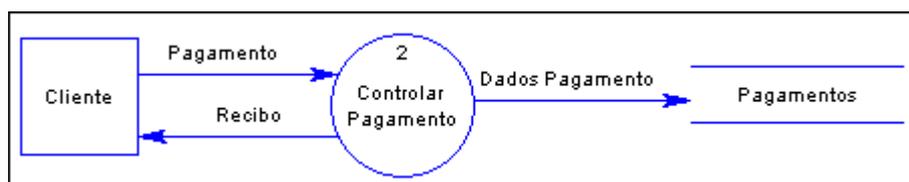


Figura 18: DFD do Evento É hora de emitir relatório gerencial

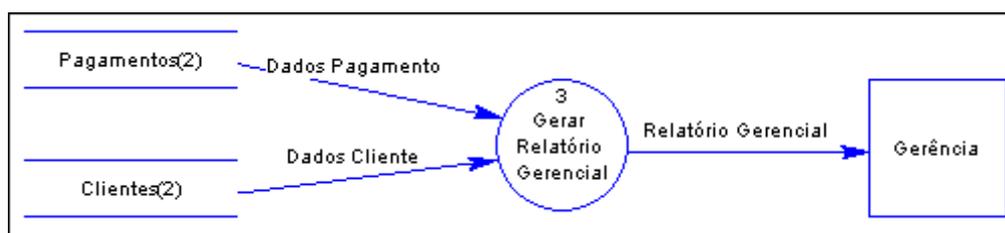
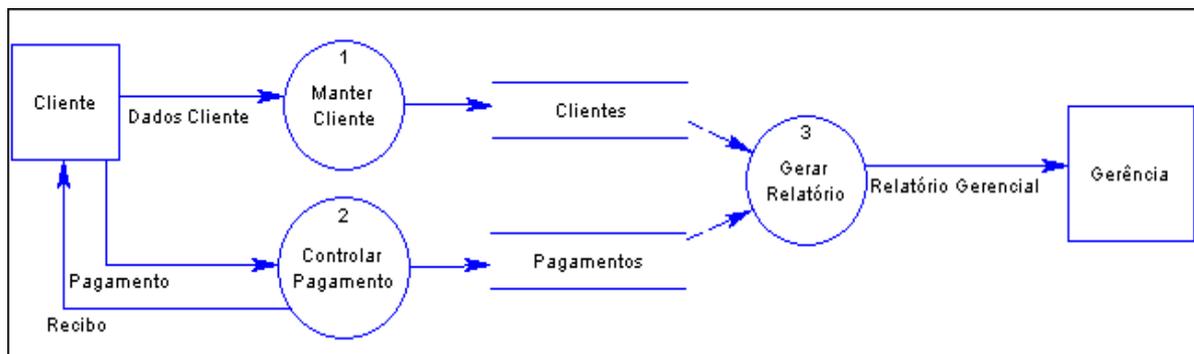


Figura 19: Diagrama Completo de Fluxo de Dados do sistema Exemplo



3.1.3 DICIONÁRIO DE DADOS (DD)

O Dicionário de Dados(DD) é quem fornece a informação de texto de suporte para auxiliar a informação gráfica mostrada no DFD. O DD é um grupo organizado de definições de todos os elementos de dados que estão envolvidos no sistema que está sendo modelado.

3.1.4 MODELO ENTIDADE RELACIONAMENTO (MER)

O Modelo Entidade Relacionamento (MER) enfatiza os principais objetos(Entidades) com que o sistema lida, bem como a relação entre eles. Os objetos(Entidades) no MER correspondem normalmente aos locais de armazenagem de dados(Depósitos) no DFD. O DFD não mostra a relação entre os objetos. O MER fornece uma visão simples e gráfica do sistema para certos usuários que não se importam muito com os detalhes de processamento(funcionais) do sistema. Dá ênfase para o que não é óbvio no DFD. O MER fornece a entrada para os projetistas de banco de dados que devem eventualmente decidir qual a melhor forma de implementar os requisitos de armazenamentos do sistema.

3.2 MODELO DE IMPLEMENTAÇÃO

O Modelo de Implementação serve para identificar algumas atividades essenciais, buscando tornar mais aparentes as principais decisões de implementação nele contidas. Ele deve ser uma planta para orientar, de modo preciso e completo, as etapas restantes do processo de construção do sistema.

Conforme [GRA99], basicamente este modelo abrange quatro aspectos:

- a) alocação do Modelo Essencial x Máquinas: determinação de quais dados serão manipulados manualmente e quais serão automatizados.;
- b) detalhes da interação Homem / Máquina: relacionado com os dispositivos de entrada e saída bem como de seus formatos;
- c) atividades manuais adicionais que podem vir a ser necessárias;
- d) restrições operacionais que o usuário deseja impor ao sistema.

4 DATAFLEX

Conforme [DAT99], DataFlex é uma linguagem de 4.^a geração, projetada para o desenvolvimento de aplicações que utilizam Banco de Dados.

Existem versões do Dataflex para as seguintes plataformas: DOS, Novell, Windows 32 bits, IBM/AIX, HP-UX, SCO Unix, Linux, Intel Based Unix, SUN Solaris, Compaq Digital, MIPS ABI Compliant. A empresa que produz o Dataflex (DataAccess), está há mais de vinte anos no mercado na criação de produtos para o aumento da produtividade no desenvolvimento de aplicativos que trabalham com banco de dados. [DAT99]

As características principais do Dataflex são:

- a) suporte total ao Bug do Milênio [DAT99];
- b) não gera programas executáveis, apenas um arquivo objeto;
- c) é necessário um programa adicional (DFRUN) para rodar os programas compilados.

4.1 PROGRAMA-FONTE

Um programa fonte do Dataflex é dividido em duas partes principais: a definição de Imagens e o programa propriamente dito, separados por uma linha assim:

```
/*
```

No Quadro 4 está um programa exemplo.

Quadro 4: Programa fonte exemplo do Dataflex.

```

/Tela1
+-----+
| Confirma a Impressão do Relatório: _ (S)im/(N)ao |
+-----+
/Pronta
+-----< Impressora: _ (012) >--+
/Header
+-----+
PMJS/DPD          Relatório Simplificado de Funcionarios    __/__/____ __:__
                                     Pagina..: _____.

  Cadastro Nome
+-----+
/Body
_____.-_ _____
/Total
+-----+
Total de Funcionarios Impressos: _____.
+-----+
/*
#include /usr/dataflex/comandos.inc
Page Set Tela1 at 03 00
Open FPCAD

Inicio:
  Page Tela0
  Accept Tela1.1 { Capslock Check="SN0" }
  If Tela1.1 in "N0" System

  Showtime Header.1 Header.2 Header.3
  Outfile "/us0/fp/s/fpcadr01.spl"
  Gosub Imp-Header

  Clear FPCAD
  Repeat
    Find gt FPCAD by INDEX.2
    [ Found] Begin
      Print FPCAD.CADASTRO to Body.1
      Print FPCAD.DIGITO   to Body.2
      Print FPCAD.NOME     to Body.3
      Print (Total.1 + 1) to Total.1
      Output Body
      If Linecount gt 60 Gosub Imp-Header
    End
  Until [~ Found]
  Output Total
  Formfeed
  Outclose
  Accept Pronta.1 { Capslock Check="012" }
  If Pronta.1 eq "1" Runprogram Wait "lp -dpes /us0/fp/s/fpcadr01.spl"
  If Pronta.1 eq "2" Runprogram Wait "lp -dpes2 /us0/fp/s/fpcadr01.spl"
  System

#include /usr/dataflex/impheader.inc

```

Na parte da definição das imagens (parte superior ao /*), as imagens (conjunto de caracteres que podem ser uma tela, um cabeçalho ou linha de relatório, e podem ser mostradas na tela ou arquivo) definem-se da seguinte maneira: uma barra, e um nome sem espaços, e o que segue no Quadro 5, é o corpo da imagem, até aparecer uma nova barra [OLI96].

Conforme [OLI96], na definição das telas, existem as janelas. Janelas são espaços nos quais o programador pode solicitar que seja mostrada uma determinada expressão (como por exemplo uma sequência de caracteres ou o resultado de um cálculo). Um exemplo do que são janelas pode-se ver no Quadro 5, em sombreado.

