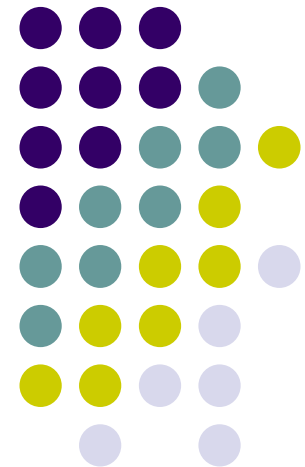


FERRAMENTA PARA DESENVOLVIMENTO DE OBJETOS DE BANCO DE DADOS EM LINGUAGEM PROCEDURAL

Acadêmico: Davi Rodrigo Bianchi

Orientador: Alexander Roberto Valdameri



Roteiro

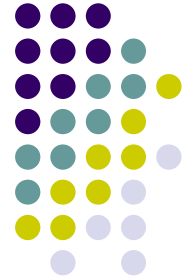


- Introdução
- Objetivos
- Fundamentação
- Especificação
- Implementação
- Operacionalidade
- Conclusões
- Relevância Pessoal

Introdução



- Sistemas Gerenciadores de Banco de Dados
- Compiladores



Objetivos

- Extensão do trabalho de conclusão desenvolvido por Hiebert (2003)
- Definição de novas estruturas para a linguagem procedural
- Geração de código para os SGBDs abordados
- Conexão aos SGBDs abordados

Compiladores

- Análise léxica
- Autômatos finitos
- Análise sintática
- Gramáticas livres de contexto



Compiladores

- Análise descendente
- Analisador preditivo LL(1)
- Tradução dirigida por sintaxe
- Geração de código



Sistemas Gerenciadores de Banco de Dados



- Oracle
- SQL Server
- PostgreSQL

Linguagem SQL



- *Data Definition Language* – DDL
- *Data Manipulation Language* – DML
- *Data Query Language* – DQL

Procedimentos Armazenados



- Procedimentos internos do SGDB
- Criados através da linguagem DDL específica de cada SGDB
- PL/SQL
- Transact-SQL
- PL/pgSQL



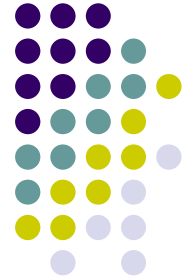
Trabalhos Correlatos

- Trabalho correlato de Hiebert (2003)
- Protótipo de compilador para a linguagem PL/SQL
- Geração de código para os SGBDs Oracle e SQL Server



Requisitos

- Possibilidade de escrita de um procedimento na linguagem procedural definida
- Detecção de erros através de análises léxica e sintática
- Geração de código para os SGBDs abordados
- Conexão aos SGBDs abordados para criação do procedimento, para fins de validação



Requisitos

- Possibilidade de salvar procedimento editado em arquivo
- Possibilidade de abrir um arquivo já salvo para edição



Especificação

Especificação através dos seguintes diagramas da UML:

