

AVALIAÇÃO DA QUALIDADE DO CÓDIGO FONTE ESCRITO EM PL/SQL

Acadêmica: Giselle Maфра Schlosser
Orientador: Everaldo Artur Grahl



www.furb.br



Roteiro

- ↳ Introdução
 - ↳ Objetivos do trabalho

- ↳ Fundamentação teórica

- ↳ Desenvolvimento da ferramenta
 - ↳ Requisitos principais
 - ↳ Especificação
 - ↳ Implementação
 - ↳ Resultados

- ↳ Conclusão

- ↳ Extensões



Introdução

- ↪ Analisar padrões de programação em fontes PL/SQL
 - ↪ A ferramenta permite definir os padrões
 - ↪ Cadastrar glossário de termos
 - ↪ Cadastrar prefixos
 - ↪ Gerar a aplicação Forms em vários arquivos PL/SQL
 - ↪ Validar os fontes de acordo com os padrões cadastrados



Objetivos do trabalho

Desenvolver uma ferramenta que permita cadastrar padrões de desenvolvimento e que faça a análise destes padrões.

↳ Desenvolver interfaces

➤ Para permitir a parametrização dos padrões a serem adotados pela empresa

↳ Listar os erros de padrões encontrados

↳ Gerar um novo código

➤ Para identificar as correções necessárias



Fundamentação teórica

- ↳ Conceitos estudados para o desenvolvimento da ferramenta
 - ↳ Padrões de programação
 - ↳ Escolha de nomes
 - ↳ Recuos, espaçamento e alinhamento
 - ↳ Comentários e documentação
 - ↳ Evitar o uso de literais
 - ↳ Entrada e saída de dados utilizando compiladores
 - ↳ Análise léxica
 - ↳ Análise sintática
 - ↳ Análise semântica
 - ↳ Linguagem PL/SQL
 - ↳ Tipos de declarações
 - ↳ Tipos de dados
 - ↳ Atribuições de valores à variáveis



Fundamentação teórica

- ↳ Linguagem PL/SQL
 - ↳ Herança de tipo e tamanho
 - ↳ Cursores
 - ↳ Controle condicional
 - ↳ Controle iterativo
 - ↳ Tratamento de exceções
 - ↳ Subprogramas
 - ↳ Parâmetros
 - ↳ Procedures
 - ↳ Function
 - ↳ Package
 - ↳ Comentários



Padrões de programação

- ↪ Escolha de nomes
 - Facilita o entendimento do seu significado

- ↪ Recuos, espaçamento e alinhamento
 - Identifica claramente o funcionamento e regras do programa

- ↪ Comentário e documentação
 - Facilitar a leitura do programa
 - Ter segurança nas regras

- ↪ Evitar o uso de literais
 - Atualizar em menos lugares possíveis
 - Agilizar a manutenção



Entrada e saída de dados utilizando compiladores

↳ Analisadores de linguagem

Para análise dos arquivos de entrada foram utilizados analisadores léxico, sintático e semântico.

↳ Análise léxica

- Fazer a leitura dos caracteres de entrada
- Produzir uma seqüência de tokens

↳ Análise sintática

- Obtém uma cadeia de tokens proveniente do analisador léxico
- Verifica se a cadeia pode ser gerada

↳ Análise semântica

- Utiliza árvore gerado pelo analisador sintático
- Determina o significador do código fonte



Linguagem PL/SQL

- ↳ Linguagem da Oracle
- ↳ Performance
- ↳ Portabilidade

↳ Tipos de declarações

- Declaração de variáveis
- Inicialização de variáveis

VARIÁVEL	TIPO
ww_nr_lidos	constant number(4) := 15;
ww_tentativa	number(4) := 0;
ww_nr_lidos_2	constant number(4) default 15;
ww_nr_tentativa2	number(4) default 0;
ww_cd_cliente	number(4) NOT NULL := 1;



↳ Tipos de dados

Definem o domínio de valores que cada coluna ou argumento pode possuir.

