

# **FERRAMENTA DE APOIO A REESTRUTURAÇÃO DE CÓDIGO FONTE EM LINGUAGEM PL/SQL BASEADO EM PADRÕES DE LEGIBILIDADE**

Formando: Dyckson Dyorgio Dolla

Orientador: Prof. Everaldo Artur Grahl

Julho / 2001

# ROTEIRO DA APRESENTAÇÃO

- INTRODUÇÃO;
- OBJETIVOS;
- CONCEITOS BÁSICOS;
- TRABALHOS CORRELATOS;
- REQUISITOS DO PROBLEMA A SER TRABALHADO;
- TÉCNICAS E FERRAMENTAS UTILIZADAS;
- ESPECIFICAÇÃO;
- IMPLEMENTAÇÃO;
- CONCLUSÃO;
- EXTENSÕES;

# INTRODUÇÃO

Construção e manutenção de software:

- Legibilidade;
- Documentação interna no código fonte;
- Lógica na construção.

Automatizar a formatação e padronização de código fonte para alcançar melhores resultados no processo de manutenção e correção de software.

# OBJETIVOS

- Construir uma ferramenta que apóie a reestruturação de código fonte escrito em linguagem PL/SQL utilizando padrões de legibilidade;
- Efetuar a indentação do código fonte;
- Incluir documentação em pontos-chave do código fonte;
- Demonstrar avisos sobre determinadas construções não-recomendadas;
- Formatar ou modificar estruturas para atender os requisitos de legibilidade estudados;

# CONCEITOS BÁSICOS - ORACLE E PL/SQL

- Linguagem SQL: definição, manipulação e controle de dados de um banco de dados relacional.
- PL/SQL:
  - Linguagem de quarta geração criada pela ORACLE;
  - Integração nativa com o banco de dados e ferramentas;
  - Portabilidade;
  - Segurança.

# CONCEITOS BÁSICOS - ORACLE E PL/SQL

```
DECLARE
    CURSOR num1_cur IS SELECT num FROM num1_tab
        ORDER BY sequence;
    CURSOR num2_cur IS SELECT num FROM num2_tab
        ORDER BY sequence;
    num1          num1_tab.num%TYPE;
    num2          num2_tab.num%TYPE;
    pair_num      NUMBER := 0;
BEGIN
    OPEN num1_cur;
    OPEN num2_cur;
    LOOP
        FETCH num1_cur INTO num1;
        FETCH num2_cur INTO num2;
        IF (num1_cur%FOUND) AND (num2_cur%FOUND) THEN
            pair_num := pair_num + 1;
            INSERT INTO sum_tab VALUES (pair_num, num1 + num2);
        ELSE
            EXIT;
        END IF;
    END LOOP;
    CLOSE num1_cur;
    CLOSE num2_cur;
END;
/
```

# CONCEITOS BÁSICOS - QUALIDADE DE SOFTWARE

- Atividades de reestruturação são raramente ou nunca executadas pela falta de recursos ou ferramentas.
- Legibilidade se degenera com o tempo, à medida que mudanças são realizadas no sistema.
- Facilidade de leitura e entendimento dos procedimentos para os quais o programa foi criado.

# CONCEITOS BÁSICOS - COMPILADORES

- Analisador léxico lê os caracteres de entrada e produz uma sequência de tokens para a análise sintática.
- A ferramenta lê as linhas de um arquivo texto e gera uma lista em memória de todos os tokens extraídos de cada linha.
- É verificado se os tokens extraídos condizem com a gramática da linguagem, para algumas das tarefas de reestruturação.

# TRABALHOS CORRELATOS

Foi redigido por Denis Alberto Dalmolin (2000), a proposta de uma ferramenta para reestruturação de código fonte da linguagem de programação C/C++.