- Diagrama de casos de uso
- Diagrama de classes
- Diagrama de atividades

Casos de uso

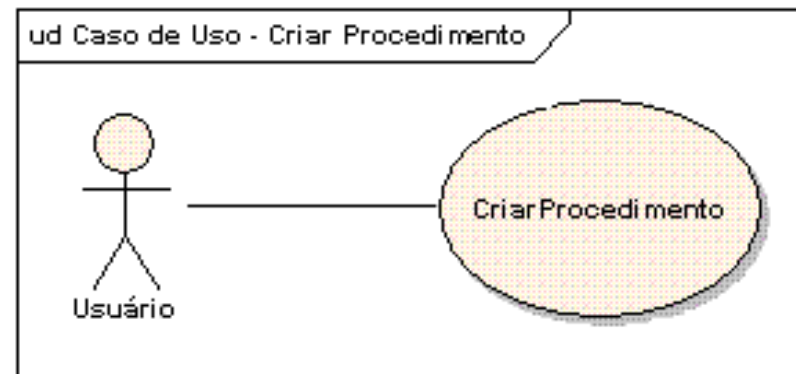
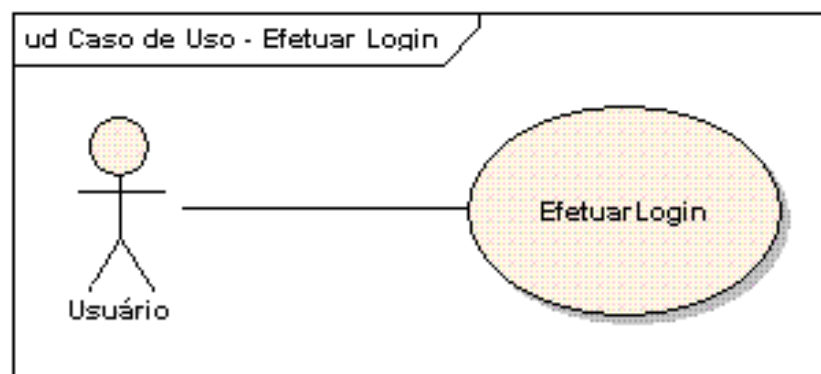
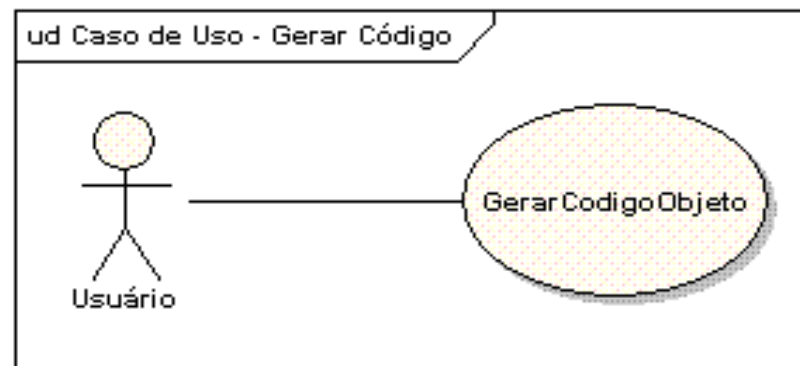
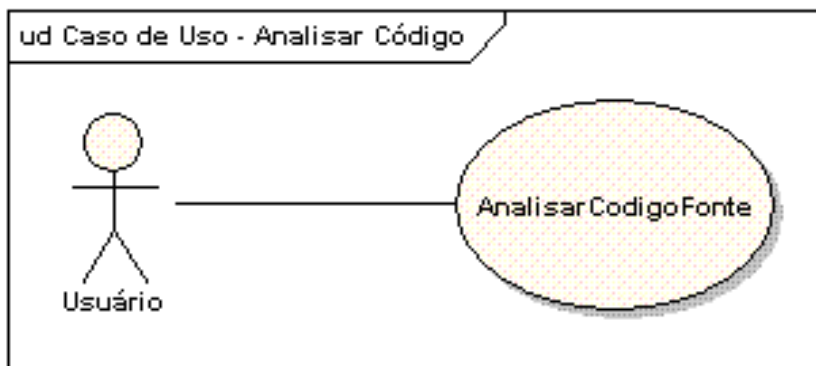


