ITCR – IC5701.COMPILADORES E INTÉRPRETES

TAREA #1 - HERRAMIENTA: ANTLR

Estudiantes:
Antonio Fernández García
Josué Mena González
Marco Rodríguez Vargas

Profesor:
Emmanuel Ramírez S.

II Semestre 2024

ANTLR

- **Descripción**

ANTLR (ANother Tool for Language Recognition) es una herramienta que se utiliza para generar analizadores (parsers) a partir de gramáticas definidas en un formato específico. Su principal uso es en el desarrollo de lenguajes de programación, traductores de lenguajes, y en la creación de herramientas de análisis de código. Fue desarrollada por Terence Parr, un profesor de Ciencias de la Computación en la Universidad de San Francisco, esta herramienta es Open Source bajo la licencia BSD, actualmente sigue siendo mantenida y actualizada, principalmente por Parr y la comunidad, su última versión es la 4.13.2

ANTLR

- **Enlaces de Descarga**

ANTLR puede ser descargada desde su sitio oficial en
https://www.antlr.org/download.html

Así mismo y al ser Open Source, su código está disponible en
https://github.com/antlr/antlr4/tree/master

Para descargar versiones anteriores se pueden conseguir en
https://github.com/antlr/website-antlr4/tree/gh-pages/download

ANTLR

Lenguajes

Entre los lenguajes con los que se puede integrar se tienen los siguientes:

- *Java*
- *C#*
- *JS*
- *TS*
- *Go*
- *C++*
- *PHP*

ANTLR

- **Etapas de generación de procesadores de lenguaje**

ANTLR genera un parseador (analizador sintáctico) específicamente y también puede generar un analizador sintáctico en conjunto. Su funcionalidad es muy útil ya que ayuda a construir de forma automática el AST. Por otra parte, también implementa los "tree walkers" que se usan para recorrer el AST y visitar sus nodos.

El algoritmo utilizado es el conocido LL(*) o LL(K) (visto en clase) para generar el analizador sintáctico. Sus funcionalidades le permiten transformarse en un intérprete también.

ANTLR

- **Cómo se utiliza?**

La utilidad de esta herramienta se basa en dos puntos importantes:

- *Generar el lexer y parser para procesar gramáticas.*
- *Generar el árbol de parseo para una entrada específica.*

Con esta herramienta es bastante sencillo generar el código para procesar las gramáticas y posteriormente pasarle entradas que generan un árbol de forma visual con la interfaz gráfica. La interfaz gráfica para visualizar el árbol viene con la instalación de una vez.

Para lograr esto basta con simplemente escribir una gramática en un archivo.g4

Posteriormente se ejecuta el comando "antlr4 archivo.g4" para generar el código del Lexer y Parser.

Si se quiere probar algún ejemplo se deben compilar los archivos y ejecutar "grun archivo primeraregla -gui"

ANTLR

- **Compatibilidad con otras herramientas**
  - ANTLR es una herramienta muy compatible con otros lenguajes y plataformas, como Java (utilizado en este ejemplo).
  - Al trabajar con el JDK, ANTLR permite generar analizadores léxicos y sintácticos que se pueden integrar en proyectos de lenguajes como como Java, C#, JavaScript, TypeScript, Go, C++, y PHP.
  - ANTLR depende de Java para su ejecución y se puede integrar fácilmente en flujos de trabajo automatizados con herramientas como Maven o Gradle, lo que facilita la generación y compilación de los analizadores de manera automática.
  - También permite la visualización de árboles sintácticos mediante una interfaz gráfica en Java, lo que lo hace muy útil en entornos de desarrollo.

