



Teoría de Lenguajes de Computación

Profesor: Luis Ramiro Basto Díaz

ADA 05: Identificadores y Variables

Fecha de Entrega: 24 de Febrero de 2022

Integrantes:

Nombre	Matricula
Emanuel Rafael Chávez Bagundo	17003636
Karyme Maylin Bautista Poot	17003222

Teoría de Lenguajes de Programación LIS

PROYECTO INTEGRADOR TLP 1 DOCUMENTACIÓN:



Primero descargamos el .jar de ANTLR, luego creamos los archivos por lote (bat) para poder ejecutar de manera más fácil en la terminal

Y los agregamos a la PATH de nuestras computadoras para poder trabajar la termina

El archivo antlr4.bat contiene lo siguiente:

```
java org.antlr.v4.Tool %*
```

El archivo grun.bat contiene lo siguiente:

```
java org.antlr.v4.gui.TestRig %*
```

Luego generamos un archivo .txt con la gramática que usamos, la cual básicamente era muy sencilla, lo único era verificar que no fuera recursiva por la izquierda, ya que eso generaba un problema al compilar

A screenshot of a Notepad window titled 'Expr.g4: Bloc de notas'. The window contains the following text:

```
grammar Expr;

prog: (expr NEWLINE)* ;

expr: expr ('*' | '/') expr
    | expr ('+' | '-') expr
    | INT
    | '(' expr ')'
    ;

NEWLINE : [\r\n]+ ;
INT      : [0-9]+ ;
```

Posterior a esto lo convertimos a .g4.

En la terminal utilizamos los siguientes comandos para generar un árbol de parseo, como se observaba en el ejemplo que usamos de base en la página <https://www.antlr.org/>.

```
C:\antlr\bin>antlr4 Expr.g4

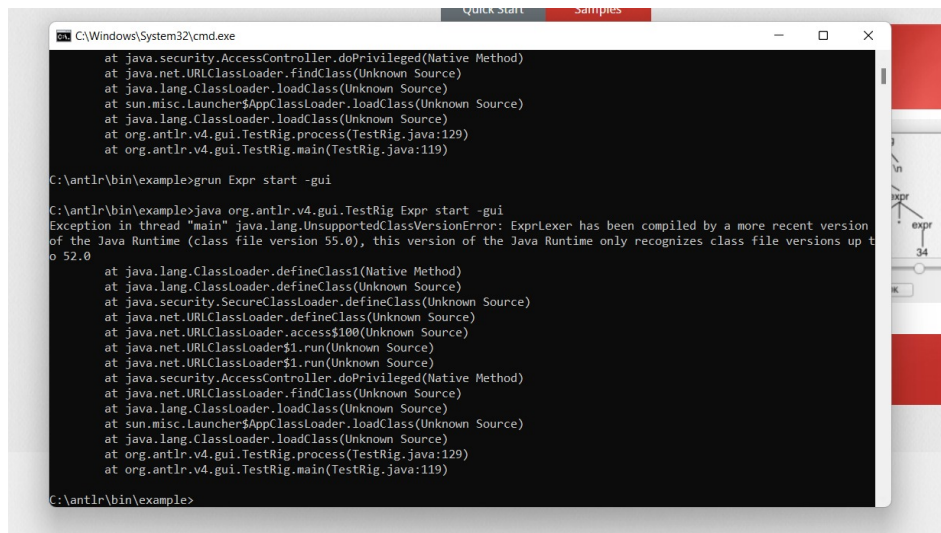
C:\antlr\bin>java org.antlr.v4.Tool Expr.g4

C:\antlr\bin>javac *.java

C:\antlr\bin>grun Expr prog -gui

C:\antlr\bin>java org.antlr.v4.gui.TestRig Expr prog -gui
100+2
^Z
```

Sin embargo, el instalar el programa nos resultó demasiado complicado, ya que al intentar compilar el código, sucedían demasiados errores distintos, como el siguiente:



```
C:\Windows\System32\cmd.exe
at java.security.AccessController.doPrivileged(Native Method)
at java.net.URLClassLoader.findClass(Unknown Source)
at java.lang.ClassLoader.loadClass(Unknown Source)
at sun.misc.Launcher$AppClassLoader.loadClass(Unknown Source)
at java.lang.ClassLoader.loadClass(Unknown Source)
at org.antlr.v4.gui.TestRig.process(TestRig.java:129)
at org.antlr.v4.gui.TestRig.main(TestRig.java:119)

C:\antlr\bin\example>grun Expr start -gui

C:\antlr\bin\example>java org.antlr.v4.gui.TestRig Expr start -gui
Exception in thread "main" java.lang.UnsupportedClassVersionError: Exprlexer has been compiled by a more recent version
of the Java Runtime (class file version 55.0), this version of the Java Runtime only recognizes class file versions up t
o 52.0
at java.lang.ClassLoader.defineClass1(Native Method)
at java.lang.ClassLoader.defineClass(Unknown Source)
at java.security.SecureClassLoader.defineClass(Unknown Source)
at java.net.URLClassLoader.defineClass(Unknown Source)
at java.net.URLClassLoader.access$100(Unknown Source)
at java.net.URLClassLoader$1.run(Unknown Source)
at java.net.URLClassLoader$1.run(Unknown Source)
at java.security.AccessController.doPrivileged(Native Method)
at java.net.URLClassLoader.findClass(Unknown Source)
at java.lang.ClassLoader.loadClass(Unknown Source)
at sun.misc.Launcher$AppClassLoader.loadClass(Unknown Source)
at java.lang.ClassLoader.loadClass(Unknown Source)
at org.antlr.v4.gui.TestRig.process(TestRig.java:129)
at org.antlr.v4.gui.TestRig.main(TestRig.java:119)

C:\antlr\bin\example>
```

Nos fue demasiado complicado instalar el programa, finalmente nos dimos cuenta que el problema fue que ANTLR usaba java 8.0 y el programa necesita una versión avanzada de java, así que instalamos jdk 17.0 y de paso, desinstalamos java 8.0 para evitar que el programa lo lea por accidente. Incluso consideramos empezar a usar PowerShell en lugar del Command prompt. Esto funcionó y al final logramos que el programa se compilará correctamente.