

Gramática criada

A gramática foi definida com base na linguagem Fortall, respeitando sua estrutura de declarações.

Foi criado um parser dividido em 3 partes (lexico, sintático e semântico)

Léxico:

Tokens principais:

- **Palavras-chave:** se, senao, enquanto, retorno, etc.
- **Identificadores:** var, x, resultado (como em C)
- **Números decimais:** 42, 0, -17
- **Cadeias de caracteres:** "texto\n", "com\taba"
- **Operadores:** +, -, *, /, ==, !=, <, <=, etc.
- **Comentários:** /* ... */ (sem aninhamento)

Sintático:

```
<programa> ::= <decls> <stmts>
<decls> ::= <decl> <decls> | ε
<decl> ::= tipo id ;
<stmts> ::= <stmt> <stmts> | ε
<stmt> ::= 'se' '(' <expr> ')' <stmt> ['senao' <stmt>] | ...
```

Semântico:

Verificações:

- Tipagem de variáveis
- Uso de variáveis não declaradas
- Tipos corretos em se e enquanto (inteiros)
- Coerência entre os tipos nas expressões
-

Passos para instalação:

Foi usado na implementação Python + PLY (Lex/Yacc), então será necessário ter no mínimo o Python 3 e a biblioteca PLY instalada.

Para instalar a PLY, basta usar no terminal: **pip install ply**

Na pasta principal existem 4 arquivos principais sendo: parser, interpreter, lexer e main. E 5 arquivos de exemplos quais podem serem usados no main.

Implementação:

- A implementação do analisador léxico foi feita com a biblioteca PLY (Python Lex-Yacc), utilizando o módulo lex.
- Tratamento de erro léxico foi incluída uma função t_error para reportar tokens inválidos, exibindo a linha e o caractere inesperado.
- O parser também foi construído com a biblioteca PLY, utilizando o módulo yacc.
- O parser constrói uma árvore sintática abstrata (AST) para representar o código de forma estruturada
- O tratamento de erro sintático é feito com mensagens informando o token inesperado e a linha do erro
- O analisador semântico foi implementado de forma implícita durante a execução
- Foram desenvolvidos 5 arquivos .ft de exemplo, testando: Atribuições simples, expressões com operadores variados, estruturas de controle se, senão, enquanto.