



Problema Can't load AnalizadorParser as lexer or parser

```
217 | expresion EQ expresion
218 | expresion NEQ expresion
219 | expresion LT expresion
220 | expresion GT expresion
221 | expresion LE expresion

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

[283,2586:2588='end', '<'end'>',99:0]
[284,2589:2588='<EOF>', '<EOF>',99:3]
red@red-Latitude-3150:~/Escritorio/DDDDD$ antlr4 AnalizadorLexer.g4 AnalizadorParser.g4
red@red-Latitude-3150:~/Escritorio/DDDDD$ antlr4 -listener -visitor AnalizadorParser.g4
red@red-Latitude-3150:~/Escritorio/DDDDD$ javac AnalizadorLexer*.java AnalizadorParser*.java
red@red-Latitude-3150:~/Escritorio/DDDDD$ grun AnalizadorParser programa -tree < entrada1.hy
Can't load AnalizadorParser as lexer or parser
red@red-Latitude-3150:~/Escritorio/DDDDD$
```

Algunas respuestas en foros:

oh, resolví ese problema usando javac -g *.java en ese directorio

– [usuario3162707](#)

[Comentó28/04/2014 a las 15:11](#)

Obtuve el mismo problema, lo resolví agregando . a classpath. (Ya compilé archivos java)

– [JaDogg](#)

[Comentó16/08/2015 a las 13:25](#)

Pruebas

```
Can't load AnalizadorParser as lexer or parser
josed@josed-VMware-Virtual-Platform:~/Documentos/Analizador$ javac -g.*java
error: invalid flag: -g.*java
Usage: javac <options> <source files>
use --help for a list of possible options
```

```
only showing the first 100 errors; of 192 total; use --stacktrace if you would like to see more
josed@josed-VMware-Virtual-Platform:~/Documentos/HYDRA_ANTLR$ grun AnalizadorParser entrada1 -tree < entrada1.hy
Exception in thread "main" java.lang.ClassCastException: class AnalizadorParser
    at java.base/java.lang.Class.asSubclass(Class.java:4102)
    at org.antlr.v4.gui.TestRig.process(TestRig.java:135)
    at org.antlr.v4.gui.TestRig.main(TestRig.java:119)
```

`java.lang.ClassNotFoundException` se activa si `ClassLoader` no puede encontrar la clase en su sistema. En la biblioteca principal de JVM (Java Virtual Machine), `ClassLoader` se usa para cargar y ubicar una clase.

Este error lo arroja ClassLoader si no puede cargar una clase de la biblioteca de la aplicación.

Además, debe tener en cuenta la naturaleza comprobada de esta excepción y la necesidad de manejarla adecuadamente al llamar a métodos que pueden desencadenar la excepción `java.lang.ClassNotFoundException` en Java, ya sea a través de un bloque `try-catch` o lanza condición.

Solución

No dividir el analizador en un lexer y un parser, si no todo junto.

The screenshot shows the Visual Studio Code interface with the following components:

- Explorer Sidebar:** Displays the project structure under the "ANALIZADOR" folder. Files include:
 - antlr
 - entrada1.hy
 - Hydra.g4
 - Hydra.interp
 - Hydra.tokens
 - HydraBaseListener.class
 - HydraBaseListener.java
 - HydraLexer.class
 - HydraLexer.interp
 - HydraLexer.java
 - HydraLexer.tokens
 - HydraListener.class
 - HydraListener.java
 - HydraParser.class
 - HydraParser.java
 - HydraParser\$AsignacionContext.class
 - HydraParser\$AtomContext.class
 - HydraParser\$BloqueContext.class
 - HydraParser\$CaseLockContext.class
 - HydraParser\$CuerpoContext.class
 - HydraParser\$CuerpoGeneralContext.class
 - HydraParser\$DeclaracionContext.class
 - HydraParser\$DeclaratorContext.class
 - HydraParser\$DeclaratorListContext.class
 - HydraParser\$DefaultBlockContext.class
 - HydraParser\$DefinicionClaseContext.class
 - HydraParser\$DoWhileStatementContext...
 - HydraParser\$EmpleadoCuerpoContext.class
 - HydraParser\$ExencionContext.class
 - HydraParser\$ExencionListContext.class
 - HydraParser\$FuncionContext.class
 - HydraParser\$IfStatementContext.class
 - HydraParser\$ImportListContext.class
 - HydraParser\$ImportStatementContext.class
- Main Editor Window:** Shows the content of "entrada1.hy". It contains ANTLR grammar rules and Java code.

```
183 AND: '&&';  
184 OR: '|';  
185 NOT: '!';  
186  
187 // --- Simbolos ---  
188 LPAREN: '(';  
189 RPAREN: ')';  
190 LBRACE: '{';  
191 RBRACE: '}';  
192 LBACKET: '[';  
193 RBACKET: ']';  
194 SEMICOLON: ';';  
195 COLON: ':';  
196 COMMA: ',';  
197 DOT: '.';  
198  
199 // --- Identificadores ---  
200 CONSTID: [A-Z] [A-Z0-9]*;  
201 IDENTIFIER: [a-zA-Z] [a-zA-Z0-9]*;  
202  
203 // --- Ignorar Tokens ---  
204 COMMENT: '#' ~[\r\n]* -> skip;  
205 WS: [\t\r\n]+ -> skip;
```
- Terminal Panel:** Shows the command being executed in the terminal:

```
ls -l /usr/bin/hydrasim.jar | xargs java -jar
```

Luego ejecutar los siguientes comando:

antlr4 Hydra.g4

```
javac Hydra*.java
```

```
grun Hydra programa -tree < entrada1.hy
```