Diagrama de Classes

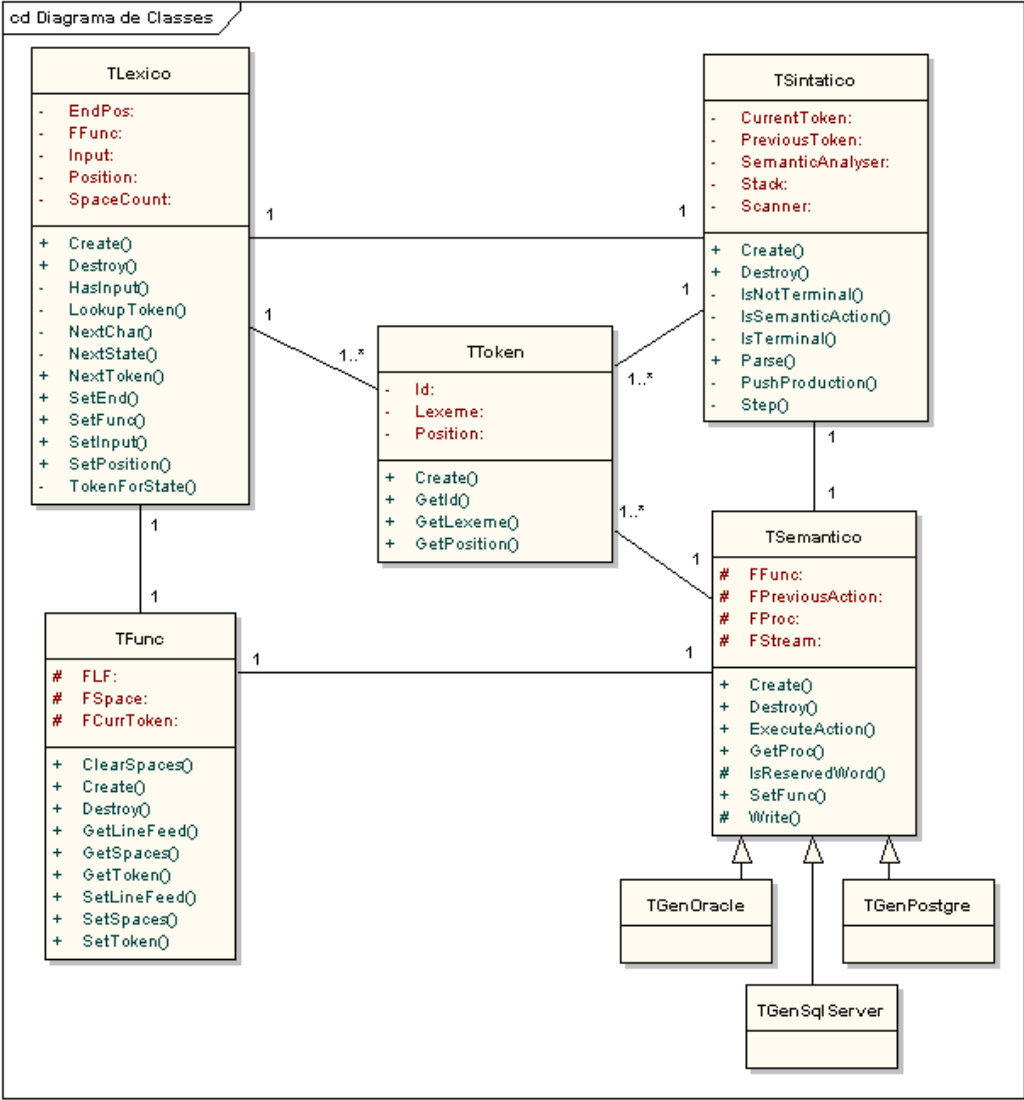