Quadro 5: Exemplo de uma imagem num programa fonte Dataflex

```

/Header
+-----+
PMJS/DPD                Relatório de Funcionários                /_/ /_ :
Cadastro Nome                Pagina...: .
+-----+

```

Existe nesta imagem, quatro janelas. A primeira "_/_/___" é uma janela do tipo data. A seguir aparece "_:__", que são mais duas janelas do tipo string, separadas pelo caracter ":". Por último aparece uma janela do tipo numérica "___.". Janelas do tipo data, sempre serão "_/_/___" ou "_/_/__". Janelas string sempre serão uma sequência de underscores sem barras ou pontos. E janelas numéricas, sempre possuirão um ponto. Este pode ser no final (no caso de números inteiros) ou no meio (no caso de números com casas decimais).

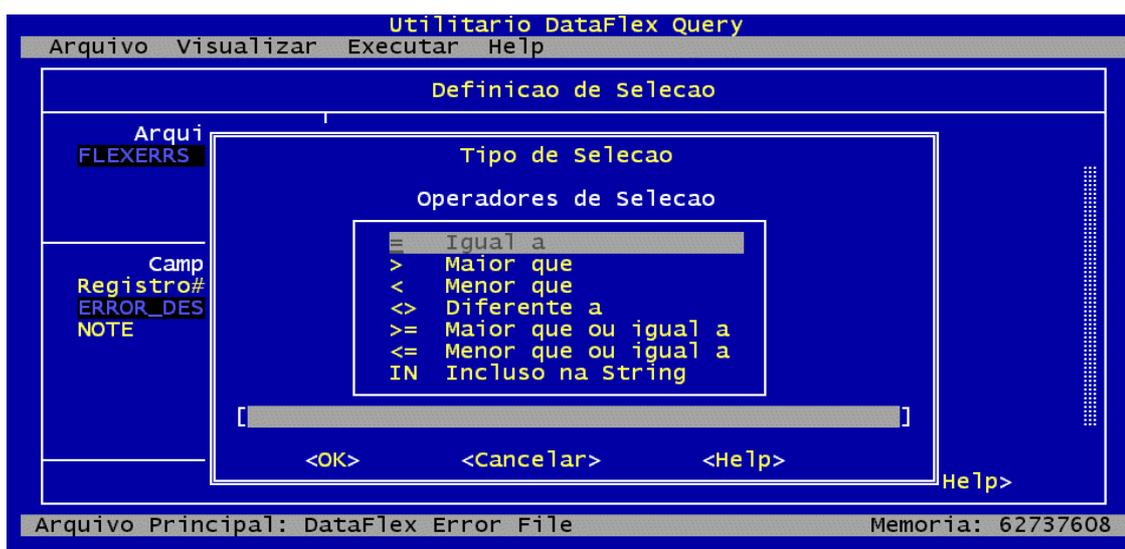
Finalmente, na parte inferior, vem o código de programação propriamente dito. Como se pode notar no Quadro 4, a linguagem Dataflex possui muitas diferenças a uma linguagem como Pascal. Por exemplo, no Dataflex, em um comando if, não aparecem caracteres como "=", ">", "<", "<>", ">=" e "<=". Eles são respectivamente substituídos por "eq", "gt", "lt", "ne", "ge" e "le". As variáveis lógicas são substituídas por Indicadores, que dependendo do caso podem ou não aparecer entre colchetes. E a principal mudança em relação a linguagens como Pascal e C++, é a pré-definição de telas (imagens).

4.2 UTILITÁRIOS

Existem na linguagem Dataflex, alguns utilitários. Aqui serão descritos apenas os principais: DFFILE, DFINDEX, DFQUERY e DFMAINT.

- a) DFFILE: sua função principal é a criação de arquivos de dados e índices;
- b) DFINDEX: utilitário para reindexação de arquivos;
- c) DFMAINT: manutenção do FILELIST e DFCLEAN (exclusão de registros duplicados);
- d) DFQUERY: gerador de consultas e relatórios (uma tela de exemplo deste programa é mostrada na Figura 20).

Figura 20: Tela do utilitário DFQUERY



No momento em que é solicitado a criação de um arquivo, o DFFILE busca automaticamente, o primeiro número disponível no FILELIST. O FILELIST (FILELIST.CFG), é um arquivo onde estão cadastrados os arquivos que serão utilizados naquele diretório base da aplicação. Cada um dos arquivos então cadastrados possuem um determinado número. O número máximo de arquivos por FILELIST é de 255 arquivos.

Depois de saber qual o número do FILELIST, é necessário informar o nome base do arquivo (nome com o qual ele será gravado no disco), e depois o nome Dataflex (nome que é

utilizado dentro dos programas fonte Dataflex), nome para o usuário (descrição do arquivo que aparece em utilitários como DFQUERY).

Após isto é realizado então a definição de campos, onde é informado o nome do campo, tipo (numérico, ascii, data) e se este campo está relacionado com algum outro existente em outro arquivo. Posteriormente são criados os índices. Um arquivo Dataflex pode possuir no máximo 15 índices. Existem dois diferentes tipos de índices: Online e Batch. Os índices podem ser reindexados a qualquer momento utilizando-se o utilitário DFINDEX. Após terem sido executados todos estes passos, o arquivo de dados Dataflex estará pronto para ser usado.

O utilitário DFMAINT é um utilitário mais usado para manutenção do FILELIST. Com ele é possível acessar as opções de indexação, limpar base de dados, gerar arquivo FD (File Definition), DEF (Descrição Detalhada do Arquivo) e DFCLEAN (Manutenção de Registros Duplicados), entre outras. Exemplos de arquivos “.FD” e “.DEF” são mostrados nos quadros 6 e 7.

Quadro 6: Exemplo de arquivo “.FD” do Dataflex.

```
#REPLACE FILE1 FPCAD
#REPLACE FPCAD.RECNUM |FN1,0
#REPLACE FPCAD.CADASTRO |FN1,1
#REPLACE FPCAD.NOME |FS1,2
#REPLACE FPCAD.SECRET |FN1,3
#REPLACE FPCAD.DEPTO |FN1,4
#REPLACE FPCAD.VINCULO |FS1,5
```

Quadro 7: Exemplo de arquivo “.DEF” do Dataflex.

```

-----
DATE: 11/24/99          TIME: 09:27                      PAGE: 1
FILE DEFINITION FOR FILE: FPCAD (# 1)
-----
FILE ROOT NAME          : FPCAD
USER DISPLAY NAME       : FPCAD Cadastro de Funcionarios
DATAFLEX FILE NAME      : FPCAD
-----
RECORD LENGTH           : 512          ( USED: 47 )
MAX NUMBER OF RECORDS   : 1000        ( USED: 0 )
FILE COMPRESSION        : NONE
RE-USE DELETED SPACE    : YES
LOCKING TYPE            : FILE
HEADER INTEGRITY CHECKING : YES
TRANSACTION TYPE        : NONE
-----
NUM  FIELD NAME          TYPE  SIZE  OFFST  IX  RELATES TO FILE.FIELD
-----
  1  CADASTRO             NUM   6.0   1    1
  2  NOME                  ASC   40    4    2
  3  SECRET               NUM   2.0   44   3    FPSEC.CAMPO_1 (2,1)
  4  DEPTO                NUM   4.0   45   3    FPDEP.CAMPO_1 (3,1)
  5  VINCULO              ASC    1    47   3    FPVIN.CAMPO_1 (4,1)
-----
INDEX#  FIELDS           DES  U/C    LENGTH  LEVELS  SEGMENTS  MODE
-----
  1     CADASTRO         NO   NO      3        2        1     ON-LINE
  2     NOME             NO   NO     43        3        2     ON-LINE
        RECNUM          NO   NO
  3     SECRET          NO   NO     46        3        4     ON-LINE
        DEPTO           NO   NO
        NOME            NO   NO
        RECNUM          NO   NO

```

Por último vem o utilitário DFQUERY, que é um gerador de consultas e relatórios. Com ele é possível realizar uma consulta rápida, sem que seja necessário criar um programa para isto. Se necessário, o DFQUERY pode automaticamente, gravar o programa fonte do relatório em disco.

5 ESPECIFICAÇÃO DO PROTÓTIPO

Neste capítulo, está descrita a especificação do protótipo, que inclui: a lista de eventos, as descrições de cada um dos eventos, o diagrama de contexto e os diagramas de fluxo de dados.

O protótipo deste sistema tem como finalidade principal, a edição de programas fonte da linguagem Dataflex. Baseado na última palavra digitada e em um arquivo de definição de sintaxe, o sistema aplica cores conforme o tipo de palavra digitada. Se for um comando, cor azul. Se for um operador, cor vermelha. Se for uma variável pré-definida do Dataflex, cor verde. Textos entre aspas e entre colchetes aparecerão em cor púrpura e finalmente os comentários aparecerão em cor azul e em itálico.