↳ Atribuição de valores às variáveis

```
SELECT qt_produto
       ,vl_produto
INTO ww_qt_produto
     ,ww_vl_produto
FROM item_nota_fiscal;

ww_total := ww_qt_produto * ww_vl_produto;
```



↳ Tipos de dados

Definem o domínio de valores que cada coluna ou argumento pode possuir.

↳ Atribuição de valores às variáveis

- Utilizando o operador de atribuição
- À partir de um comando select

↳ Herança de tipo e tamanho

```
Nome_da_Variável    Nome_da_Tabela.Nome_da_Coluna%TYPE;

Nome_da_Variável    Nome_da_Tabela%ROWTYPE;

Nome_da_Variável    Nome_da_Variável%TYPE;
```



↳ Tipos de dados

Definem o domínio de valores que cada coluna ou argumento pode possuir.

↳ Atribuição de valores às variáveis

- Utilizando o operador de atribuição
- À partir de um comando select

↳ Herança de tipo e tamanho

- Tipo de uma coluna de uma tabela
- Tipo de registro (linha inteira) de uma tabela
- Tipo de uma variável previamente declarada

↳ Cursores

São áreas compostas de linhas e colunas em memória que servem para armazenar o resultado de uma seleção

```
CURSOR nome_do_cursor (relação_de_parâmetros) IS
  SELECT ds_nome nome
         ,nr_idade idade
  FROM funcionário
  WHERE código = 1356;
```



Tipos de dados

Definem o domínio de valores que cada coluna ou argumento pode possuir.

Atribuição de valores às variáveis

```

DECLARE
  ww_vendas NUMBER(5);
  CURSOR c_produtos IS
    SELECT cd_produto
           ,vl_custo_medio
    FROM produto;
BEGIN
  FOR r_produtos IN c_produtos LOOP
    SELECT COUNT(*)
    INTO ww_vendas
    FROM item_nota_fiscal
    WHERE cd_produto = r_produtos.cd_produto;
    IF ww_vendas < 4 THEN
      UPDATE produto
      SET vl_custo_medio = vl_custo_medio * 0.95
      WHERE current of c_produtos;
      IF SQL%NOTFOUND THEN
        DBMS_OUTPUT.PUT_LINE('Erro na atualização!');
      END IF;
    END IF;
  END LOOP;
END;
    
```



↳ Controle condicional

➤ Comando if

```
IF condição THEN
    relação_de_comandos1
[ELSIF condição THEN
    relação_de_comandos2]
[ELSE
    relação_de_comandos3]
END IF;
```



↳ Controle condicional
 ➤ Comando if

↳ Controle iterativo
 ➤ Estrutura loop

```
LOOP
  Relação_de_comandos
  IF condição_de_saida THEN
    EXIT;
  END LOOP;
```



↳ Controle condicional

➤ Comando if

↳ Controle iterativo

➤ Estrutura loop

↳ Tratamento de exceções

➤ Aborta a execução

↳ Subprogramas

➤ Bloco PL/SQL armazenados no banco

↳ Parâmetros

➤ in

```
CREATE OR REPLACE PROCEDURE nome_procedure (argumento1 modo tipo_de_dados
                                             ,argumento2 modo tipo_de_dados
                                             ,argumenton modo tipo_de_dados)
IS ou AS
  Variáveis locais, constantes,...
BEGIN
  Bloco PL/SQL
END nome_procedure;
```




↳ Function

➤ Subprograma que retorna um valor

```
CREATE OR REPLACE FUNCTION nome_função (argumento1 IN tipo_de_dados
                                         ,argumento2 IN tipo_de_dados
                                         ,argumenton IN tipo_de_dados)
RETURN tipo_de_dados IS ou AS
BEGIN
    Bloco PL/SQL
END nome_função;
```