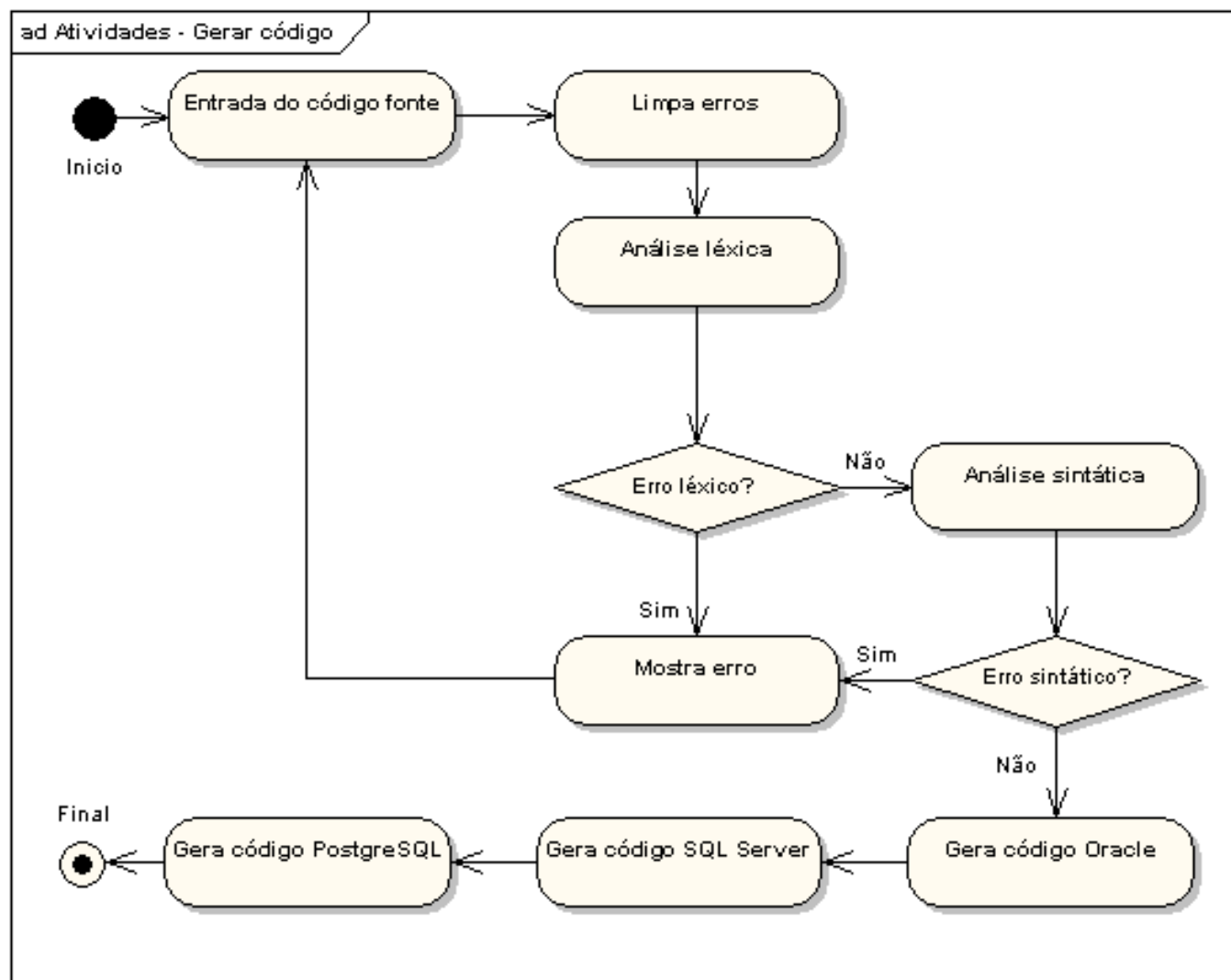
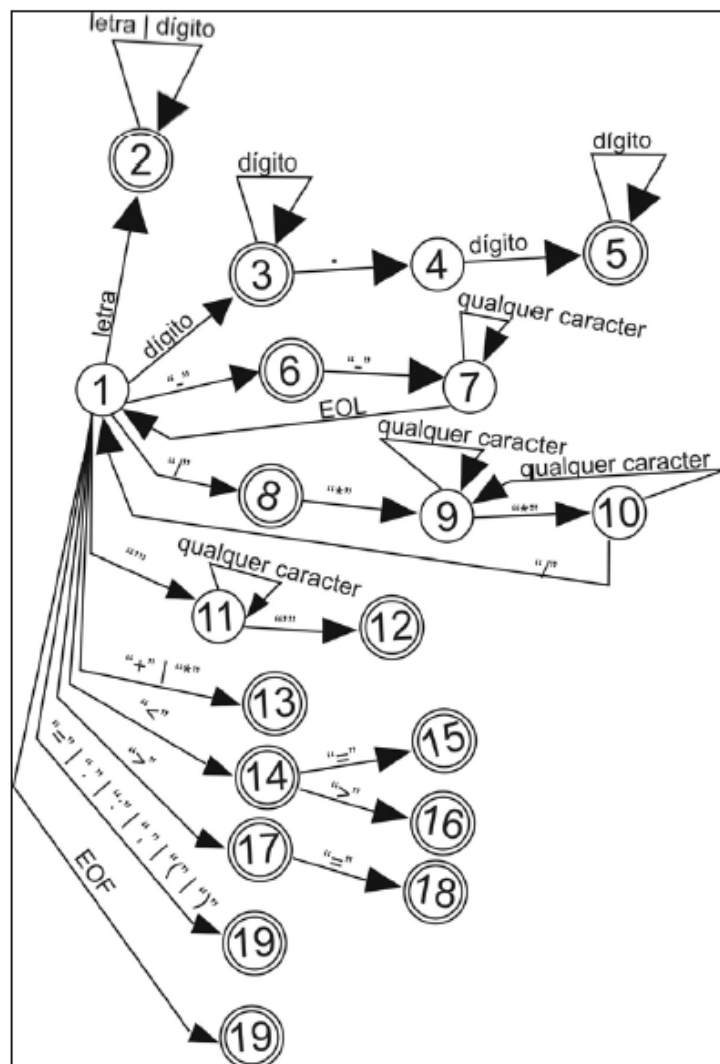




