

1. Erros Léxicos

Os erros léxicos são identificados pelo Compilador.jj (tokens ERRO_...) e formatados pela classe ErrorHandler.java (método processLexicalError).

1.1 Formato da Mensagem

O formato padrão para um erro léxico é: Erro Léxico na linha [X], coluna [Y]: [Descrição Específica]

1.2 Mensagens Específicas

As descrições específicas são preenchidas no Map LEXICAL_ERROR_MESSAGES do ErrorHandler.java. Com base nos tokens definidos no Compilador.jj, as mensagens correspondentes são:

- **ERRO_ID_INICIA_COM_DIGITO**
 - **Mensagem:** "Identificador inválido: não pode começar com um dígito."
- **ERRO_ID_DIGITOS_CONSECUTIVOS**
 - **Mensagem:** "Identificador inválido: não pode conter dois ou mais dígitos consecutivos."
- **ERRO_ID_TERMINA_COM_DIGITO**
 - **Mensagem:** "Identificador inválido: não pode terminar com um dígito."
- **ERRO_INT_LONGO**
 - **Mensagem:** "Constante numérica inteira muito longa."
- **ERRO_REAL_FRACAO_LONGA**
 - **Mensagem:** "Constante real com parte fracionária muito longa."
- **ERRO_REAL_INTEIRO_LONGO**
 - **Mensagem:** "Constante real com parte inteira muito longa."
- **ERRO_REAL_INCOMPLETO**
 - **Mensagem:** "Constante real malformada (faltam dígitos após o ponto)."

1.3 Erro Léxico Genérico

Se um token de erro for encontrado e não estiver no Map acima, ou se for um token ERRO_LEXICO (símbolo desconhecido), ele usará a mensagem de fallback do ErrorHandler.java:

- **ERRO_LEXICO (Catch-all)**

- **Mensagem:** Erro Léxico na linha [X], coluna [Y]: Símbolo não reconhecido '[símbolo]'
 - **Exemplo:** Erro Léxico na linha 5, coluna 1: Símbolo não reconhecido '@'
-

2. Erros Sintáticos

Os erros sintáticos são capturados pelos blocos try...catch no Compilador.jj e formatados pelo método processParseException no ErrorHandler.java.

2.1 Formato da Mensagem

O formato padrão para um erro sintático é: Erro Sintático na linha [X], coluna [Y] ([contexto]): Encontrado [token_encontrado], esperado: [tokens_esperados].

2.2 Contextos de Erro

O [contexto] é definido diretamente nos blocos catch do Compilador.jj:

- "na estrutura principal do programa" (Disparado pela acaoCompilar ao tentar chamar parser.programa())
- "em declaração de variáveis" (Disparado pela regra declaracao_de_variaveis)
- "em comando" (Disparado pela regra lista_de_comandos)
- "em comando 'set'" (Disparado pela regra atribuicao)
- "em comando 'read'" (Disparado pela regra entrada)
- "em comando 'show'" (Disparado pela regra saida)
- "em comando 'if'" (Disparado pela regra selecao)
- "em comando 'loop'" (Disparado pela regra repeticao)

2.3 Mensagens de Tokens Esperados (Gatilhos Especiais)

O ErrorHandler.java possui uma lógica (no getExpectedTokensDescription) para "embelezar" a lista de tokens esperados.

1. **Tradução Padrão:**

- <IDENTIFIER> é traduzido para "um identificador"
- <CONST_INT> é traduzido para "um número inteiro"
- <CONST_REAL> é traduzido para "um número real"
- <CONST_LITERAL> é traduzido para "um texto"
- <EOF> é traduzido para "o final do arquivo"

2. **Gatilho 1: Esperando um Tipo**

- **Se o parser esperar:** num, real, text ou flag
- **A mensagem será:** esperado: tipo do identificador.

3. **Gatilho 2: Esperando Fim da Declaração ou Inicialização**

- **Se o parser esperar:** ;, = ou [
- **A mensagem será:** esperado: ';' ou inicialização de identificador.

4. **Gatilho 3: Esperando Fim de Comando ou Expressão**

- **Se o parser esperar:** Um conjunto grande de operadores (como !=, %, *, +, ,, ;, ==, etc.)
- **A mensagem será:** esperado: ';' , ')' ou expressão.

5. **Fallback (Padrão):**

- Se nenhum gatilho for ativado, ele lista os tokens esperados.
- **Exemplo:** esperado: 'if', 'loop' ou 'set'.

3. Erros Semânticos

Os erros semânticos são definidos diretamente nas regras do Compilador.jj e usam o método addError do ErrorHandler.java.

3.1 Formato da Mensagem

O formato padrão para um erro semântico é: Erro Semântico (linha [X], coluna [Y]): [Descrição Específica]

3.2 Mensagens Específicas

Abaixo estão **todas** as mensagens de erro semântico que o seu compilador pode gerar, extraídas diretamente do Compilador.jj:

- **Durante a Declaração:**

- "Identificador '[nome_do_id]' já declarado."
 - (Disparado pelas regras lista_de_identificadores e lista_de_identificadores_linha)
- "Tamanho do vetor deve ser maior que 0."
 - (Disparado pela regra constante_inteira)
- **Durante Comandos (set, read, expressões):**
 - "Identificador '[nome_do_id]' não declarado."
 - (Disparado por atribuicao, entrada, elemento)
 - "Identificador de vetor '[nome_do_vetor]' deve ser indexado."
 - (Disparado por atribuicao, entrada, elemento quando se usa um vetor sem [...])
 - "Identificador escalar '[nome_do_escalar]' não pode ser indexado."
 - (Disparado por atribuicao, entrada, elemento quando se usa um escalar com [...])