

**Ferramenta de apoio a
reestruturação de código
fonte em linguagem C++
baseado em padrões de
legibilidade**

Denis Alberto Dalmolin

Orientador: Everaldo Artur Grahl

Roteiro de Apresentação

- Introdução
- A Linguagem C
- Qualidade de Software
- Padrões Estudados
- Análise do Programa Fonte
- Descrição do Software
- Diagrama de Casos de Uso
- Diagrama de Classes
- Diagrama de Seqüência
- Definição do Software
- Análises do Software
- Funcionamento do Software
- Conclusão

Introdução

- As empresas procuram aumentar a qualidade do software
- As empresas necessitam de padrões de desenvolvimento do código fonte para agilizar e reduzir custos com a manutenção do software
- Apesar de toda a flexibilidade da linguagem C e C++, os programas podem se tornar desorganizados e difíceis de serem compreendidos

Objetivo Principal

- Especificar e implementar uma ferramenta que apoie a reestruturação de código fonte escrito em linguagem C++ utilizando padrões de legibilidade. Estes padrões foram obtidos a partir de pesquisa bibliográfica.

A Linguagem C

- Nos anos 70, a linguagem C foi inventada e implementada por Dennis Ritchie
- Para resolver o problema do seu difícil gerenciamento foi desenvolvido a linguagem C++
- Linguagem C++ possui suporte à programação orientada ao objeto

Qualidade de Software

- Nos primeiros anos da sua existência o software era projetado sob medida
- Com o crescimento do tamanho e da complexidade dos programas, aumentou a existência de diversas propostas de métodos para o desenvolvimento de programas com elevado nível de qualidade
- A reestruturação do código-fonte contribui para a melhoria da produtividade na manutenção de sistemas, conseqüentemente aumentando a qualidade do software

Padrões

- **Padrão de composição de módulos**
 - como criar e declarar módulos de definição e de implementação em C e C++
- **Padrão geral**
 - como utilizar a linguagem padrão , fazer declarações, uso de expressões, comandos e sequência de execução de comandos
- **Padrão de estilo**
 - espaçamento da margem esquerda, estilo de declarações de variáveis, estilo de expressões e atribuições, estilo de blocos e estruturas de controle
- **Padrão para a escolha de nomes**
 - trata sobre a estrutura genérica do nome de variáveis

Análise do Programa Fonte

- Para o desenvolvimento do software, seguiu-se a mesma linha da análise de código feita por compiladores, que segue as três fases:
 - Análise linear: extrai os tokens
 - Análise hierárquica: análise dos tokens

Descrição do Software

- Ferramentas utilizadas
 - *Rational Rose*
 - *Microsoft Visual C++ 6.0*
- Principal função do software é efetuar a reestruturação de código-fonte em C++
 - Análise que gera avisos
 - Análise que gera modificações