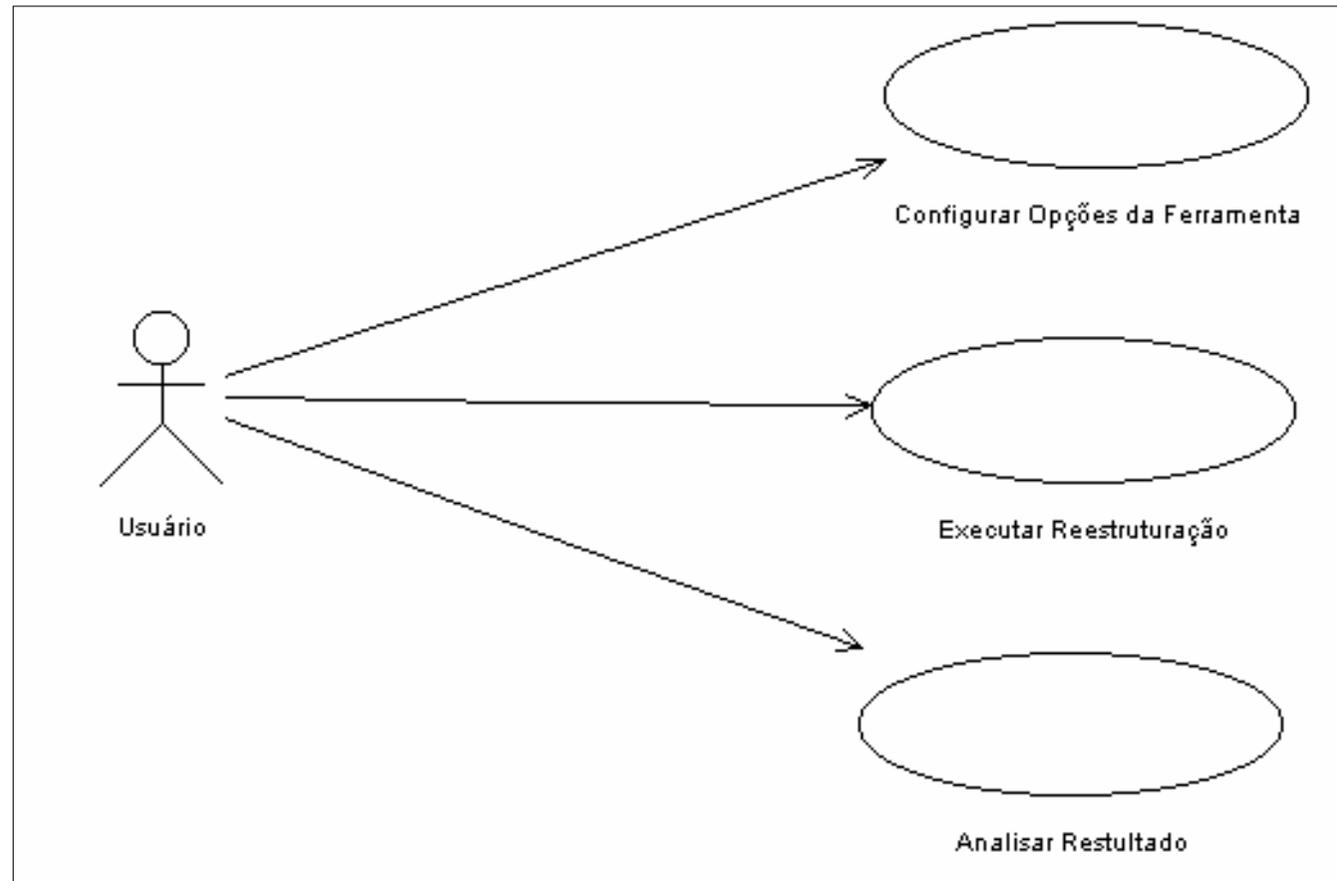
# REQUISITOS DO PROBLEMA A SER TRABALHADO

- Gerar documentação no início do programa;
- Gerar documentação antes de cada subrotina;
- Gerar documentação tabela X operação;
- Formatar palavras reservadas;
- Formatar a cláusula ORDER BY;