↳ Function

➤ Subprograma que retorna um valor

```

↳ CREATE OR REPLACE PACKAGE nome_da_package IS
    PROCEDURE nome_da_procedure (lista_de_parâmetros);
    FUNCTION nome_da_function (lista_de_parâmetros);
    Declaração de variáveis, constantes, exceções e cursores públicos
END nome_da_package;

CREATE OR REPLACE PACKAGE BODY nome_da_package IS
    Declaração de variáveis, constants, exceções e cursores privados

    PROCEDURE nome_da_procedure (lista_de_parâmetros) IS
    BEGIN
    END;
    FUNCTION nome_da_procedure (lista_de_parâmetros) IS
    RETURN tipo
    BEGIN
        RETURN
    END;
END nome_da_package;
    
```



↳ Function

- Subprograma que retorna um valor

↳ Package

- São objetos de banco de dados, equivalente a bibliotecas
- Procedures
- Funções
- Definições de cursores
- Variáveis e constantes
- Definição de exceções
- Corpo
- Especificação

↳ Comentários

- Linha única
- Linhas múltiplas



Desenvolvimento do trabalho

- ↳ Levantamento dos requisitos
- ↳ Análise e identificação das informações necessárias para listar os erros e gerar o arquivo reestruturado.
- ↳ Especificação da ferramenta através dos diagramas UML de casos de uso, classes e atividades
- ↳ Implementação da ferramenta



Requisitos da ferramenta

↳ Requisitos funcionais

- ↳ Permitir cadastrar o glossário de palavras abreviadas para o entendimento do domínio da aplicação
- ↳ Permitir cadastrar os prefixos
- ↳ Permitir cadastrar os parâmetros estabelecidos pela empresa para a análise dos padrões
- ↳ Permitir gerar vários arquivos contendo blocos PL/SQL à partir de um arquivo gerado por uma aplicação Forms 6i
- ↳ Permitir gerar um arquivo contendo o conteúdo do arquivo PL/SQL reestruturado de acordo com os padrões cadastrados
- ↳ Permitir gerar uma listagem contendo os erros encontrados, listando linha a linha e identificando qual foi o comando que apresentou o erro



Requisitos da ferramenta

↳ Requisitos funcionais

- ↳ Permitir destacar em vermelho o conteúdo que se apresenta fora dos padrões
- ↳ Permitir gerar um relatório estatístico com os principais problemas identificados em vários programas

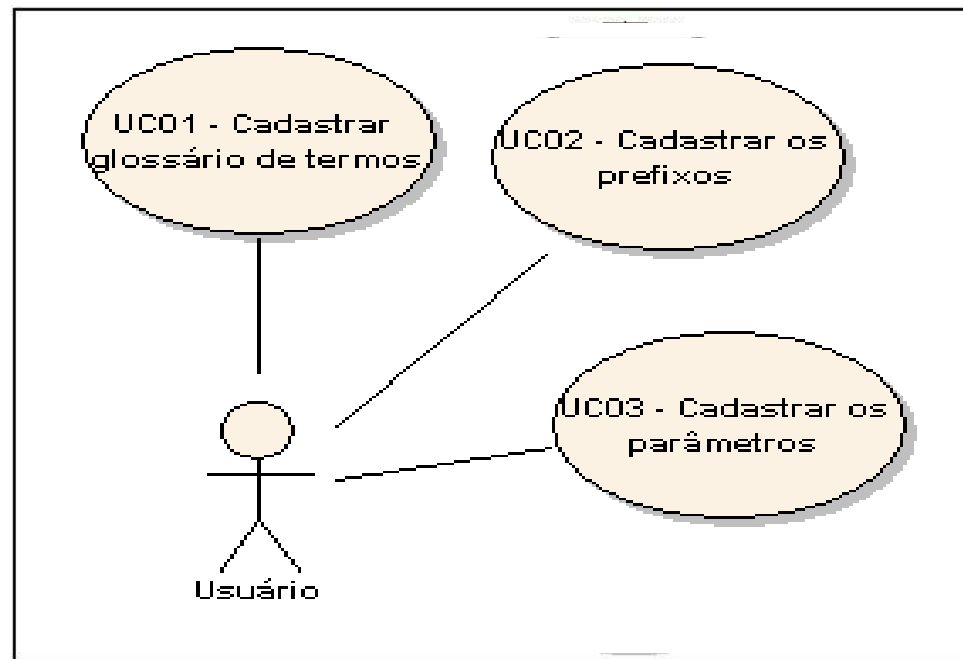
↳ Requisitos não-funcionais

- ↳ Ser compatível com o sistema operacional Windows
- ↳ Ser desenvolvida utilizando o ambiente Borland Delphi
- ↳ Utilizar o banco de dados Interbase