Diagrama de Casos de Uso

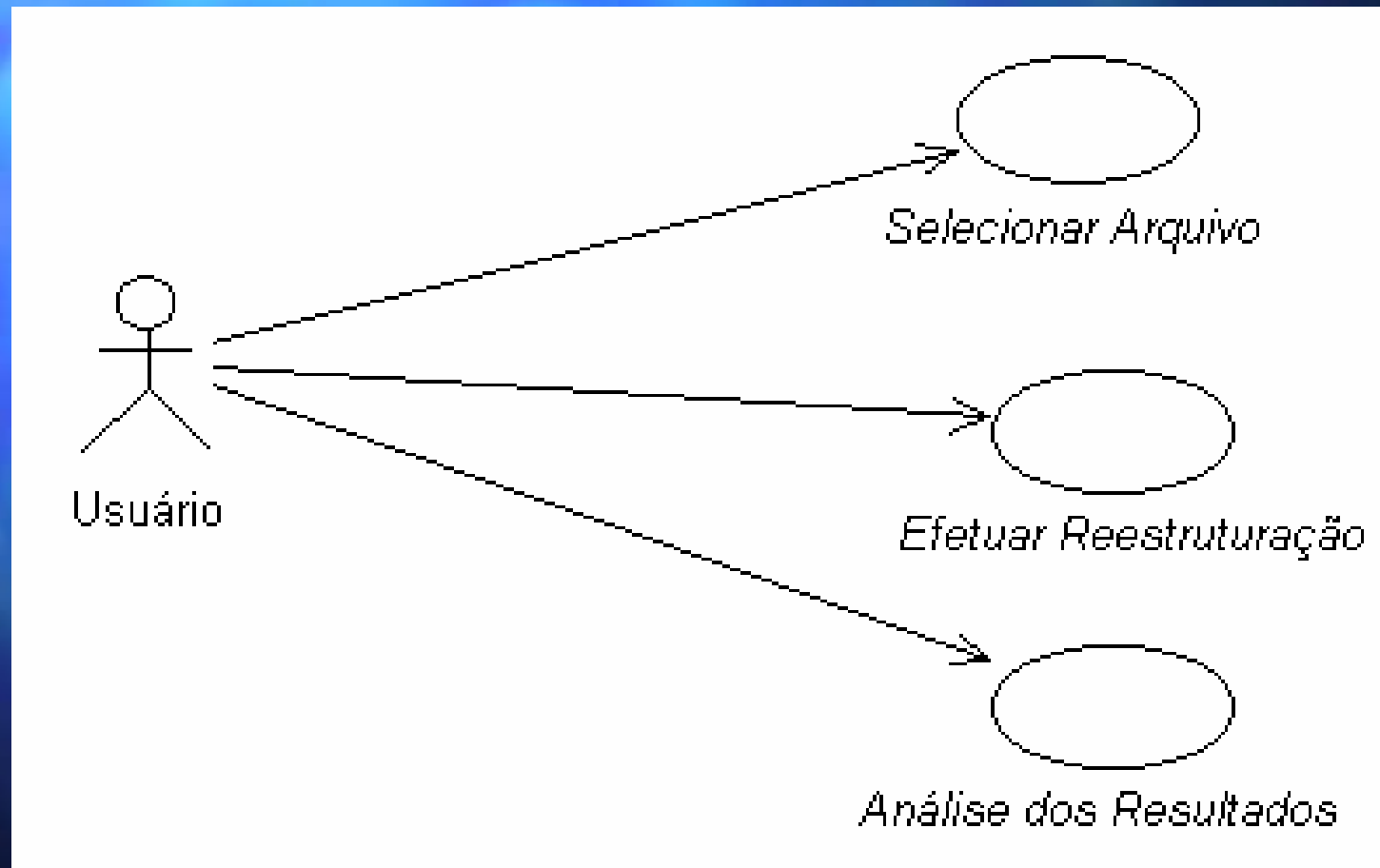