Baseando-se também na última palavra digitada e em um arquivo de gabaritos, o sistema pode inserir um gabarito para o comando digitado, se ele estiver neste arquivo de gabaritos. Automaticamente após a inserção do gabarito, ocorrerá uma análise do texto inserido pelo sistema com relação à verificação das cores.

Pode-se ativar ou não o modo FTP. Caso este esteja ligado, será estabelecida uma conexão com o servidor FTP baseada em informações contidas na janela de Configurações. O usuário trabalha com arquivos do servidor FTP, como se estivesse trabalhando na sua própria máquina.

Existirão ainda algumas utilidades no protótipo deste editor como: selecionar tudo, recortar, copiar, colar, localizar e substituir.

A Ferramenta CASE utilizada para a especificação do protótipo é o Power Designer - Process Analyst, disponível na FURB - Universidade Regional de Blumenau. Antes dela foram tentadas outras ferramentas, não somente ferramentas CASE como ErWin, DFDraw, FlowCharting, SmartDraw, porém, a mais fácil de usar e que permite detalhamento de processos sem a necessidade de criar um novo arquivo e que efetua algumas consistências, foi o Power Designer - Process Analyst.

Para a implementação, foi utilizado o ambiente de programação Delphi, versão 4, disponível na Prefeitura Municipal de Jaraguá do Sul. Foram feitos testes anteriores com a versão 3, porém o componente FTP para transferência de arquivos, não funcionou adequadamente. O componente FTP da versão 4 é também muito mais fácil de usar. O hardware usado foi:

- a) um computador RISC (como servidor FTP);
- b) um computador Pentium II (para desenvolver o software);
- c) rede TCP/IP (interligando os computadores).

5.1 LISTA DE EVENTOS

Neste protótipo foram identificados os seguintes eventos:

- a) Desenvolvedor Informa Sintaxe
- b) Desenvolvedor Informa Gabaritos
- c) Usuário Edita Programa
- d) Usuário Abre Programa
- e) Servidor Transfere Programa
- f) Sistema Informa Arquivos Abertos
- g) Sistema Salva Programa
- h) Sistema Transfere Programa
- i) Usuário Configura Sistema
- j) Sistema Informa Definições de Arquivos

5.2 DESCRIÇÃO DOS EVENTOS

A seguir, serão apresentadas as descrições detalhadas dos eventos.

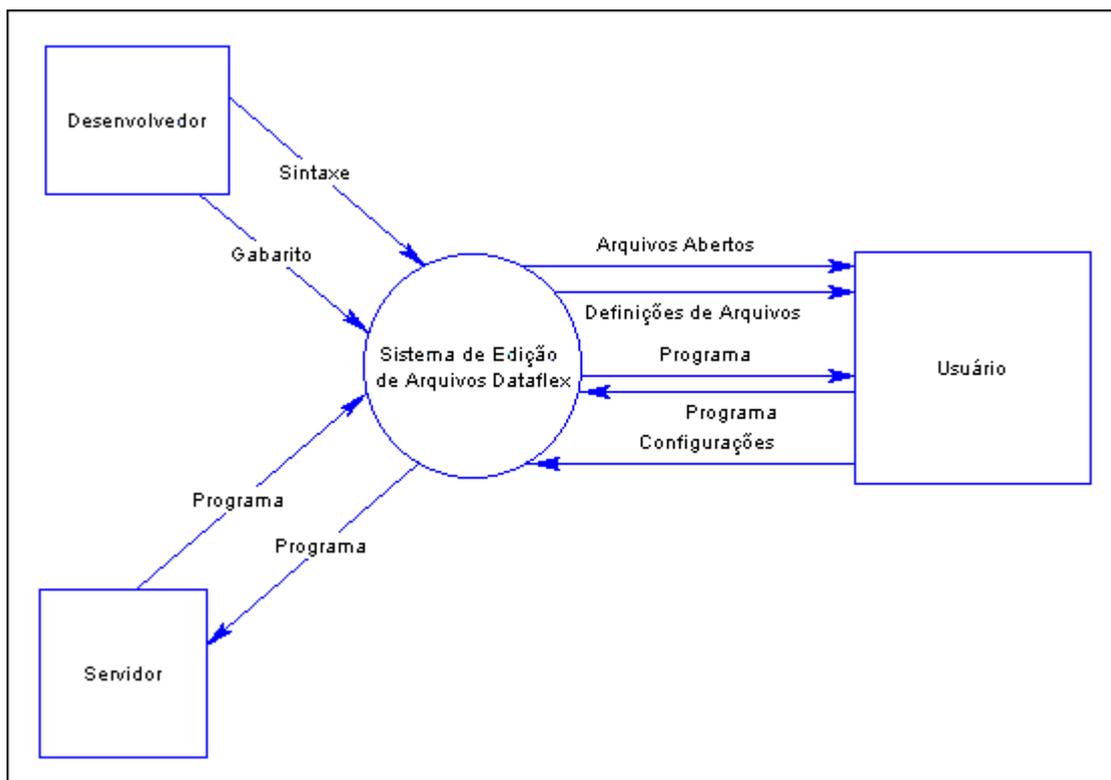
- a) Desenvolvedor Informa Sintaxe: é um evento que acontece uma única vez no sistema. O Desenvolvedor cria um arquivo de definição de sintaxe com os nomes dos comandos, operadores, variáveis pré-definidas.
- b) Desenvolvedor Informa Gabaritos: assim como o anterior, é um evento que ocorre uma única vez no sistema. O Desenvolvedor cria um arquivo com os gabaritos a serem inseridos para os comandos que achar mais conveniente (mais utilizados e extensos).

- c) Usuário Edita Programa: neste evento, o usuário faz a edição propriamente dita do programa fonte. Aqui existe um sub-evento muito importante, que é o pressionamento de uma tecla. A cada tecla pressionada, o sistema deve analisar a palavra anterior e a posterior e alterar (se necessário) as cores dos comandos baseado em um arquivo de definição de sintaxe previamente carregado;
- d) Usuário Abre Programa: aqui, o usuário abre um programa fonte de um dispositivo local de armazenamento;
- e) Servidor Transfere Programa: se o usuário optar por abrir um arquivo de um servidor FTP (no caso da Prefeitura este modo será o mais utilizado, porque os programas a serem editados encontram-se no servidor HP), será realizada uma conexão do tipo FTP(TCP/IP) com o servidor. Depois de estabelecida a conexão, o sistema faz uma requisição ao servidor e este transfere o arquivo para a máquina local;
- f) Sistema Informa Arquivos Abertos: aqui, o sistema analisa o arquivo procurando pelo comando OPEN, que no Dataflex, é o comando de abertura de arquivo de dados. A cada comando OPEN encontrado, será adicionado um item com o nome do arquivo aberto em uma janela, para o usuário saber quais arquivos serão abertos pelo programa em edição;
- g) Sistema Salva Programa: o usuário, muito provavelmente salvará as alterações que realizou em um programa, e o arquivo será salvo num dispositivo de armazenamento local.
- h) Sistema Transfere Arquivo: se o arquivo for proveniente de um dispositivo de armazenamento remoto, o sistema deverá transferir o programa editado de volta para o servidor FTP.
- i) Usuário Configura Sistema: o usuário pode configurar o sistema quanto às seguintes características: localização e nome dos arquivos de definição de sintaxe e de gabaritos, informações do servidor FTP (nome do host(IP), porta, usuário e senha), e fonte padrão para os programas fonte abertos.
- j) Sistema Informa Definições de Arquivos: se o usuário quiser, ele pode visualizar as Definições de Arquivos. Este arquivo contém o nome, tipo e tamanho dos campos, bem como informações sobre índices.

5.3 DIAGRAMA DE CONTEXTO

Na Figura 21 está representado o Diagrama de Contexto do Protótipo.

Figura 21: Diagrama de Contexto do Sistema



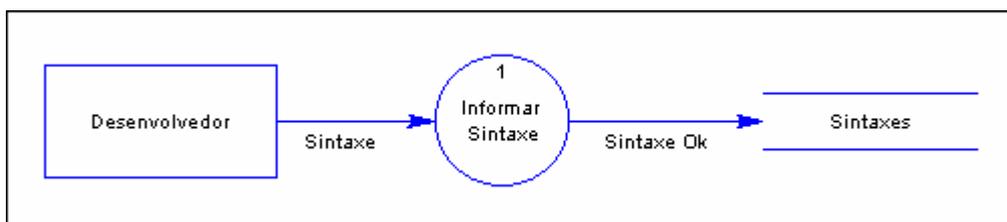
5.4 DIAGRAMA DE FLUXO DE DADOS PARTICIONADO

A seguir serão apresentados os diagramas de fluxo de dados (DFD) de cada um dos eventos citados anteriormente. Descrições mais detalhadas também podem ser encontradas abaixo das figuras.