Especificação da ferramenta

- ↪ Enterprise Architect
 - ↪ UML
 - ↪ Diagrama de casos de uso
 - ↪ Diagrama de classes conceitual
 - ↪ Diagrama de atividades

↪ Diagrama de Casos de uso



Especificação da ferramenta

↳ Diagrama de Casos de uso

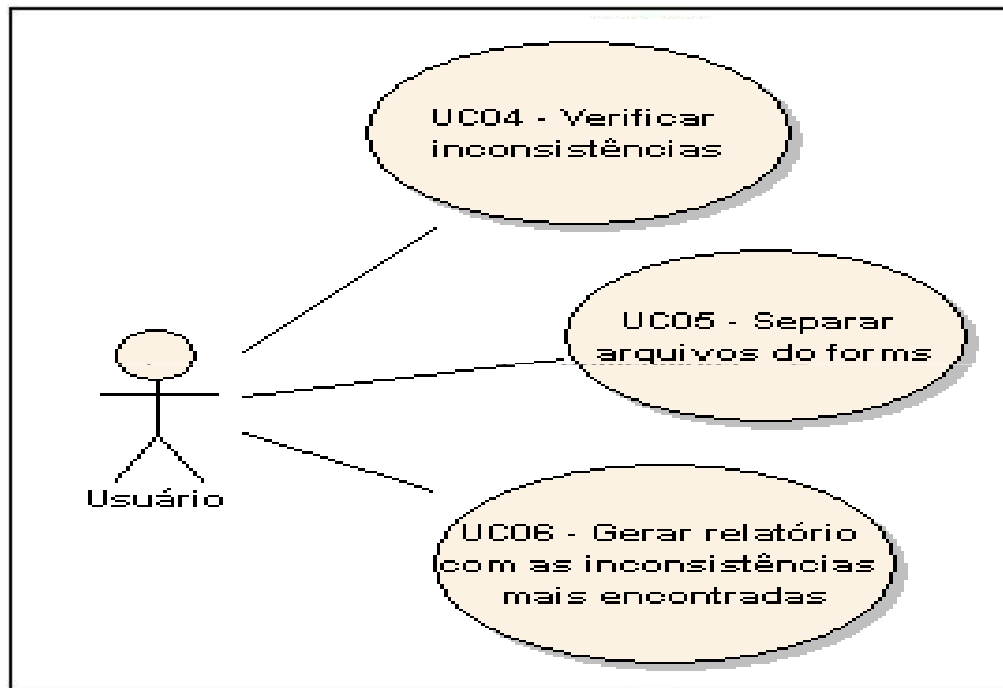




Diagrama de Classes Conceitual

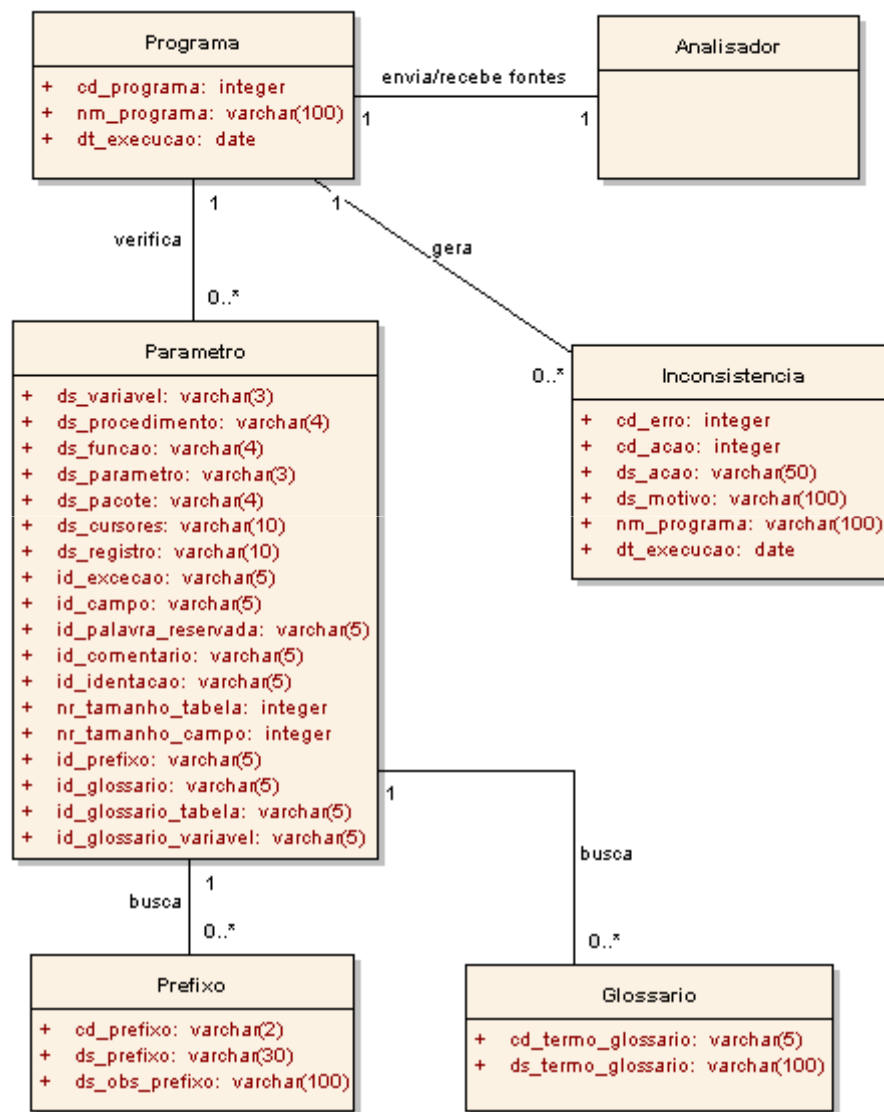
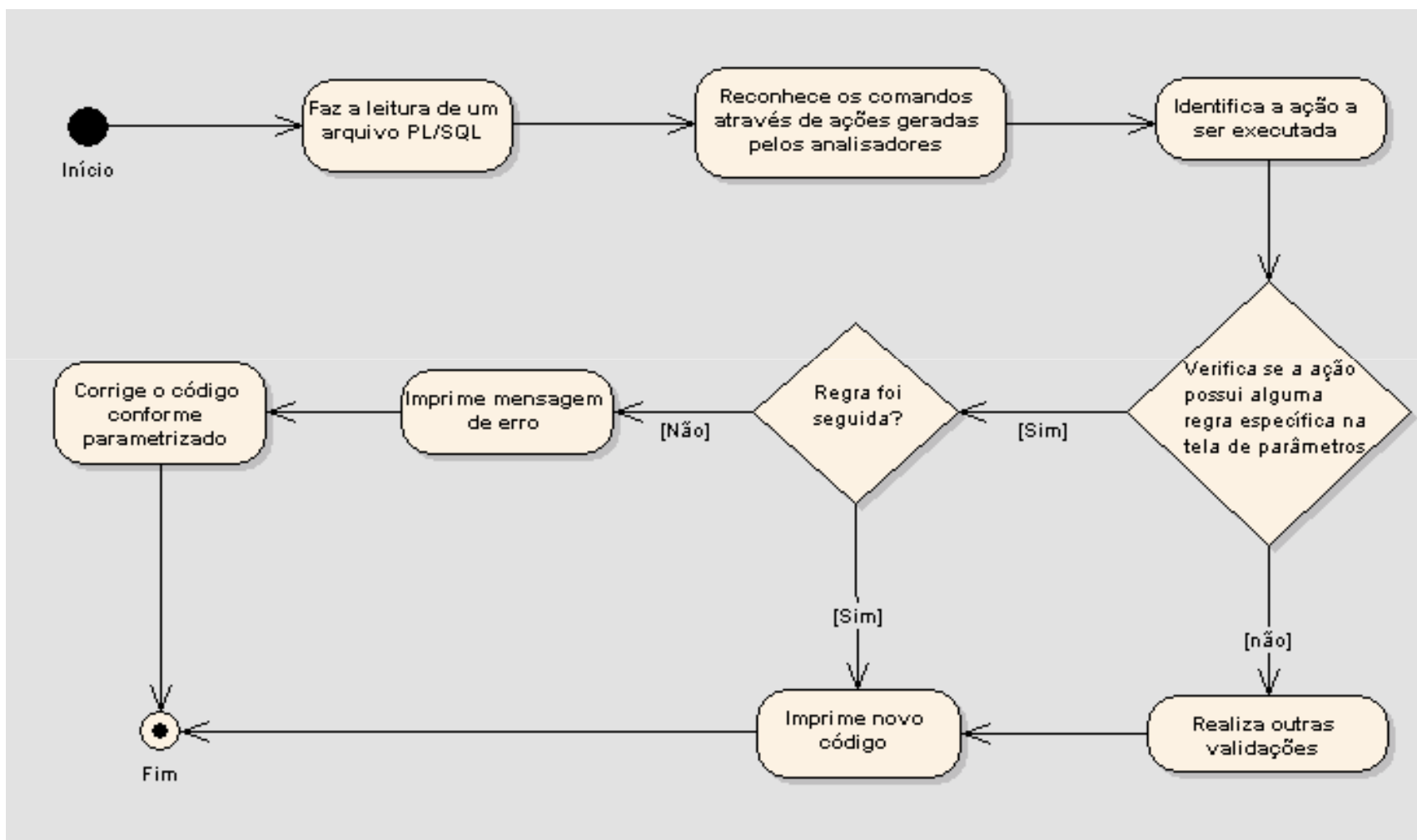