Diagrama de Classes

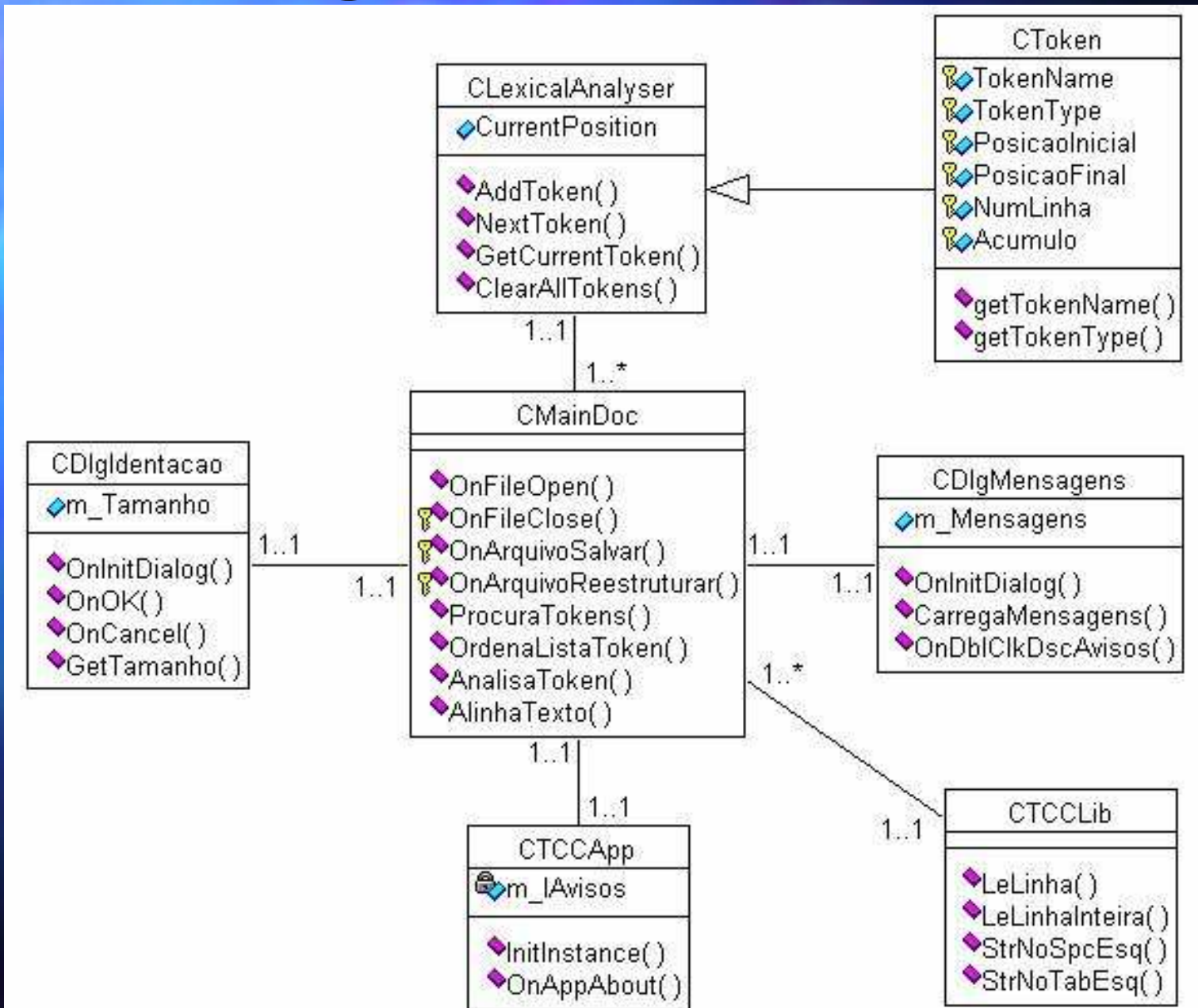