5.4.1 DESENVOLVEDOR INFORMA SINTAXE

Na figura 22, é possível observar o detalhamento do processo número 1 (desenvolvedor informa sintaxe).

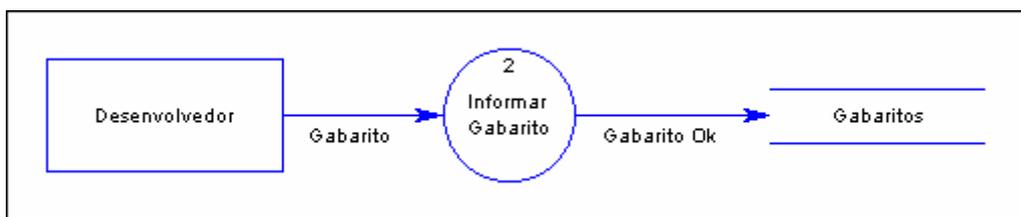
Figura 22: Detalhamento de Processo - Desenvolvedor Informa Sintaxe



5.4.2 DESENVOLVEDOR INFORMA GABARITOS

Na figura 23, é possível observar o detalhamento do processo número 2 (desenvolvedor informa gabaritos).

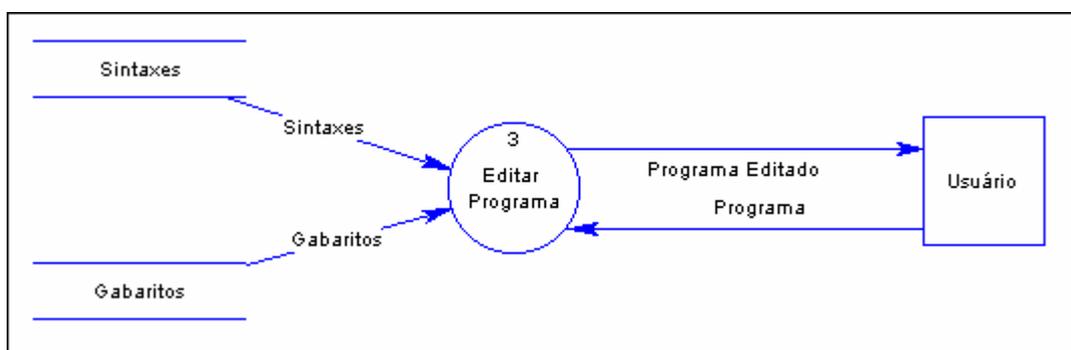
Figura 23: Detalhamento de Processo - Desenvolvedor Informa Gabaritos



5.4.3 USUÁRIO EDITA PROGRAMA

Na figura 24, é possível observar o detalhamento do processo número 3 (usuário edita programa).

Figura 24: Detalhamento de Processo - Usuário Edita Programa



Aqui acontece a edição do programa fonte propriamente dito. Na verdade, o sistema verifica os seguintes caracteres digitados: espaço (ASCII 32), enter (ASCII 13), backspace (ASCII 8), aspas duplas (ASCII 34), chaves (ASCII 91 e 93) e a barra (ASCII 47).

Quando um espaço ou a tecla enter é pressionada, é então analisada a palavra anterior e posterior a este espaço. Se estas palavras estiverem contidas no arquivo de definição de sintaxe, então elas recebem cores: azul para comandos, vermelho para operadores e verde para variáveis pré definidas. Se houver aspas ou chaves nesta palavra, a cor aplicada será a cor púrpura e se houver “//”, que é o indicador de comentários da linguagem Dataflex, a cor aplicada é azul e o estilo itálico.

Quando um caracter backspace é pressionado, é realizado processo semelhante ao anterior, porém, considerando que o caracter anterior ao cursor já fora excluído.

Se o caracter aspas duplas for pressionado, é armazenado em uma variável a posição desta aspa dupla e um flag é ligado dizendo ao sistema que o que está sendo digitado é uma

string. Quando a segunda aspa dupla for pressionada, será colorido com a cor púrpura o texto que compreende a primeira aspa e a segunda aspa. O mesmo acontece com os colchetes.

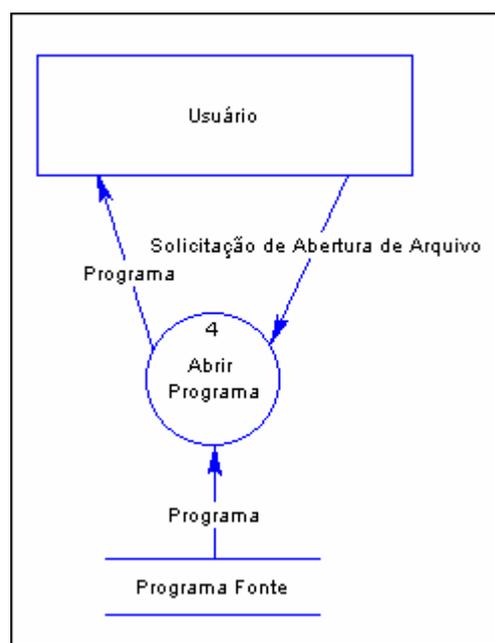
No caso do comentário, quando a primeira barra é pressionada, é armazenado em uma variável, a posição desta barra. Quando a segunda barra for pressionada, é verificado se a posição desta barra é a primeira (que foi armazenada) mais um. Se isto for verdade, o texto que for sendo digitado em seguida a estas duas barras recebe a cor azul e com estilo itálico.

O usuário pode ainda, utilizar opções de busca e substituição para localizar e/ou substituir determinados textos dentro do programa fonte.

5.4.4 USUÁRIO ABRE PROGRAMA

Na figura 25, é possível observar o detalhamento do processo número 4 (usuário abre programa).

Figura 25: Detalhamento de Processo - Usuário Abre Programa



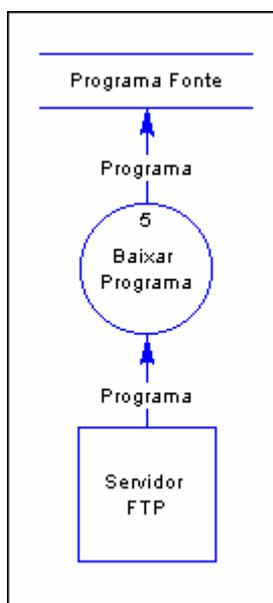
Conforme descrito anteriormente, este processo fará a abertura deste arquivo a partir do dispositivo de armazenamento local. Se o usuário optar por abrir um arquivo de um dispositivo de armazenamento remoto, a transferência do programa fonte do servidor FTP é necessária ser realizada antes.

Após a abertura deste programa fonte, acontece uma varredura no mesmo, a fim de colorir os comandos, operadores e variáveis pré-definidas, baseando-se no arquivo de definição de sintaxe.

5.4.5 SERVIDOR TRANSFERE PROGRAMA

Na figura 26, é possível observar o detalhamento do processo número 5 (servidor transfere programa).

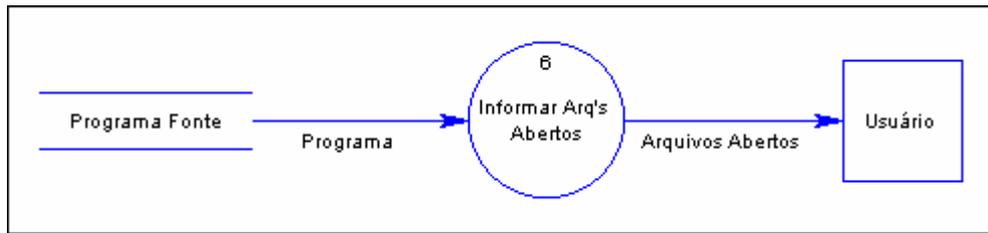
Figura 26: Detalhamento de Processo - Servidor Transfere Programa



5.4.6 SISTEMA INFORMA ARQUIVOS ABERTOS

Na figura 27, é possível observar o detalhamento do processo número 6 (sistema informa arquivos abertos).