- Formatar as variáveis e as colunas das tabelas;
- Incluir tratamento de exceção nos blocos BEGIN..END;
- Realizar a indentação do código fonte;
- Gerar aviso sobre uso do comando GOTO;
- Gerar aviso sobre o uso de literais;

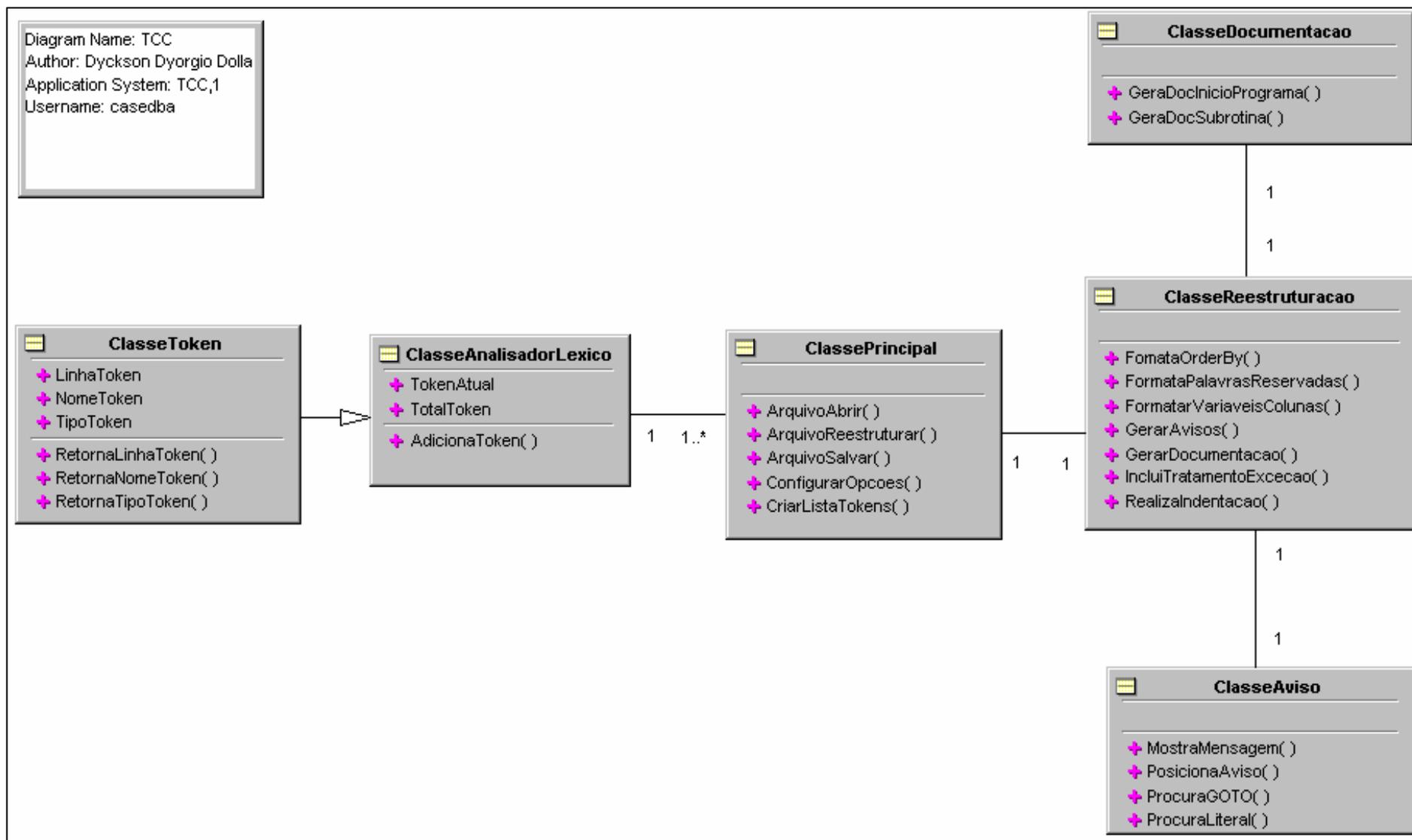
# **ESPECIFICAÇÃO - Técnicas e Ferramentas**