Diagrama de atividades





Implementação

↳ Técnicas e ferramentas utilizadas

↳ GALS

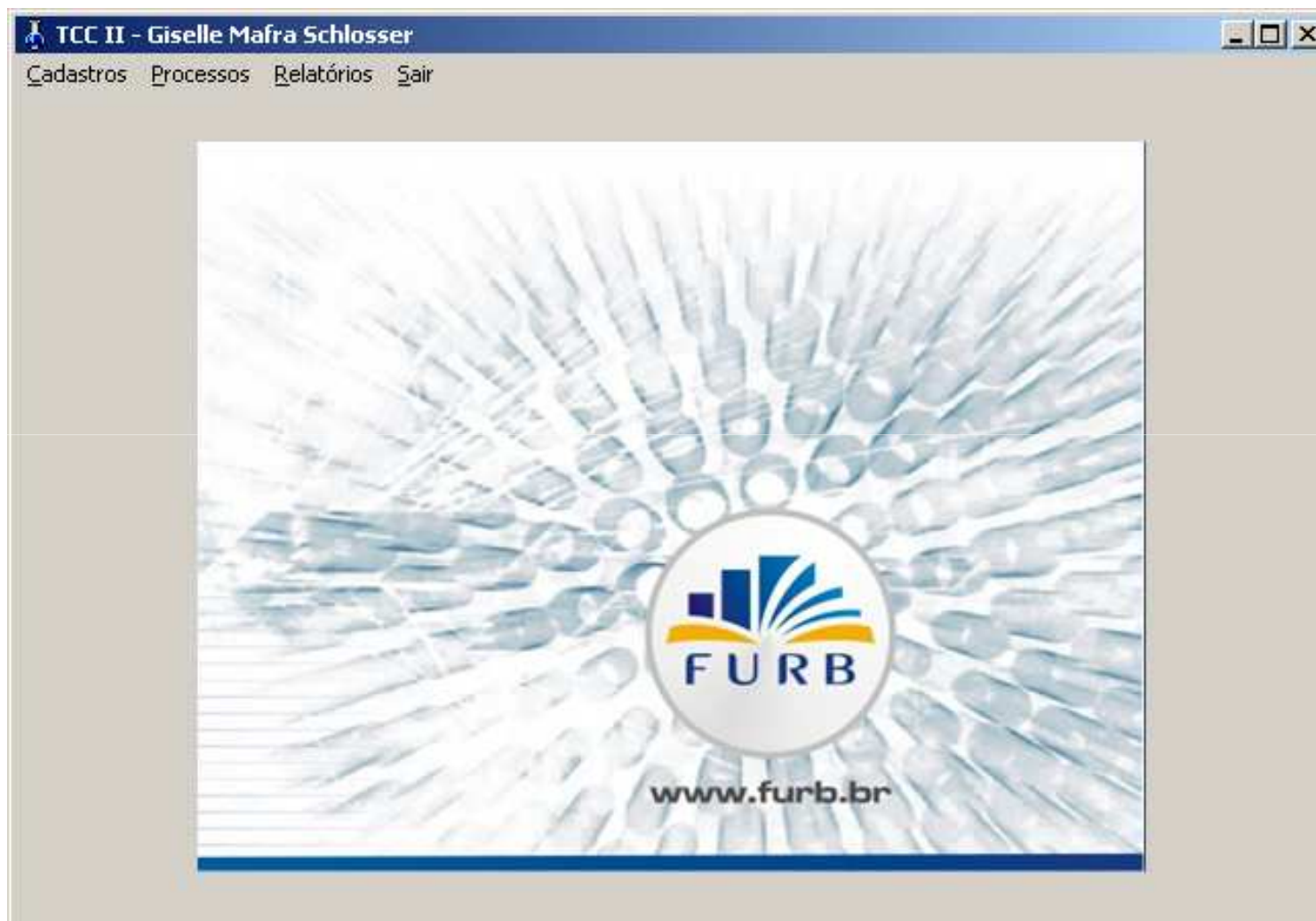
↳ Especificação da gramática e geração das análises léxica e sintática