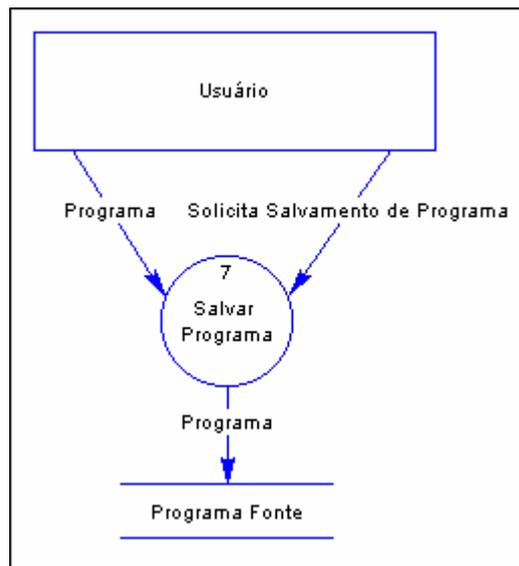
Figura 27: Detalhamento de Processo - Sistema Informa Arquivos Abertos



5.4.7 SISTEMA SALVA PROGRAMA

Na figura 28, é possível observar o detalhamento do processo número 7 (sistema salva programa).

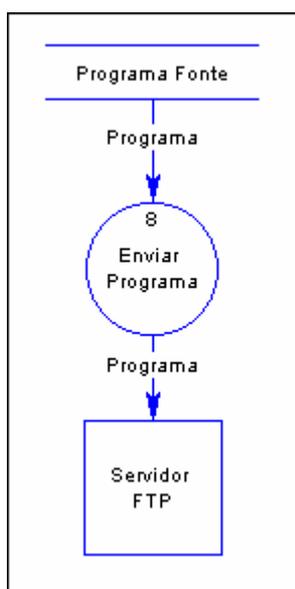
Figura 28: Detalhamento de Processo - Sistema Salva Programa



5.4.8 SISTEMA TRANSFERE PROGRAMA

Na figura 29, é possível observar o detalhamento do processo número 8 (sistema transfere programa).

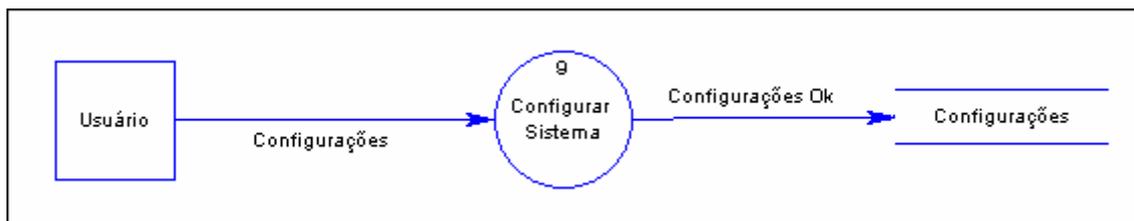
Figura 29: Detalhamento de Processo - Sistema Transfere Programa



5.4.9 USUÁRIO CONFIGURA SISTEMA

Na figura 30, é possível observar o detalhamento do processo número 6 (usuário configura sistema).

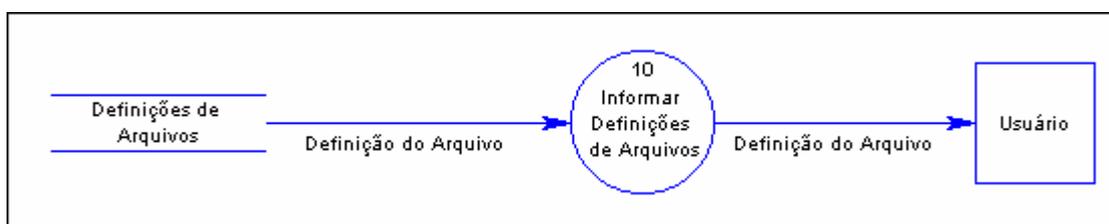
Figura 30: Detalhamento de Processo - Usuário Configura Sistema



5.4.10 SISTEMA INFORMA DEFINIÇÕES DE ARQUIVOS

Na figura 31, é possível observar o detalhamento do processo número 7 (sistema informa definições de arquivos).

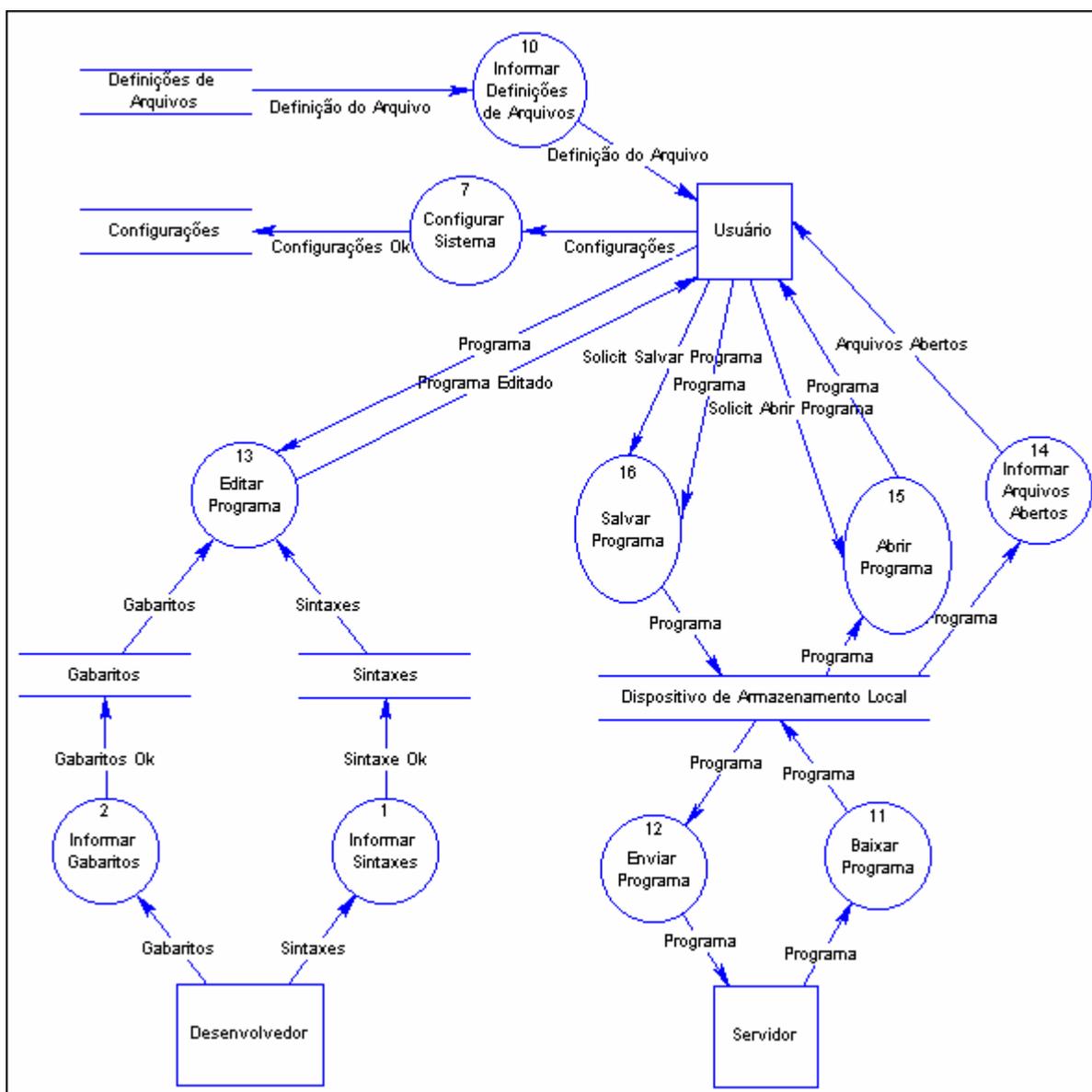
Figura 31: Detalhamento de Processo - Sistema Informa Definições de Arquivos



5.4.11 DFD - SISTEMA COMPLETO

Na figura 32, é possível observar o diagrama de fluxo de dados completo do sistema de edição orientada por gabaritos(sintaxe). Aqui, cada processo, refere-se a um evento do sistema (dfd particionado por evento).

Figura 32: Diagrama de Fluxo de Dados completo do Sistema



5.5 DICIONÁRIO DE DADOS (DD)

O Dicionário de Dados(DD) é quem fornece a informação de texto de suporte para auxiliar a informação gráfica mostrada no DFD. O DD é um grupo organizado de definições de todos os elementos de dados que estão envolvidos no sistema que está sendo modelado.

A seguir serão detalhados todos os fluxos e depósitos de dados que estão apresentados nos DFD's anteriores.