- Análise Orientada a Objetos utilizando U.M.L. (Unified Modelling Language).
- Ferramentas Utilizadas: Oracle Designer 6.0 e Rational Rose 4.0

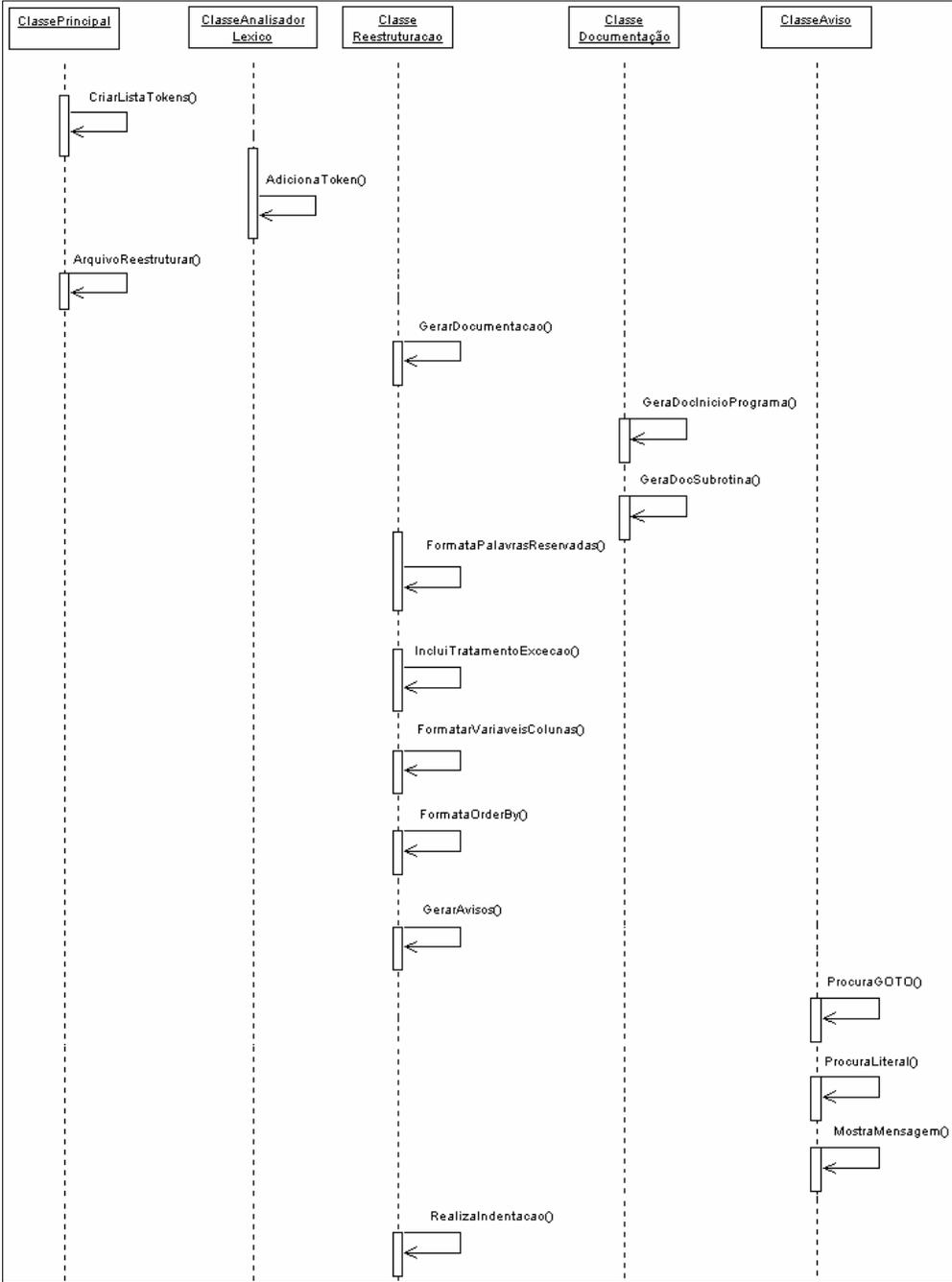
# Diagrama de Caso de Uso



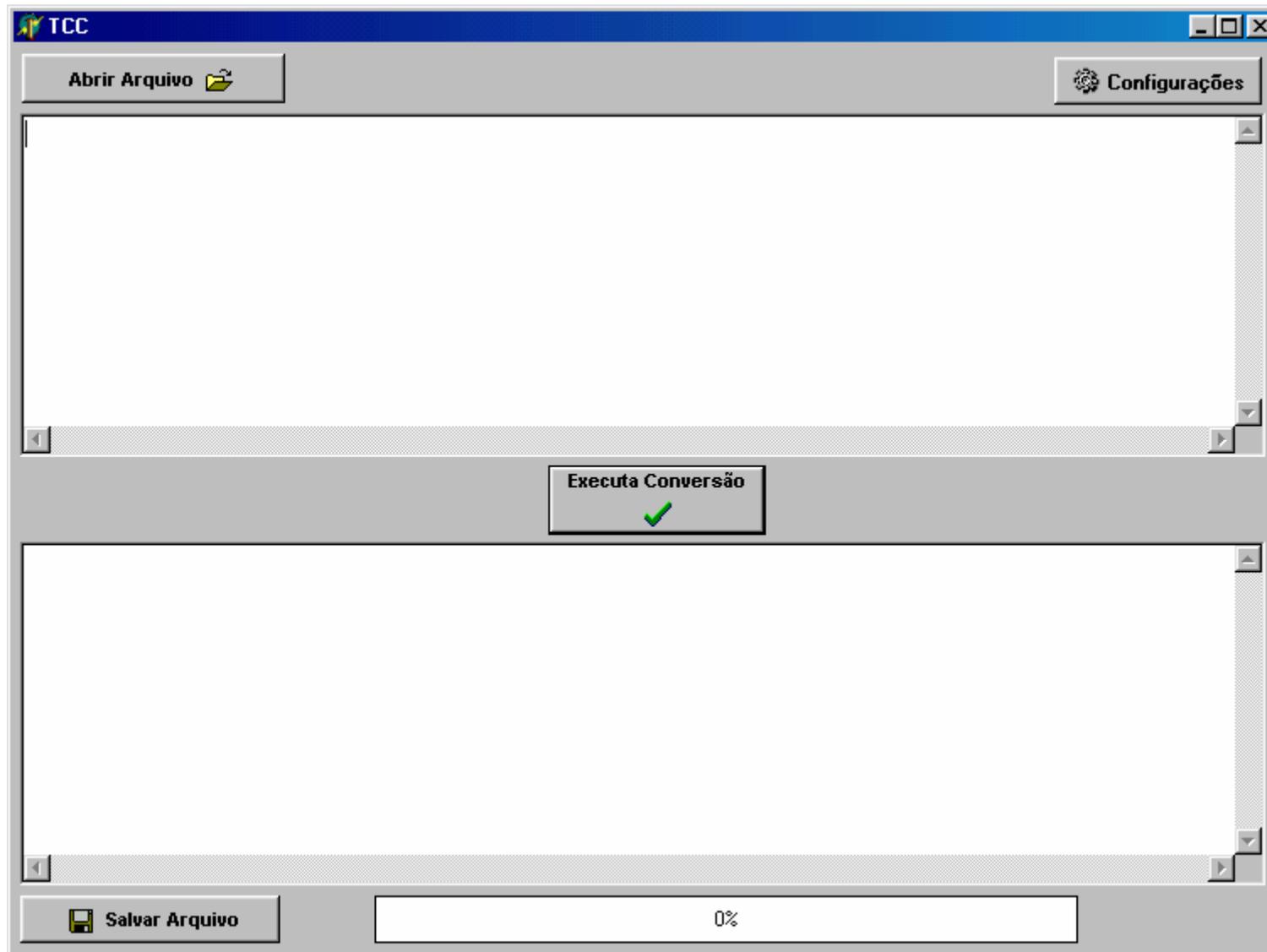
# Diagrama de Classes



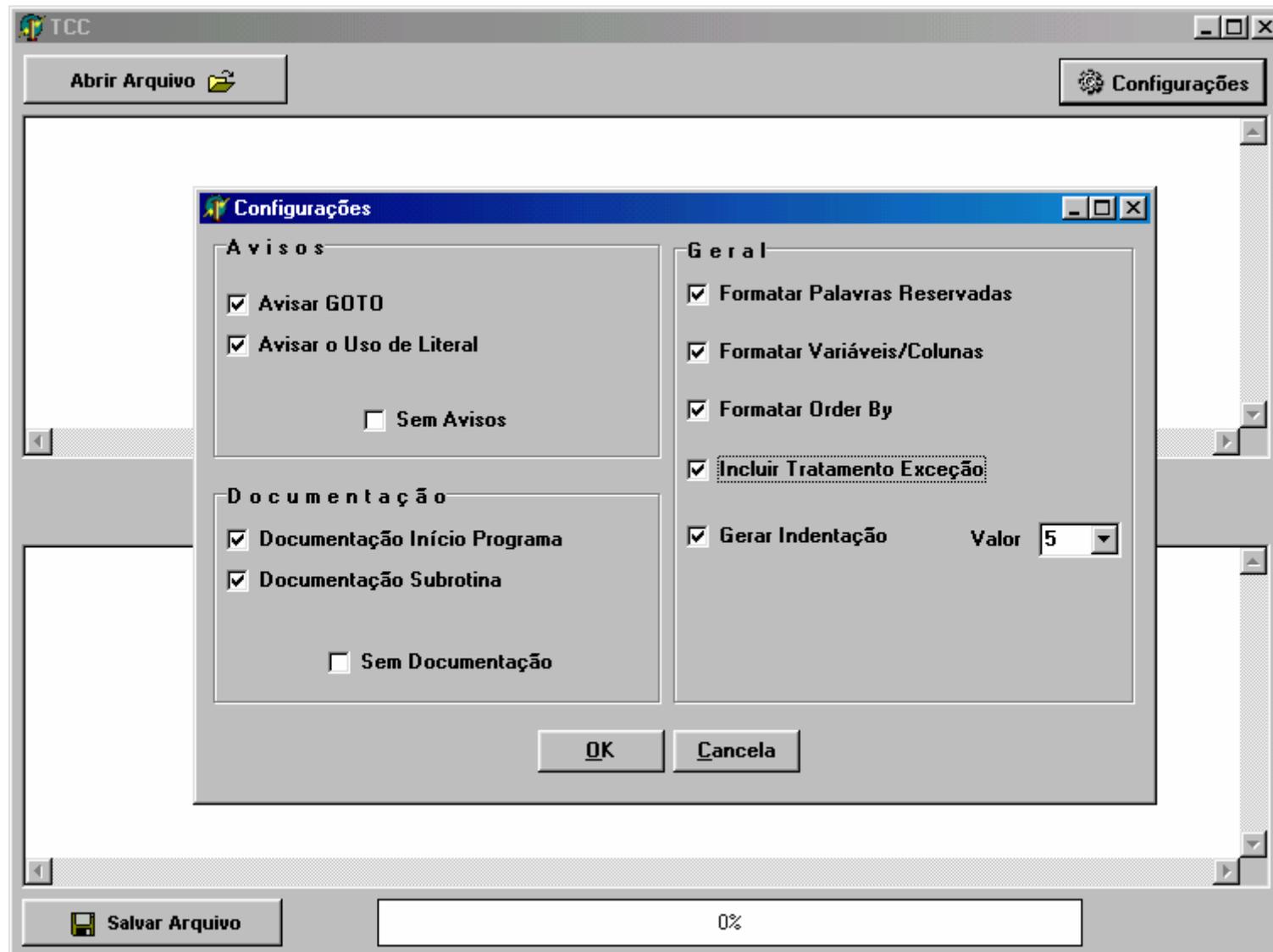
# Diagrama de Seqüência



# Tela inicial do protótipo



# Tela de configurações



## Tela após processo de reestruturação

The screenshot displays a software application window titled 'TCC'. The main area is a code editor showing PL/SQL code. The code is presented in two states: the original code at the top and the converted code at the bottom. A 'Mensagens' (Messages) dialog box is open, listing messages found during the conversion process. The messages are all 'Encontrado Literal 1' (Literal 1 found) and are associated with specific line numbers. A progress bar at the bottom indicates that the conversion process is 100% complete. The interface includes buttons for 'Abrir Arquivo' (Open File), 'Configurações' (Settings), 'Executa Conversão' (Execute Conversion), and 'Salvar Arquivo' (Save File).

```
BEGIN
if r0001.cd_divisao_gerencia not in (7,8) then
if aprova_pedido(r0001.vl_unitario_pedido,r0001.vl_un
null;
end if;
elsif r0001.cd_divisao_gerencia = 9 then
  null;
else
if busca_condicao_pgt_pedido <> - 1 then
if busca_tabela_vigente <> -1 then
if busca_destino_tab_preco <> -1 then
```