↳ Delphi

↳ A ferramenta foi desenvolvida utilizando Delphi



Operacionalidade





Operacionalidade

↳ Cadastros de glossário de termos

Manter Glossário de Termos

Novo Alterar Gravar Excluir Consulta Sair

Código: ABA

Nome: ABATE

↳ Cadastros de prefixos

Manter Prefixos

Novo Alterar Gravar Excluir Consulta Sair

Código: CD

Descrição: CODIGO

Observação: IDENTIFICADOR DE CODIGO



Operacionalidade

↳ Cadastros de glossário de termos

Manter Glossário de Termos

Novo Alterar Gravar Excluir Consulta Sair

Código: ABA

Nome: ABATE

↳ Cadastros de prefixos

Manter Prefixos

Novo Alterar Gravar Excluir Consulta Sair

Código: CD

Descrição: CODIGO

Observação: IDENTIFICADOR DE CODIGO



Operacionalidade

↳ Cadastros de parâmetros

Manter Parâmetros para Padronização Código Fonte

Nomenclatura

Variáveis:
 Packages:
 Nome Campos: Prefixo Glossário
 Procedures:
 Cursores:
 Nome Tabela: Glossário
 Function:
 Parâmetros:
 Variáveis: Glossário

Identação/Comentários

Palavras Reservadas Maiúscula
 Comentários no Início do Programa
 Identação

Tratamento de Exceção

Final Programa
 INSERT
 UPDATE
 DELETE

Tamanho

Tabela: (Caracteres)
 Campos: (Caracteres)



Operacionalidade

↳ Transformação arquivos forms

```

TCC II - Giselle Mafra Schlosser - ppsk0240.txt
Arquivo Edição Sair
* Name PPSK0240
- Subclass Information
* Comments
*****
FORMS MODULE NAME
PPSK0240
MODIFIED
RCOSTA3 by Form Generator 20-DEC-2005 09:21
WORKAREA
GLOBAL SHARED WORKAREA
APPLICATION SYSTEM
PPSK (1)
PURPOSE
.
MODULE DESCRIPTION
MODULE NOTES
Executar
    
```