5.5.1 FLUXOS DE DADOS

Sintaxe:

definição da sintaxe;

Gabarito:

definição dos gabaritos;

Configurações:

quantidade de espaços da indentação : numérico;

nome da fonte padrão : texto;

tamanho da fonte padrão : numérico;

nome completo do arquivos de definição de sintaxe : texto;

nome completo do arquivos de definição de gabarito : texto;

informações do servidor FTP : host, usuário, senha : texto; porta : numérico;

proxy : lógico; host_proxy : texto;

caminho dos arquivos DEF locais : texto;

caminho dos arquivos DEF remoto : texto;

diretório temporário : texto;

diretório de inicio para abertura de arquivos : texto;

cores dos comandos, operadores, variáveis, pal.chave 1, pal.chave 2, pal.chave 3, comentário, indicador e strings : texto;

nome completo dos utilitários dataflex (dffile, dfmaint, dfquery, dfcomp, dfrun) : texto;

Arquivos Abertos:

lista com os nomes de arquivos que aparecem no programa fonte após “OPEN” :
texto;

Definição do Arquivo:

texto contendo nome, tipo e tamanho dos campos de um arquivo de dados
Dataflex bem como informações sobre índices e relacionamentos : texto;

Programa:

texto do programa fonte : texto;

5.5.2 DEPÓSITOS DE DADOS

Sintaxe:

arquivo chamado dataflex.sdf, que contém a definição da sintaxe;

Gabaritos:

arquivo chamado dataflex.gab, que contém a definição dos gabaritos;

Programa Fonte:

arquivo de programa fonte sendo editado;

Configurações:

arquivo no disco chamado flexed.ini, com as configurações;

Definições de Arquivos:

arquivos texto com as definições dos arquivos de dados do Dataflex;

No canto direito inferior, existe uma pequena janela, que informa quais são os arquivos de dados abertos pelo programa em edição.

Na parte superior da Tela Principal (figura 33), está uma barra de ferramentas (figura 34), que será agora explicada.

Figura 34: Barra de Ferramentas



Na primeira barra de ferramentas podem ser encontrados os seguintes botões:

- 1) **Modo FTP:** liga ou desliga o modo FTP do editor. Se este estiver ligado, todas as operações com arquivos serão realizadas via FTP (Servidor de Arquivos Remoto);
- 2) **Conectar:** conecta-se ao servidor FTP baseado em configurações estabelecidas na tela de configurações (ver explicação logo abaixo no botão de configurações);
- 3) **Desconectar:** desconecta-se do servidor FTP;
- 4) **Novo Arquivo:** se um arquivo está sendo editado, pede para este arquivo ser salvo. Logo após, limpa a janela de edição para um novo arquivo ser editado;
- 5) **Abrir Arquivo:** abre um arquivo de um dispositivo de armazenamento. Se o modo FTP estiver ligado, abre o arquivo de um servidor FTP, senão abre o arquivo de um dispositivo local de armazenamento;
- 6) **Salvar Arquivo:** salvar um arquivo em um dispositivo de armazenamento. Como o anterior, se o modo FTP estiver ligado, salva o arquivo em um servidor FTP, senão salva o arquivo em um dispositivo local de armazenamento;
- 7) **Salvar Como:** idem ao anterior, porém, sempre é solicitado um nome para o arquivo;
- 8) **Fechar Arquivo:** executa a mesma rotina que o quarto botão novo;
- 9) **Recortar:** Recorta o texto selecionado para a área de transferência;
- 10) **Copiar:** Copia o texto selecionado para a área de transferência;

- 11) **Colar:** Cola o texto que está na área de transferência para o texto que está sendo editado, na posição do cursor;
- 12) **Analisa:** Se houver alguma disparidade de cores no sistema, este botão chama uma rotina (que é a mesma que é executada quando o arquivo é aberto que aplica as cores corretas ao programa em edição).
- 13) **Diminuir Identação:** Move o texto para a esquerda em n caracteres. O valor de n é definido pelo usuário na tela de configurações;
- 14) **Aumentar Identação:** Idem ao anterior, porém move o texto para a direita;
- 15) **Maiúsculas:** Formata o texto selecionado para letras Maiúsculas;
- 16) **Minúsculas:** Formata o texto selecionado para letras Minúsculas;
- 17) **Desfazer:** Desfaz a última ação de edição;
- 18) **Configurações:** Opções de configuração do editor como: Servidor FTP (nome do host/endereço IP, nome e senha do usuário, Fonte utilizada no editor, Local do Arquivo de Definição de Sintaxe, etc...)
- 19) **Mostrar Arquivos Abertos:** Exibe / Esconde na janela de edição, um painel com o nome dos arquivos abertos;
- 20) **Localizar:** Procura por um conjunto de strings no programa em edição.
- 21) **Localizar e Substituir:** Procura por um conjunto de strings no programa em edição e efetua substituições se necessário.
- 22) **Janela Anterior:** Se houver mais de uma janela aberta, pula para a janela anterior;
- 23) **Janela Posterior:** Se houver mais de uma janela aberta, pula para a janela posterior;
- 24) **Janelas em Cascata:** Se houver mais de uma janela aberta, coloca-as em cascata;
- 25) **Janelas Lado a Lado Horizontalmente:** Se houver mais de uma janela aberta, coloca-as lado a lado horizontalmente;
- 26) **Janelas Lado a Lado Verticalmente:** Se houver mais de uma janela aberta, coloca-as lado a lado verticalmente;
- 27) **Sobre:** Exibe uma caixa com informações sobre o FlexEd.

Na segunda barra de ferramentas podem ser encontrados os seguintes botões:

- 1) **DFComp:** Se um compilador Dataflex for configurado, este é ativado para compilar o programa atual;
- 2) **DFRun:** Se um compilador Runtime do Dataflex for configurado, este é ativado para rodar o programa atual;
- 3) **DFQuery:** Chama o utilitário para pesquisa no banco de dados, se este estiver configurado;
- 4) **DFFile:** Chama o utilitário para realização de alterações na estrutura do banco de dados, se este estiver configurado;
- 5) **DFMaint:** Chama o utilitário para manutenção do banco de dados, se este estiver configurado.

Na parte inferior da tela do protótipo do editor, existe uma barra de status, que informa a posição do cursor no texto (linha, coluna), o status do texto, se foi modificado e o status do Servidor FTP (conectado, desconectado).

Existem várias opções de configuração no FlexEd. Estas opções podem ser vistas nas próximas figuras.