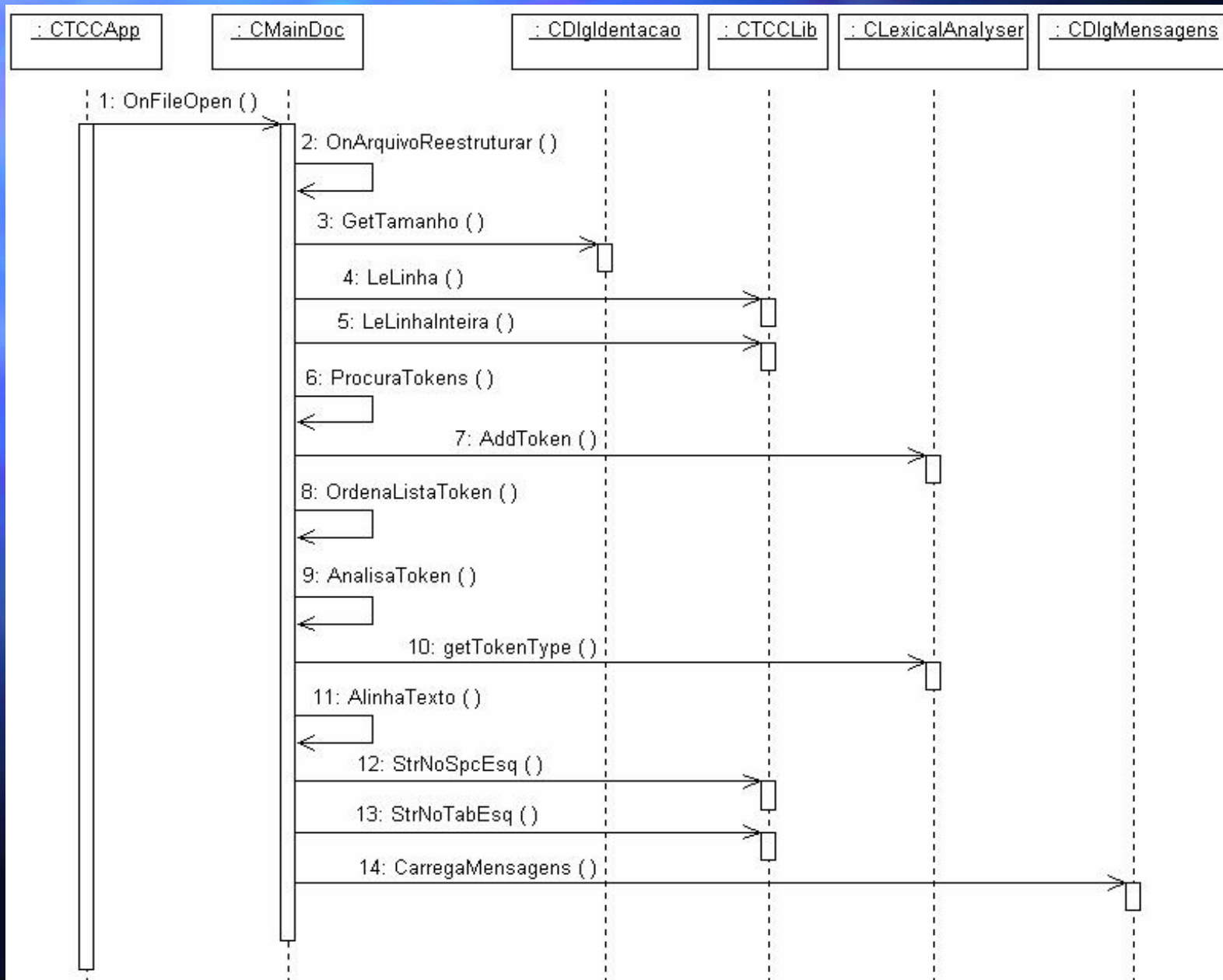
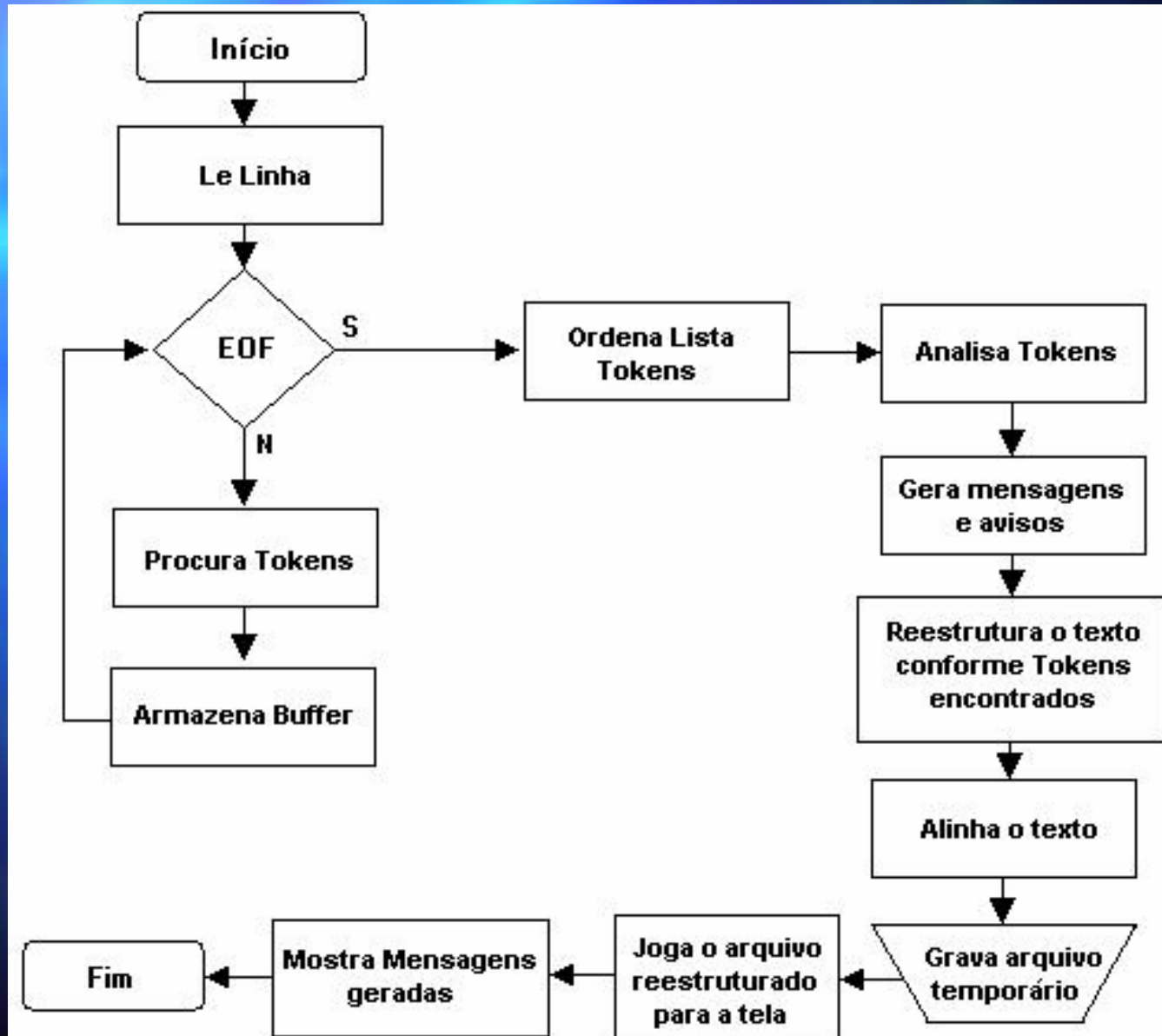