Diagrama de Atividades



Autômato Finito Determinístico





Implementação

Ferramentas utilizadas:

- Delphi 7
- Suíte de componentes SQLDirect 4.2.0
- SGBD Oracle 10g Express Edition
- SGBD SQL Server 2005 Express
- SGBD PostgreSQL 8.1
- GALS 2003.10.03

Operacionalidade



```
Compilador
Arquivo Edição Compilação Sobre
SQL Padrão Oracle SQL Server Postgre SQL
Declare dVlrDesc Decimal(7,2)
Declare dVlrPgto Decimal(10,2)
Declare CURSOR Funcionarios IS select COD_FUNC, SAL_FUNC from FUNC
BEGIN
open Funcionarios
loop
fetch Funcionarios into iCodFunc, dSalFunc
exit when Funcionarios notfound
select VLR_DESP into dVlrDesc from DESP
where COD_FUNC = iCodFunc
and DAT_DESP between to_date(pDatIni) and to_date(pDatPgt)
set dVlrPgto = dSalFunc-dVlrDesc
Linha: 1 Coluna: 8 Arquivo: H:\TCC\TCC II\implementacao2\exemplos\folhapgto.lp
```

Área para entrada de código

Operacionalidade



Compilador

Arquivo Edição Compilação Sobre

SQL Padrão Oracle SQL Server PostgreSQL

```
CREATE PROCEDURE FolhaPgto (pDatIni IN TimeStamp, pDatPgt IN TimeStamp, pQtdFunc IN Integer,
  Declare iQtdFunc Integer
  Declare iCodFunc Integer
  Declare dSalFunc Decimal(10,2)
  Declare dVlrDesc Decimal(7,2)
  Declare dVlrPgto Decimal(10,2)
  Declare CURSOR Funcionarios IS select COD_FUNC, SAL_FUNC from FUNC
BEGIN

  open Funcionarios

  loop
    fetch Funcionarios into iCodFunc, dSalFunc
    exit Funcionarios notfound

    select VLR_DESP into dVlrDesc from DESP
    where COD_FUNC = iCodFunc
```

-> Erro sintático na linha 14, posição 10: Palavra reservada WHEN esperada. Encontrado: "Funcionarios"
-> Programa compilado com erros.

Linha: 14 Coluna: 10 Arquivo: H:\TCC\TCC II\implementacao2\exemplos\folhapgto.lp

Detecção de erros

Operacionalidade



The screenshot shows a window titled "Compilador" with a menu bar (Arquivo, Edição, Compilação, Sobre) and a toolbar. Below the toolbar are tabs for "SQL Padrão", "Oracle", "SQL Server", and "Postgre SQL". The main text area contains the following SQL code:

```
CREATE OR REPLACE FUNCTION FolhaPgto (pDatIni IN TIMESTAMP, pDatPgt IN TIMESTAMP, pQtdFunc
DECLARE
  iQtdFunc NUMERIC;
  iCodFunc NUMERIC;
  dSalFunc DECIMAL(10,2);
  dVlrDesc DECIMAL(7,2);
  dVlrPgto DECIMAL(10,2);
  Funcionarios CURSOR FOR SELECT COD_FUNC, SAL_FUNC FROM FUNC;
BEGIN
  OPEN Funcionarios;
  LOOP
    FETCH Funcionarios INTO iCodFunc, dSalFunc;
    EXIT WHEN Funcionarios%NOTFOUND;
    SELECT VLR_DESP INTO dVlrDesc FROM DESP WHERE COD_FUNC = iCodFunc AND DAT_DESP BETWEEN
    dVlrPgto := dSalFunc-dVlrDesc;
    INSERT INTO FOLHA (COD_FUNC, VLR_LIQD, VLR_DESP, VLR_PGTO) VALUES (iCodFunc, dSalFunc,
```

At the bottom of the window, a status bar indicates "PostgreSQL Desconectado." and a message: "-> Geração de código efetuada com sucesso." The status bar also shows "Linha: 30", "Coluna: 44", and the file path "Arquivo: H:\TCC\TCC II\implementacao2\exemplos\folhapgto.lp".