Figura 35: Configurações de Edição / Exibição

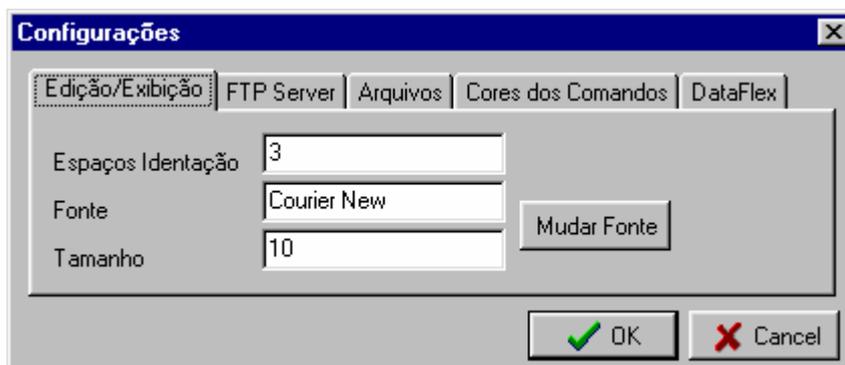


Figura 36: Configurações do Servidor FTP

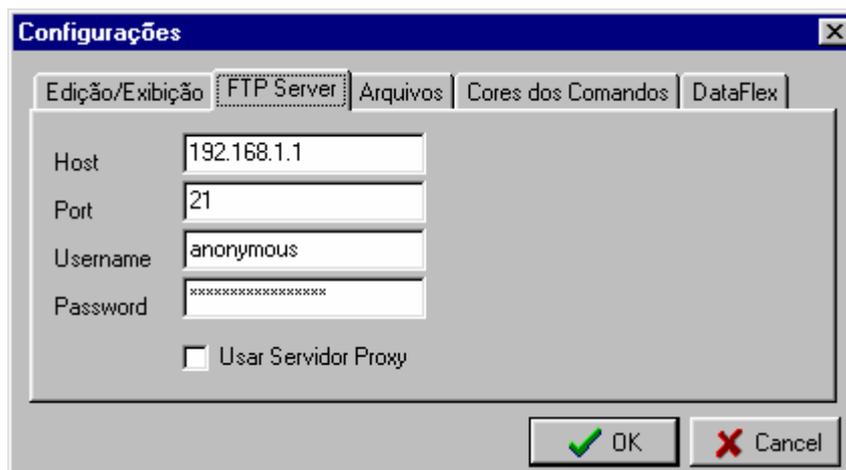


Figura 37: Configurações de Arquivos

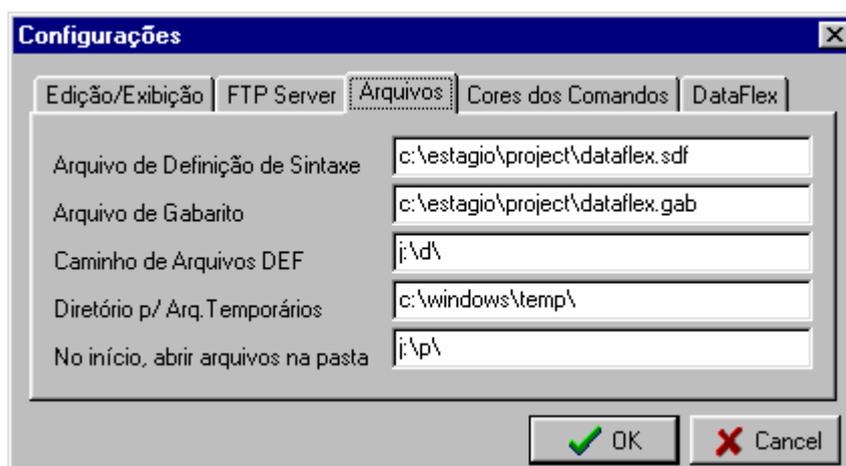


Figura 38: Configurações das Cores de Comandos

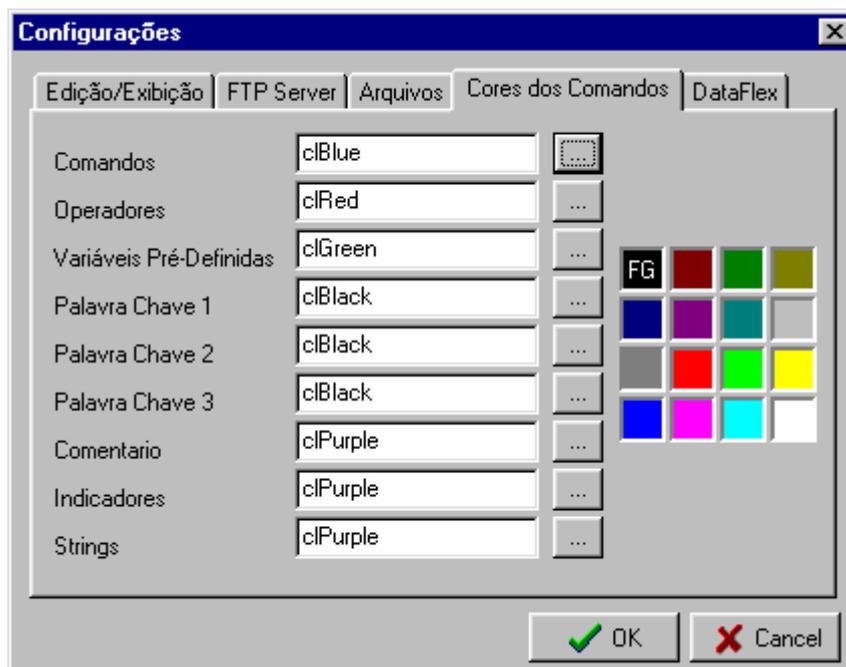
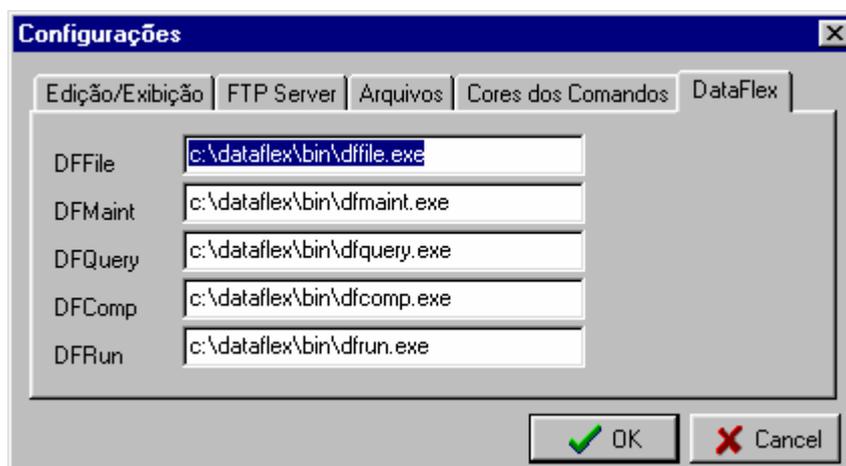


Figura 39: Configurações da linguagem Dataflex



6.1 ARQUIVO DE DEFINIÇÃO DE SINTAXE

O arquivo de definição de sintaxe, é o arquivo no qual se baseia a mudança de cor dos comandos.

No quadro 8, é apresentado a lista dos comandos da linguagem Dataflex, que serão incorporados no protótipo deste editor. No quadro 9 será apresentada a lista de operadores. No quadro 10 a lista de variáveis e indicadores pré-definidas e no quadro 11, o formato do arquivo de definição de sintaxe (não será apresentado o arquivo completo pois é muito extenso).

Quadro 8: Comandos da Linguagem Dataflex

ABORT	ACCEPT	APPEND	ASCII
ATTACH	AUTOPAGE	BACKFIELD	BEGIN
BLANKFORM	BREAK	CALC	CASE
CASEELSE	CASEEND	CGC_GCHECK	CHAIN
CHANGE	CHARACTER	CIC_CHECK	CLEAR
CLEARFORM	CLEARSCREEN	CLEARWARNING	CLEARXY
CLOSE	CLOSE_INPUT	CMDLINE	COMPARING
CONDITIONAL	COPYFILE	COUNT	DATE
DEBUG	DELETE	DESPool	DIRECT_INPUT
DIRECTORY	DISPLAY	DOWN	ELSE
END	ENDGROUP	ENTAGAIN	ENTDISPLAY
ENTER	ENTEREND	ENTERGROUP	ENTERMODE
ENTRY	ERASEFILE	ERROR	ERROR
FIELD_DEF	FILE_MODE	FILE_SIZE	FILELIST
FIND	FINDKEY	FOR	FORMAT
FORMFEED	GOSUB	GOTO	GOTOXY
HELP	INCREMENT	INDEX	INDEX-DEF
INDICATE	INDICATOR	INKEY	INKEY\$
INPUT	JOIN	JOINEND	JOINMOVE
KEYCHECK	KEYPROC	LEFT	LENGTH
LOCK	LOOP	MAKE_FILE	MEMORY
MID	MOD	MOVE	MOVEDATE
MOVEINT	MOVENUM	MOVESTR	NAME
NEXT	NUMBER	OPEN	OUTCLOSE
OUTFILE	OUTPUT	PAD	PAGE
PAGECHECK	PATHNAME	PAUSE	POP
POS	PRINT	READ	READ-BLOCK
READLN	REAL	RECNUM	REGISTRATION
RELATE	REMOVE	RENAMEFILE	REPEAT
REPLACE	REPLACEALL	REPORT	REPORTEND
RERead	RETURN	RIGHT	RUNPROGRAM
SAVE	SAVERECORD	SCREENMODE	SEARCHKEY
SET	SHOW	SHOWLN	STRING
SUBTOTAL	SUM	SYSDATE	SYSTEM
SYSTEM	TRIM	UNLOCK	UNTIL
UPPERCASE	USING	WAIT	WHILE
WRITE	WRITELN	ZEROFIELD	

Quadro 9: Operadores da Linguagem Dataflex

BY	EQ	GE	GT
IF	IN	LE	LT
MATCH	OFF	ON	THRU
TO	WITH		

Quadro 10: Variáveis e Indicadores Pré-Definidos da Linguagem Dataflex

CURRENT_IMAGE	CURRENT_WINDOW	ERRLINE	FIELDINDEX
FILENUMBER	FLEXKEY	LASTERR	LINECOUNT
NEXT_MENU	NUMPAGE	NUMWINDOW	PAGECOUNT
PAGEEND	PAGEFEED	RECCOUNT	SCREENMODE
STRLEN	STRMARK	WINDOWINDEX	

Quadro 11: Formato do Arquivo de Definição de Sintaxe

#CABORT
#CACCEPT
#CAPPEND
#CASCII
#CATTACH
#CAUTOPAGE
#CBEGIN
#CBLANKFORM
#CBREAK
#CCALC
#CCASE
#CCASEELSE
#CCASEEND
#CCHAIN
#CCHANGE
#CCLEAR
#CCLEARFORM
#CCLEARSCREEN
#OBY
#OEQ
#OGE
#OGT
#OIF
#OIN
#OLE
#OLT
#VCURRENT_IMAGE
#VCURRENT_WINDOW
#VFIELDINDEX
#VFILENUMBER
#VLINECOUNT
#VPAGECOUNT
#VRECCOUNT
#VSCREENMODE
#VSTRLEN
#VSTRMARK
#VWINDOWINDEX

No quadro 11, é possível observar no início de cada uma das linhas, o sinal “#” e uma letra. Esta letra indica em que classe de palavras chave ela se encontra (v = variáveis pré-definidas, c = comandos e o = operadores).