Diagrama de Seqüência



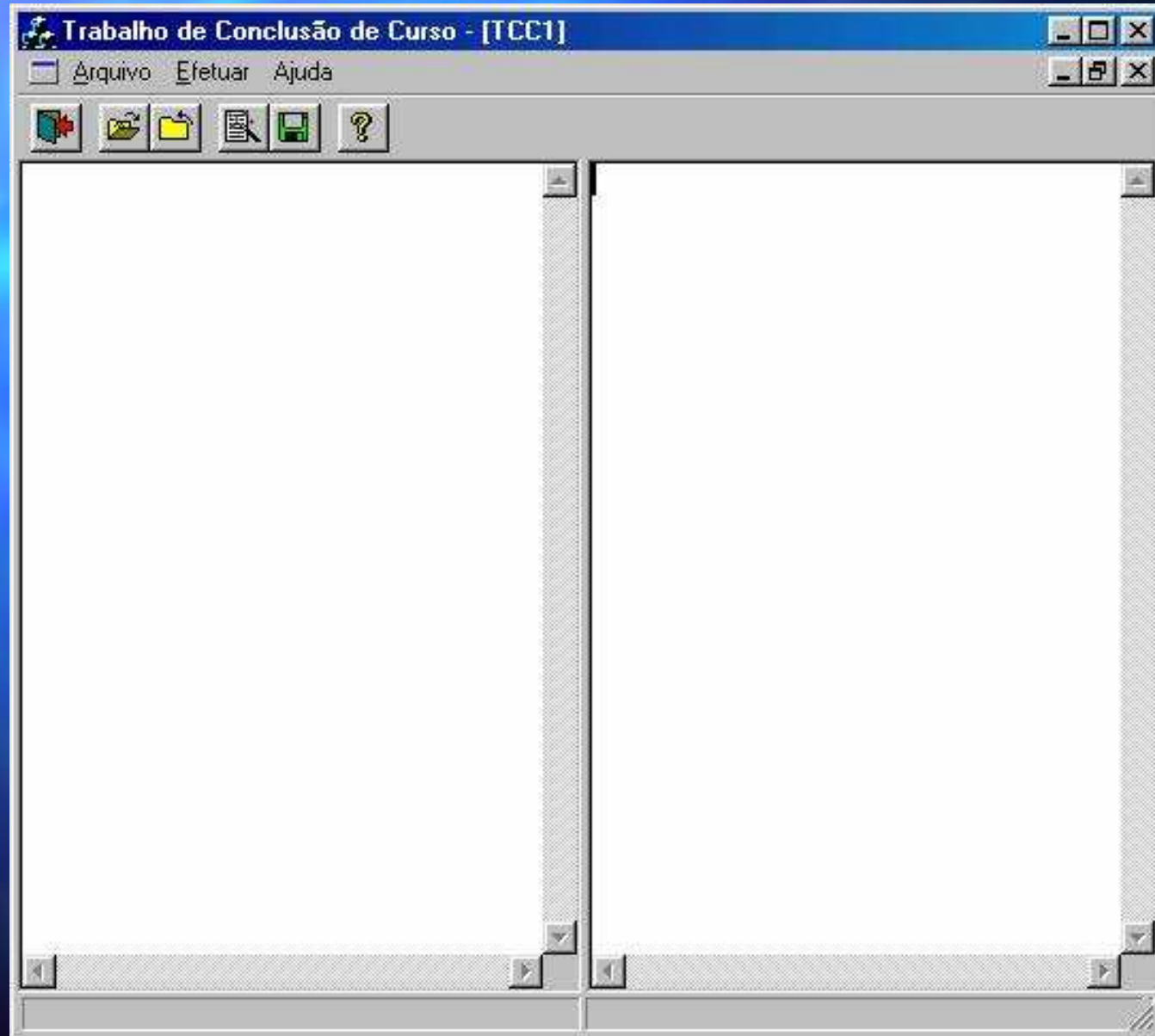
Definição do Software



Análises do Software

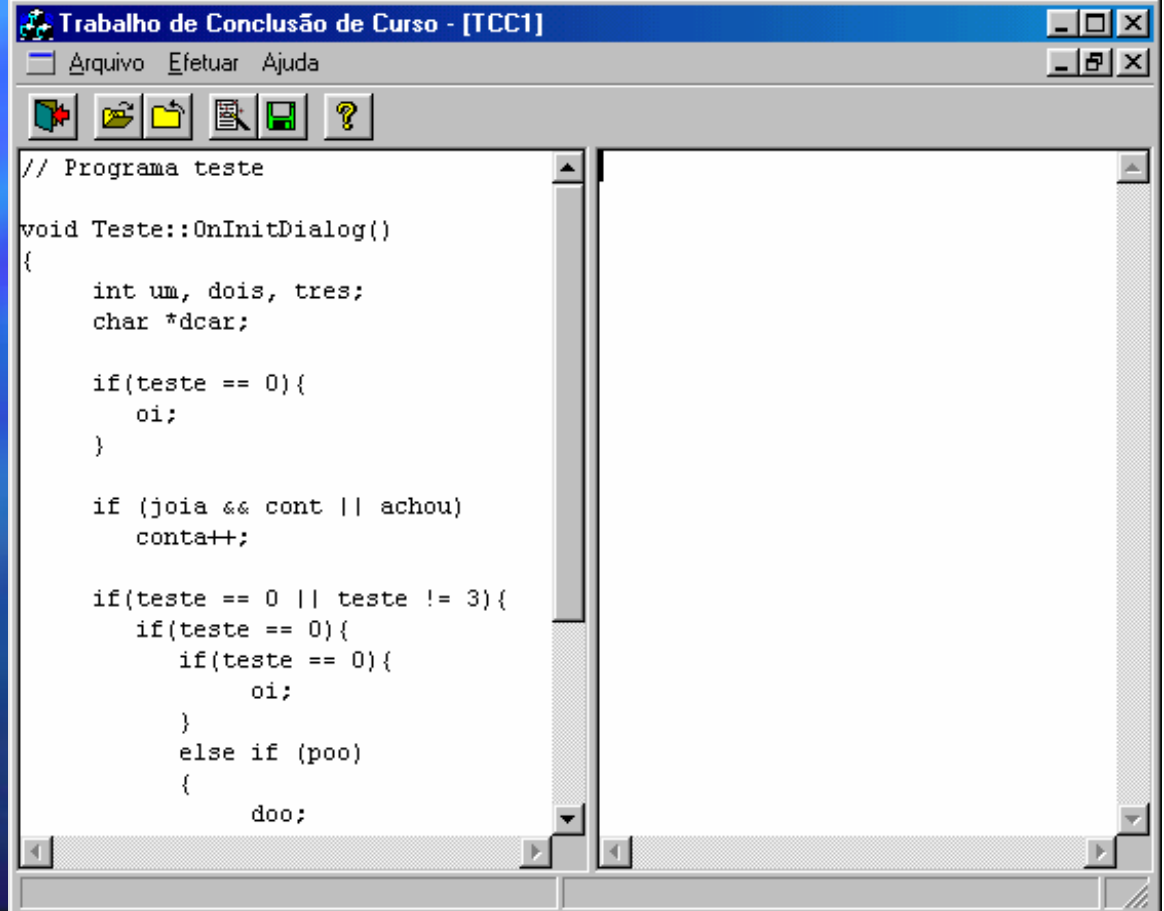
- goto
- operador ?
- continue
- friend
- malloc
- realloc
- free
- case sem break
- switch sem default
- else final em blocos else if
- bloco case grande
- tamanho linha
- uso increm. expressões
- alinhamento do código
- variável linha individual
- espaço na declaração de ponteiros
- estrutura controle separar abertura da condição
- colocar operadores lógicos em linhas separadas
- alinhar início e fim bloco
- condições devem ter início e fim de bloco
- início e fim de bloco em linhas exclusivas