Operacionalidade

Validar fonte

```

declare
  wx_teste0 NUMBER;
  ww_teste2 varchar2(30);
  xx_teste3 varchar2(35);
  rr_teste teste.cd_codigo%type;
  ww_teste4 EXCEPTION;

  CURSOR cur_teste is
  SELECT e.cd_giselle_mafra_ina
  from testetestetestetestetesteste e,
  teste2 a
  where e.cd_teste = a.cd_teste;

begin

  FOR reg_teste IN c_teste loop

  --abertura de alguma coisa dfsdfasdfsdf ssss
  open c_teste;
  fetch c_teste
  into ww_teste,
  ww_teste2;
  IF (ww_teste = 'MARIA' AND ww_teste IS NOT NULL)
  
```

```

DECLARE

  wx_teste0 number;
  ww_teste2 varchar2(30);
  xx_teste3 varchar2(35);
  rr_teste teste.cd_codigo%type;
  ww_teste4 exception;

  CURSOR cur_teste IS
  SELECT e.cd_giselle_mafra_ina
  FROM testetestetestetestetesteste e,
  teste2 a
  WHERE e.cd_teste = a.cd_teste;

BEGIN

  FOR reg_teste IN c_teste LOOP

  -- abertura de alguma coisa dfsdfasdfsdf ssss
  OPEN c_teste;
  FETCH c_teste
  INTO ww_teste,
  ww_teste2;
  IF ( ww_teste = 'MARIA' AND ww_teste IS )
  
```

```

Linha 1: Palavra reservada declare deve estar em maiúsculo;
Linha 3: Variável wx_teste0 está com o nome incorreto;
Linha 5: Variável xx_teste3 está com o nome incorreto;
Linha 6: Variável rr_teste está com o nome incorreto;
Linha 10: Cursor cur_teste está com o nome incorreto;
Linha 10: Palavra reservada is deve estar em maiúsculo;
Linha 11: O termo "giselle" não existe na tabela de glossário!
Linha 11: O termo "mafra" não existe na tabela de glossário!
Linha 11: A abreviação "ina" não existe na tabela de glossário!
Linha 12: Palavra reservada from deve estar em maiúsculo;
Linha 12: O nome da tabela testetestetestetestetesteste possui mais de 28 caracteres;
Linha 14: Palavra reservada where deve estar em maiúsculo;
Linha 17: Palavra reservada begin deve estar em maiúsculo;
Linha 19: Palavra reservada loop deve estar em maiúsculo;
Linha 22: Palavra reservada open deve estar em maiúsculo;
  
```



Resultados e discussão

- ↳ Com a realização deste trabalho, pode-se demonstrar como analisar os padrões de um código fonte escrito em PL/SQL sem a necessidade de abrir o código e analisar linha a linha.
- ↳ Através da ferramenta proposta, busca-se dar flexibilidade as empresas a cadastrarem seus próprios padrões tanto em boas práticas de programação quanto na definição de nomenclaturas
- ↳ A ferramenta também permite o cadastro dos prefixos e glossário de termos específicos utilizados pela empresa, a fim de validar a nomenclatura baseado nestes cadastros



Conclusão

- ↪ A ferramenta pode auxiliar as empresas a definirem padrões a serem estabelecidos e utilizados por desenvolvedores
- ↪ Estabelecer padrões no desenvolvimento de um software, além de obter um nível de qualidade satisfatório, facilita o entendimento do negócio, bem como auxilia os desenvolvedores em futuras manutenções
- ↪ A geração de um relatório contendo estatísticas dos erros mais cometidos, pode auxiliar no trabalho de preparação e desenvolvimento dos programadores para o seguimento dos padrões



Extensões

- ↳ Aumentar a gramática da linguagem PL/SQL no GALS, visto que alguns comandos não foram tratados neste trabalho

- ↳ Desenvolver uma gramática para comandos do Forms, pois os arquivos extraídos do Forms utilizados como exemplo neste trabalho, apenas utilizavam comandos específicos da linguagem PL/SQL



Demonstração da ferramenta