**Mensagens**

Mensagem	Linha
Encontrado Literal 1	3
Encontrado Literal 9	6
Encontrado Literal 1	9
Encontrado Literal 1	10
Encontrado Literal 1	11
Encontrado Literal 1	12
Encontrado Literal 1	13
Encontrado Literal 1	14
Encontrado Literal 1	15

```
BEGIN
  IF r0001.cd_divisao_gerencia NOT IN ( 7 , 8 ) THEN
    IF aprova_pedido ( r0001.vl_unitario_pedido ,
      NULL ;
    END IF ;
  ELSIF r0001.cd_divisao_gerencia = 9 THEN
    NULL ;
  ELSE
    IF busca_condicao_pgt_pedido <> - 1 THEN
      IF busca_tabela_vigente <> - 1 THEN
        IF busca_destino_tab_preco <> - 1 THEN
```

100%

# Formatação cláusula Order By

The screenshot shows the TCC application interface. At the top, there are buttons for 'Abrir Arquivo' and 'Configurações'. The main text area contains the following SQL code:

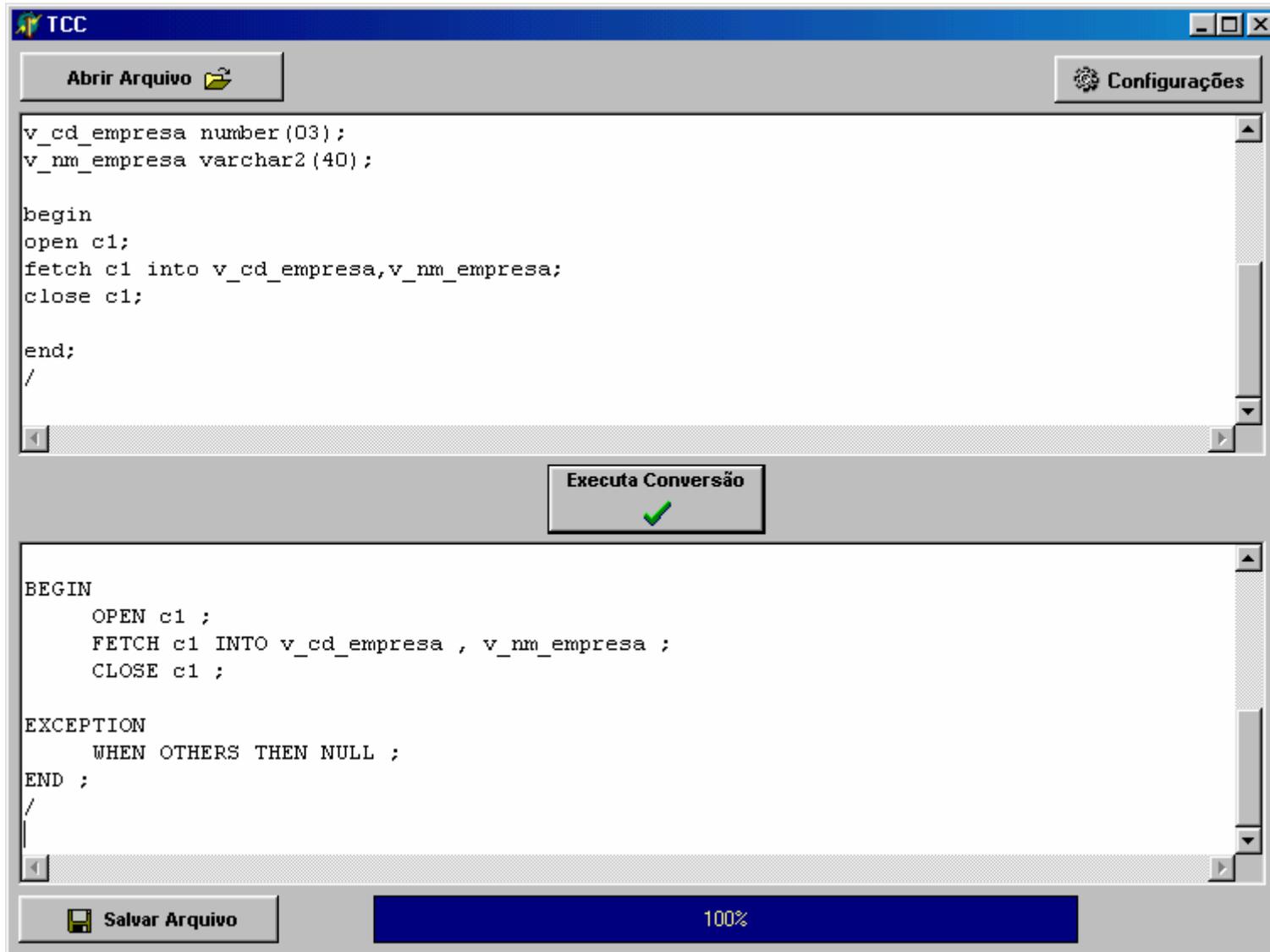
```
declare  
  
cursor c1 is  
select a.cd_empresa,  
b.nm_empresa  
from filial a,  
empresa b  
where a.cd_empresa = b.cd_empresa  
and a.id_tipo_empresa = 2  
order by 2,1;
```

Below the code is a button labeled 'Executa Conversão' with a green checkmark icon. The bottom text area shows the converted and reformatted SQL code:

```
DECLARE  
  
CURSOR c1 IS  
SELECT a.cd_empresa ,  
       b.nm_empresa  
FROM filial a ,  
      empresa b  
WHERE a.cd_empresa = b.cd_empresa  
      AND a.id_tipo_empresa = 2  
      ORDER BY b.nm_empresa , a.cd_empresa ;
```

At the bottom of the window, there is a 'Salvar Arquivo' button and a progress bar showing 100% completion.