Funcionamento do Software





Funcionamento do Software



Funcionamento do Software



A window titled "Mensagens" (Messages) with a close button (X) in the top right corner. It contains a table with three columns: "Linha" (Line), "Erro" (Error), and "Ação" (Action).

Linha	Erro	Ação
40	Encontrado bloco else if sem o else final	Recomenda-se colocar o else.
63	Encontrado switch sem o comando default	Recomenda-se colocar.
65	Encontrado case sem o comando break	Recomenda-se colocar.
65	Bloco case com mais de 5 linhas	Recomenda-se criar função.
78	Encontrado case sem o comando break	Recomenda-se colocar.

Funcionamento do Software

The screenshot shows a software development environment window titled "Trabalho de Conclusão de Curso - [TCC1]". The menu bar includes "Arquivo", "Ejetuar", and "Ajuda". The toolbar contains icons for file operations and help. A "Mensagens" (Messages) window is open, displaying a table of error messages:

Linha	Erro	Ação
40	Encontrado bloco else if sem o else final	Recomenda-se colocar o else.
63	Encontrado switch sem o comando default	Recomenda-se colocar.
65	Encontrado case sem o comando break	Recomenda-se colocar.
65	Bloco case com mais de 5 linhas	Recomenda-se criar função.
78	Encontrado case sem o comando break	Recomenda-se colocar.

Below the messages window, two code editors are visible, side-by-side. Both editors show the same C code snippet:

```
}  
  
switch (joaquim)  
{  
  case 1:  
    oii;  
  
  default:  
    welcome;  
}
```

F U N C I O N A M E N T O . D O S O F T W A R E

Trabalho de Conclusão de Curso - [TCC1]

Arquivo Ejetuar Ajuda

```
// Programa teste
void Teste::OnInitDialog()
{
    int um, dois, tres;
    char *dcar;
    if(teste == 0){
        oi;
    }
    if (joia && cont || achou)
        conta++;
    if(teste == 0 || teste != 3){
        if(teste == 0){
            oi;
        }
        else if (poo)
        {
            doo;
        }
        else
        {
            dad;
        }
    }
}

```

Verifique o espaçamento na declaração de ponteiro

Verifique que foi colocado um espaço

Verifique que o início do bloco foi colocado em nova linha

Verifique que foi colocado um início e um fim de bloco

Verifique que foi alinhado os campos em 5 posições, tamanho informado antes da reestruturação

```
// Programa teste
void Teste::OnInitDialog()
{
    int um,
    dois,
    tres;
    char * dcar;
    if (teste == 0)
    {
        oi;
    }
    if (joia
    && cont
    || achou)
        conta++;
    if (teste == 0
    || teste != 3)
    {
        if (teste == 0)
        {
            oi;
        }
        else if (poo)
        {
            doo;
        }
        else
        {
            dad;
        }
    }
}

```

Verifique que cada declaração de variável foi colocada em linhas separadas

Verifique que as condições do if foram separadas, uma em cada linha

Nota: Neste exemplo foi utilizado o if, mas o programa trata com todos os operadores (while, do, for, switch, case, etc...)

Conclusão

- Os padrões utilizados são de grande valia, pois tornam o código fonte mais legível, aumentando a qualidade do software e consequentemente diminuindo os futuros custos com a manutenção dos mesmos
- A utilização de tokens agilizou a reestruturação
- Os objetivos do trabalho foram alcançados visto que a ferramenta permite analisar e reestruturar código-fonte de programas feitos na linguagem C++, a partir de padrões de legibilidade estudados na bibliografia
- Limitações
 - o padrão para escolha de nomes e o padrão de composição de módulos possuem regras e/ou recomendações que não puderam ser implementadas.

Demonstração do Software

- Ferramenta de apoio a reestruturação de código fonte em linguagem C++ baseado em padrões de legibilidade