

**CQL**

**Trabalho Prático de Processamento de Linguagens**

**Grupo 06:**

**23016 – Dani Carvalho da Cruz**

**23279 – Hugo Ferreira Baptista**

**23010 – Hugo Tiago Mendes Cruz**

**Professor: Óscar Ribeiro**

**Licenciatura em Engenharia de Sistemas Informáticos**

Barcelos | maio, 2025

**Índice**

[1. Introdução 3](#_Toc198076102)

[2. Descrição da Linguagem CQL 4](#_Toc198076103)

[3. Especificação da Gramática (BNF) 5](#_Toc198076104)

[4. Reconhecedor Léxico (Lex) 6](#_Toc198076105)

[5. Reconhecedor Sintático (Yacc) 6](#_Toc198076106)

[6. Árvore de Sintaxe Abstrata (AST) 7](#_Toc198076107)

[7. Semântica e Execução 8](#_Toc198076108)

[8. Exemplos de Utilização 8](#_Toc198076109)

[9. Conclusão 9](#_Toc198076110)

[10. Referências 10](#_Toc198076111)

# Introdução

* Objetivo geral do trabalho.
* Importância do tema: análise léxica, sintática, semântica.
* Breve descrição do problema e da solução.

# Descrição da Linguagem CQL

Explicação da CQL

Tipos de comandos

(importação, queries, joins, procedimentos). Exemplos simples.

# Especificação da Gramática (BNF)

**G = <T, N, S, P>**

**Terminais (T):** "IMPORT", "TABLE", "FROM", "AS", "DISCARD", "RENAME", "PRINT", "SELECT", "WHERE", "LIMIT", "CREATE", "JOIN", "USING", "DO", "END", "CALL", "DELETE", '\*', ',', ';', '(', ')', id, file, num, numdec, string, operador

**Não Terminais (N):** PROG, CMDLIST, CMD, CONF, QRS, SELEC, COLLIST, CONDLIST, COND, OPERADOR, VALOR, NEW, PROCS

**Axioma (S): PROG**

P1: PROG → CMDLIST

P2: CMDLIST → CMD

P3: | CMD CMDLIST

P4: CMD → CONF

P5: | QRS

P6: | NEW

P7: | PROCS

P8: CONF → "IMPORT" "TABLE" id "FROM" file ';'

P9: | "EXPORT" "TABLE" id "AS" file ';'

P10: | "DISCARD" "TABLE" id ';'

P11: | "RENAME" "TABLE" id id ';'

P12: | "PRINT" "TABLE" id ';'

P13: QRS → "SELECT" SELEC "FROM" id ';'

P14: | "SELECT" SELEC "FROM" id "WHERE" CONDLIST ';'

P15: | "SELECT" SELEC "FROM" id "LIMIT" num ';'

P16: | "SELECT" SELEC "FROM" id "WHERE" CONDLIST "LIMIT" num ';'

P17: SELEC → '\*'

P18: | COLLIST

P19: COLLIST → id

P20: | id ',' COLLIST

P21: CONDLIST → COND "AND" CONDLIST

P22: | COND

P23: COND → id OPERADOR VALOR

P24: OPERADOR → operador

P25: VALOR → numdec

P26: | string

P27: | num

P28: NEW → "CREATE" "TABLE" id QRS

P29: | "CREATE" "TABLE" id "FROM" id "JOIN" id "USING" '('id')' ';'

P30: PROCS → "PROCEDURE" id "DO" CMDLIST "END"

P31: | "CALL" id ';'

P32: | "DELETE" id ';'

CMDLIST: permite ter apenas um comando ou vários.

id: identificador de tabelas colunas e procedures. Faz a ponte para as palavras reservadas.

file: arquivo de entrada ou saída (ex: “observacoes.csv”, “data/observacoes.csv” ).

VALOR: permite ter valores decimais inteiros ou strings.

COND: Condição usada para os filtros

CONDLIST: permite ter uma coluna ou varias

# Reconhecedor Léxico (Lex)

**Como foi definido o lexer (tokens principais).**

* Exemplos de tokens: IMPORT, SELECT, num, id, string, comentários, etc.
* Desafios (ex: leitura de strings com vírgulas ou aspas).

# Reconhecedor Sintático (Yacc)

# Árvore de Sintaxe Abstrata (AST)

* Objetivo: representar comandos ignorando açúcar sintático.
* Estrutura de representação:{'op': 'SELECT', 'args': [...]} etc. Exemplos concretos: Um SELECT simples Um SELECT com WHERE + LIMIT
* Um procedimento Comenta como foi pensada a estrutura seq para comandos múltiplos.

# Semântica e Execução

* Como cada comando AST é executado: IMPORT, EXPORT, SELECT, JOIN, etc. Como se faz JOIN de duas tabelas.
* Como se aplicam filtros WHERE e limites.
* Como são armazenadas as tabelas em memória (dicionário)

# Exemplos de Utilização

* 2 a 3 trechos de código .cql.
* AST gerada.
* Resultado produzido no terminal.
* Breves comentários sobre cada um.

# Conclusão

# Referências