# Inclusão tratamento de exceção



The screenshot shows the TCC (Toad) interface with a window titled "TCC". The window contains two text editors. The top editor shows the original PL/SQL code, and the bottom editor shows the converted code with an exception handler added. A "Executa Conversão" button with a green checkmark is located between the two editors. The bottom status bar shows "100%" and a "Salvar Arquivo" button.

**Original Code (Top Editor):**

```
v_cd_empresa number (03) ;  
v_nm_empresa varchar2 (40) ;  
  
begin  
open c1;  
fetch c1 into v_cd_empresa,v_nm_empresa;  
close c1;  
  
end;  
/
```

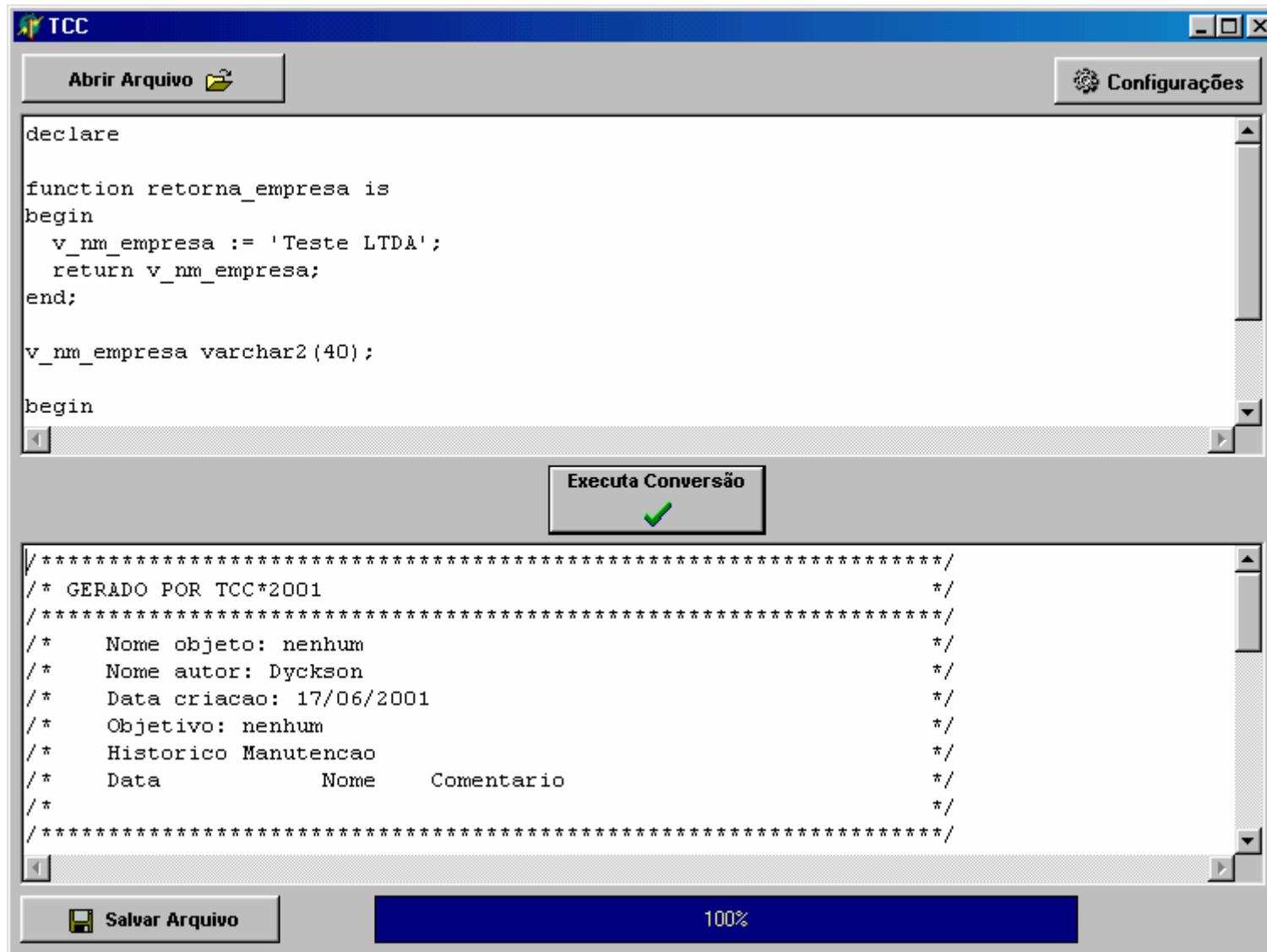
**Converted Code (Bottom Editor):**

```
BEGIN  
    OPEN c1 ;  
    FETCH c1 INTO v_cd_empresa , v_nm_empresa ;  
    CLOSE c1 ;  
  
EXCEPTION  
    WHEN OTHERS THEN NULL ;  
END ;  
/
```

**Buttons:** "Abrir Arquivo" (top left), "Configurações" (top right), "Executa Conversão" (center, with green checkmark), "Salvar Arquivo" (bottom left).