Conectar ao SGDB

Criar procedimento

Código gerado

Operacionalidade



The screenshot shows a window titled "Compilador" (Compiler) with a menu bar (Arquivo, Edição, Compilação, Sobre) and a toolbar. The main area displays a PL/SQL script for a function named "FolhaPgto". The script includes declarations for variables, a cursor, and a loop for processing employee data. A "Login" dialog box is overlaid on the script, with fields for "Servidor" (localhost), "Usuário" (postgres), and "Senha" (masked). The dialog has "OK" and "Cancel" buttons. Below the script, the status bar indicates "PostgreSQL Desconectado." and a message: "> Geração de código efetuada com sucesso." The bottom status bar shows "Linha: 30 Coluna: 44 Arquivo: H:\TCC\TCC II\implementacao2\exemplos\folhapgto.lp".

```
CREATE OR REPLACE FUNCTION FolhaPgto (pDatIni IN TIMESTAMP, pDatPgt IN TIMESTAMP, pQtdFunc
  DECLARE
    iQtdFunc NUMERIC;
    iCodFunc NUMERIC;
    dSalFunc DECIMAL(10,2);
    dVlrDesc DECIMAL(7,2);
    dVlrPgto DECIMAL(10,2);
    Funcionarios CURSOR FOR SELECT CO
  BEGIN
    OPEN Funcionarios;
    LOOP
      FETCH Funcionarios INTO iCodFunc
      EXIT WHEN Funcionarios%NOTFOUND
      SELECT VLR_DESP INTO dVlrDesc F
      dVlrPgto := dSalFunc-dVlrDesc;
      INSERT INTO FOLHA (COD_FUNC, VLR_LIQD, VLR_DESP, VLR_PGTO) VALUES (iCodFunc, dSalFunc,
```

PostgreSQL Desconectado.

> Geração de código efetuada com sucesso.

Linha: 30 Coluna: 44 Arquivo: H:\TCC\TCC II\implementacao2\exemplos\folhapgto.lp

Operacionalidade



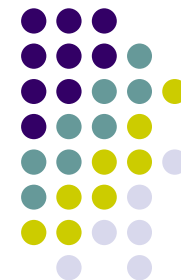
The screenshot shows a window titled "Compilador" (Compiler) with a menu bar (Arquivo, Edição, Compilação, Sobre) and a toolbar. The main area contains a SQL script for a function named "FolhaPgto". The script includes declarations for variables, a cursor named "Funcionarios", and a loop that fetches data from a table named "DESP" and inserts it into a table named "FOLHA". A status bar at the bottom indicates "PostgreSQL Conectado." and "Geração de código efetuada com sucesso." (Code generation completed successfully).

```
CREATE OR REPLACE FUNCTION FolhaPgto (pDatIni IN TIMESTAMP, pDatPgt IN TIMESTAMP, pQtdFunc
DECLARE
iQtdFunc NUMERIC;
iCodFunc NUMERIC;
dSalFunc DECIMAL(10,2);
dVlrDesc DECIMAL(7,2);
dVlrPgto DECIMAL(10,2);
Funcionarios CURSOR FOR SELECT
BEGIN
OPEN Funcionarios;
LOOP
FETCH Funcionarios INTO iCodFunc, dSalFunc,
EXIT WHEN Funcionarios%NOTFOUND;
SELECT VLR_DESP INTO dVlrDesc FROM DESP WHERE COD_FUNC = iCodFunc AND DAT_DESP BETWEEN
dVlrPgto := dSalFunc-dVlrDesc;
INSERT INTO FOLHA (COD_FUNC, VLR_LIQD, VLR_DESP, VLR_PGTO) VALUES (iCodFunc, dSalFunc,
```

Aviso
Comando executado com sucesso.
OK

PostgreSQL Conectado.
-> Geração de código efetuada com sucesso.

Linha: 30 Coluna: 44 Arquivo: H:\TCC\TCC II\implementacao2\exemplos\folhapgto.lp



Problemas e Soluções

Problemas:

- Conexão aos SGBDs abordados utilizando um único componente
- Correção da BNF original da gramática, eliminando fatorações e recursividades

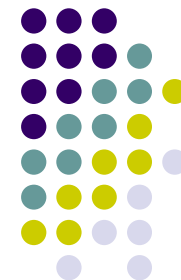
Soluções

- Suíte de componentes SQLDirect
- Utilização da ferramenta GALS para correções, extensão da gramática e geração dos analisadores

Conclusões



- Requisitos atendidos plenamente
- Utilização do GALIS possibilitou simplificar a implementação



Relevância Pessoal

- Prática nos assuntos relacionados as matérias de Linguagens Formais, Compiladores e Banco de Dados
- Possibilidade de desenvolver uma ferramenta utilizando conhecimentos de interesse pessoal
- Obtenção de conhecimento acerca da sintaxe dos comandos do SGBD SQL Server