No momento da inicialização do software, mais precisamente na criação do formulário de configuração, é checada a existência do arquivo `c:\windows\flexed.ini`. Neste arquivo, estão gravados os nomes dos arquivos de definição de sintaxe e arquivo de gabaritos. Se o arquivo não existir então é criado um novo arquivo.

Após isto, baseado neste mesmo arquivo `flexed.ini`, o arquivo de definição de sintaxe é carregado para três listas de strings diferentes, uma para comandos, uma para operadores e outra para variáveis pré-definidas. Os nomes destas listas de strings são respectivamente Comandos, Operadores e Variáveis.

O usuário pode optar por programar em outra linguagem, por exemplo PASCAL, JAVA ou C++. Neste caso, é necessário criar um novo arquivo com os comandos, operadores e outros ítems das outras linguagem e alterar as configurações para o arquivo de definição de sintaxe.

6.2ARQUIVO DE GABARITOS

O arquivo de gabaritos, é o arquivo no qual a inserção de gabaritos se baseia. Quando a barra de espaços é pressionada (ASCII 32), é verificada se a palavra anterior à barra de espaços existe no arquivo de gabaritos.

No quadro 12 é mostrado um exemplo do arquivo de gabaritos para a linguagem Dataflex.

Quadro 12: Exemplo de Arquivo de Gabarito

```
#IF
<condicao> Begin
    <comandos>
End
#REPEAT
    Find gt <arquivo> by INDEX.<n>
    Indicate ACHOU<x> as [Found]
    [ Found] Begin
        <comandos>
    End
Until [~ Achou<x>]
#FOR
<cont> from <ini> to <fim>
    <comandos>
Loop
#PRINT
<expressão> to <Destino>
#CALC
<expressão> to <Destino>
#MOVE
<expressão> to <Destino>
```

Na primeira linha deste arquivo aparece #IF que é o início do gabarito do comando IF. A linha abaixo da palavra “IF”, é colocada ao lado do comando digitado. As linhas restantes, até ser encontrado o caracter “#” ou o final do arquivo, são inseridas abaixo da linha do comando “IF”.

Este arquivo pode ser modificado ao gosto do usuário, dependendo do seu estilo pessoal de programação ou do padrão adotado pela empresa. Por exemplo, para o comando REPEAT, o programador pode querer inserir outros comandos e ainda, pode querer inserir comandos novos. É necessário ressaltar o cuidado para manter o arquivo no layout correto:

- a) antes do nome do comando a ser “gabaritado”, deve aparecer um sinal de “#”;
- b) o nome do comando deve estar em letras maiúsculas;
- c) as linhas a serem inseridas devem ser escritas abaixo da linha com o nome do comando;
- d) as linhas serão inseridas no texto editado, até que se encontre um caracter “#” ou o final do arquivo.

7 CONCLUSÃO

O protótipo foi instalado nos computadores dos programadores da Prefeitura Municipal de Jaraguá do Sul e atingiu o objetivo principal, que foi criar um ambiente de programação mais fácil de ser utilizado e mais produtivo para os programadores.

Os programadores acharam o ambiente visual produtivo, principalmente pelo fato de colorir diferentemente os comandos, operadores, indicadores, comentários, variáveis pré-definidas e strings. Sendo assim, fica inclusive mais fácil, a detecção pelo programador de erros de sintaxe. A característica de inserção de gabaritos foi aprovada por todos por reduzir o tempo de programação.

Quanto às ferramentas utilizadas, na parte de especificação foi usado o Power Designer. Uma ferramenta fácil de trabalhar e que atendeu às expectativas, não necessitando de manuais e/ou livros para seu entendimento.

Na parte de implementação, foi utilizado o ambiente de programação Borland Delphi 4. Sem dúvida, também uma ótima ferramenta, só que de nível de complexidade mais elevado. Não foi encontrada a documentação necessária sobre o componente RichEdit, principalmente, como detectar o final ou o início de uma linha a partir da posição atual do cursor, o que atrasou no início, o andamento do protótipo.

Com relação à transferência de arquivos por FTP, esta funcionou adequadamente. O componente FTP disponibilizado pelo ambiente de programação Borland Delphi 4, possui uma boa documentação e funcionalidade.

Sobre o assunto de edição orientada por gabaritos / sintaxe, os editores encontrados tiveram fundamental importância como embasamento para o desenvolvimento deste protótipo.

As características implementadas neste protótipo foram:

- a) Realce colorido para sintaxe;
- b) Auto preenchimento com o gabarito para os comandos mais utilizados (configurável pelo usuário);
- c) Edição de arquivos localizados em servidores remotos;

- d) Ferramentas de Busca / Substituição de Texto;
- e) Aumentar / Reduzir Identação;
- f) Formatação de Minúsculas / Maiúsculas;
- g) Desfazer ultima ação;
- h) Ativar Ferramentas Dataflex (DFComp, DFRun, DFFile, DFMaint, DFQuery).

Posteriormente, este protótipo de um sistema de edição por gabaritos e sintaxe, poderá ser utilizado para qualquer linguagem. Para isto é necessário apenas criar novos arquivos de definição de sintaxe e gabarito para a nova linguagem, como por exemplo: PASCAL, JAVA, C++, e outras, e especificar na janela de configurações, os nomes desses novos arquivos.

7.1 SUGESTÕES

Algumas sugestões para melhorias do protótipo são:

- a) Mais teclas devem ser analisadas, por exemplo, um problema que está acontecendo é o seguinte: Quando um usuário digita “//”, isto é, um comentário, se o usuário deslocar o cursor para cima, o texto continua aparecendo em azul e itálico. Assim, as teclas Delete, Up, Down, Left, Right, PgUp, PgDown, Tab deveriam ser analisadas.
- b) O processo de análise de cores que acontece após a abertura do está consideravelmente lenta. Isto porque análise está sendo feita baseada no conteúdo do RichEdit. Modificando este processo e baseando-se no conteúdo que estivesse na memória aumentaria muito a velocidade desta análise e modificação de cores.
- c) Hoje, não está se permitindo neste protótipo, abrir mais de um arquivo ao mesmo tempo, e isto deve ser melhorado, permitindo ao usuário editar vários arquivos ao mesmo tempo e ainda, editar vários arquivos de diferentes tipos, cada um com seu arquivo de definição de sintaxe e gabarito.
- d) E ainda, utilizando-se de uma BNF, o sistema já poderia checar a sintaxe do programa durante sua digitação.

REFERÊNCIAS BIBLIOGRÁFICAS

- [ARA99] **Araneae 2.78**. McINTYRE, Mark. Endereço Eletrônico: <http://plaza.v-wave.com/mark/products/araneae.html>
- [DAT99] Data Access Brasil Ltda. **Dataflex Console Mode**. Endereço Eletrônico: <http://www.dataaccess.com.br>
- [EDI99] **EditPlus+ 1.25**. ES-Computing. Endereço Eletrônico: <http://www.editplus.com>
- [FTP99] **FTP Edit 2.20**. HENRIKSSON, Kristoffer. Endereço Eletrônico: <http://www.ftpedit.com>
- [GAN83] GANE, Chris. **Análise Estruturada de Sistemas**. Rio de Janeiro : Livros Técnicos e Científicos, 1983.
- [GRA99] GRAHL, Everaldo Arthur. **Metodologia de Desenvolvimento de Sistemas**. Blumenau : FURB, 1999. Endereço Eletrônico: www.inf.furb.rct-sc.br/~grahl
- [MCM91] McMENAMIN, M. Stephen. **Análise Essencial de Sistemas**. São Paulo : McGraw-Hill, 1991.
- [OLI96] OLIVEIRA, Gorki Starlin da Costa. **Dataflex 3.01b**. São Paulo : Makron Books, 1996.
- [POM95] POMPILHO, S. **Análise essencial : guia prático de análise de sistemas**. Rio de Janeiro : IBPI Press, 1995.
- [TXP99] **TextPad 4.0**. Helios Software Solutions Inc. Endereço Eletrônico: <http://www.textpad.com>