**Status Bar:** "100%" (bottom center).

# Documentação no início do programa



# Documentação das subrotinas

The screenshot shows the TCC (TOOL) interface with a blue title bar. At the top left is the 'Abrir Arquivo' button with a folder icon, and at the top right is the 'Configurações' button with a gear icon. The main window is divided into two text editors. The top editor contains the following PL/SQL code:

```
declare  
  
function retorna_empresa is  
begin  
  v_nm_empresa := 'Teste LTDA';  
  return v_nm_empresa;  
end;  
  
v_nm_empresa varchar2(40);  
  
begin
```

Below the top editor is a button labeled 'Executa Conversão' with a green checkmark icon. The bottom editor displays the converted and documented code:

```
/***/  
/***/ FUNCTION : RETORNA_EMPRESA          *****/  
/***/ OBJETIVO : nenhum                    *****/  
/***/  
FUNCTION retorna_empresa IS  
BEGIN  
  v_nm_empresa := 'Teste LTDA' ;  
  RETURN v_nm_empresa ;  
EXCEPTION  
  WHEN OTHERS THEN NULL ;  
END ;
```

At the bottom of the window, there is a 'Salvar Arquivo' button with a floppy disk icon on the left, and a blue progress bar on the right showing '100%' completion.

# EXEMPLO DO PROCESSO ANTES

```
function aprova_pedido(p_vl_unitario_pedido in number
,p_vl_tabela          in number
,p_id_consiste_preco  in number ) return number is
w_aux                number(01);
begin
begin
UPDATE PEDIDO_VENDA_CTR
SET    DT_PROCESSO      = trunc(sysdate)
,DT_ATUALIZACAO      = sysdate
,ID_USUARIO          = 'CPRG3000'
WHERE  CD_PEDIDO_VENDA = r0001.cd_pedido_venda;
exception
when others then
if sqlcode = -00060 then
w_ds_erro := 'Aprovacao. Erro deadlock pedido_venda_ctr.';
raise w_erro_deadlock;
else
w_ds_erro := 'Aprovacao. Erro update pedido_venda_ctr. ';
raise w_encerra_execucao;
end if;
end;
end aprova_pedido;
```

# EXEMPLO DO PROCESSO APÓS

```

/*****
/* GERADO POR TCC*2001
/*****
/*      Nome objeto: nenhum
/*      Nome autor: Dyckson
/*      Data criacao: 11/07/2001
/*      Objetivo: aprovacao
/*      Historico Manutencao
/*      Data      Nome      Comentario
/*
/*****
/****
/**** FUNCTION : APROVA_PEDIDO
/****      OBJETIVO : Aprovar pedidos
/****
FUNCTION aprova_pedido ( p_vl_unitario_pedido IN NUMBER
, p_vl_tabela IN NUMBER
, p_id_consiste_preco IN NUMBER ) RETURN NUMBER IS
w_aux NUMBER ( 01 ) ;
BEGIN
  BEGIN
    UPDATE pedido_venda_ctr
    SET dt_processo = trunc ( SYSDATE )
    , dt_atualizacao = SYSDATE
    , id_usuario = 'CPRG3000'
    WHERE cd_pedido_venda = r0001.cd_pedido_venda ;
  EXCEPTION
    WHEN OTHERS THEN
      IF SQLCODE = - 00060 THEN
        w_ds_erro := 'Aprovacao. Erro deadlock pedido_venda_ctr.' ;
        RAISE w_erro_deadlock ;
      ELSE
        w_ds_erro := 'Aprovacao. Erro update pedido_venda_ctr.' ;
        RAISE w_encerra_execucao ;
      END IF ;
    END ;
END aprova_pedido ;
```

# CONCLUSÃO

- Analisa e reestrutura código-fonte de programas feitos na linguagem PL/SQL, a partir de alguns dos padrões de legibilidade estudados na bibliografia.
- O protótipo transforma um código-fonte complicado de se manter e corrigir por falta de legibilidade em um programa de melhor leitura e compreensão
- Estimula o processo de documentação, auxiliando as manutenções futuras em pontos-chave ou críticos do código-fonte.

# LIMITAÇÕES

- Reestruturação de variáveis (nomenclatura).
- Correções na estrutura de execução dos programas.
- Busca de literais no código-fonte.

# EXTENSÕES

- Outras linguagens de programação disponíveis no mercado.
- Padronização para telas criadas nos ambientes de programação visual.