ANTLR

- **Ejemplo. Crear un archivo Expr.g4 con el siguiente contenido.**

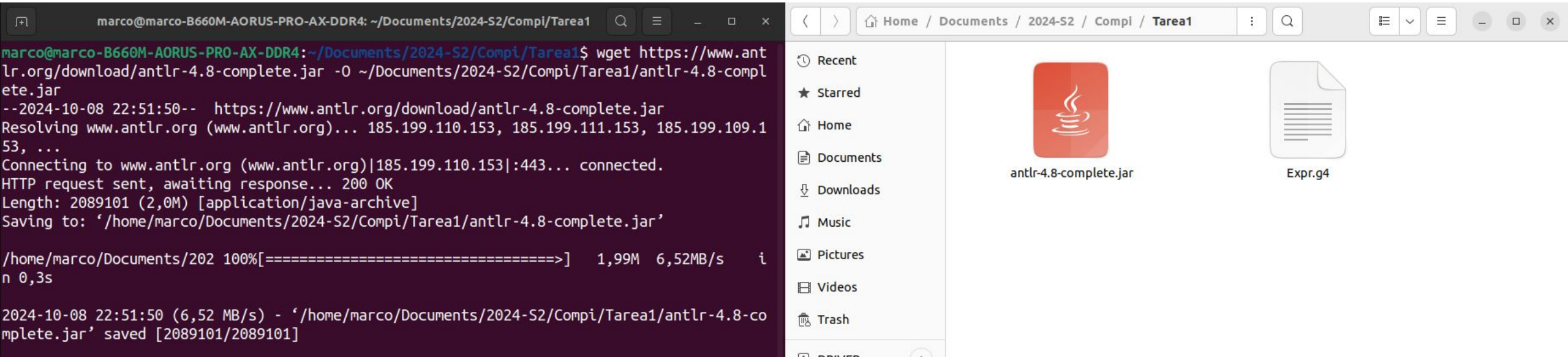
Open

Expr.g4
~/Documents/2024-S2/Compi/Tarea1

```
1 grammar Expr;
2
3 prog:    expr EOF ;
4 expr:    expr ('*'|'/') expr
5      |   expr ('+'|'-') expr
6      |   INT
7      |   '(' expr ')'
8      ;
9
10 NEWLINE : [\r\n]+ -> skip;
11 INT     : [0-9]+ ;
```

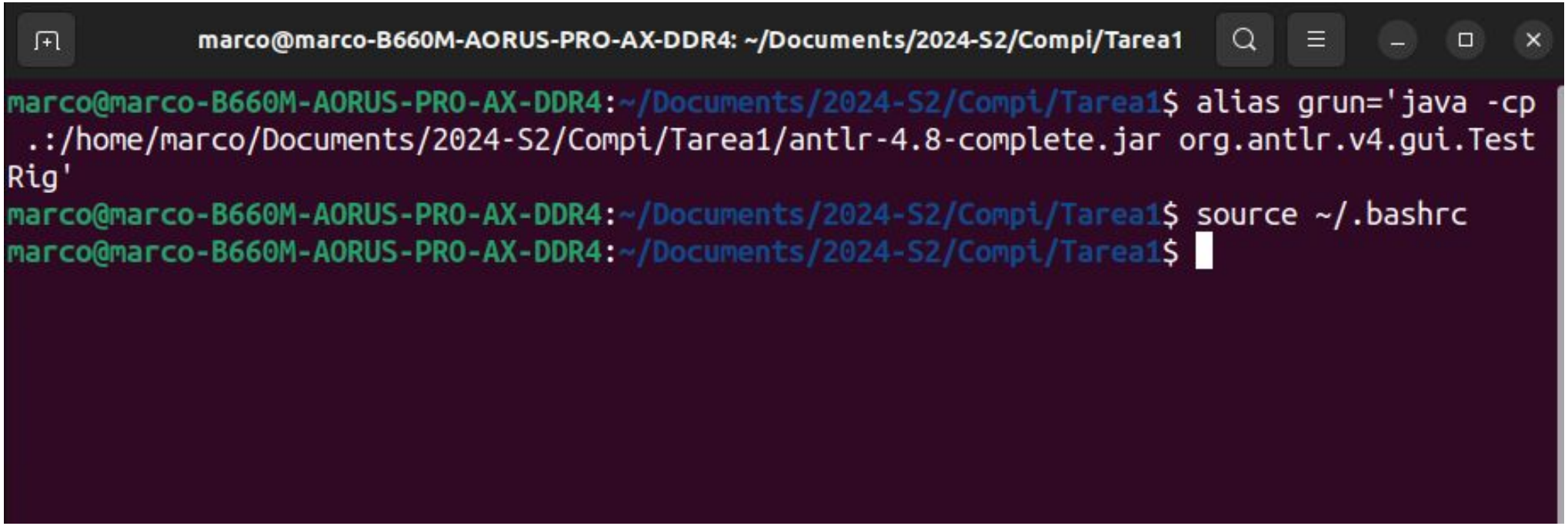
ANTLR

- **Ejemplo. Este comando descarga el archivo antlr-4.8-complete.jar en el directorio ~/Documents/2024-S2/Compi/Tarea1.**



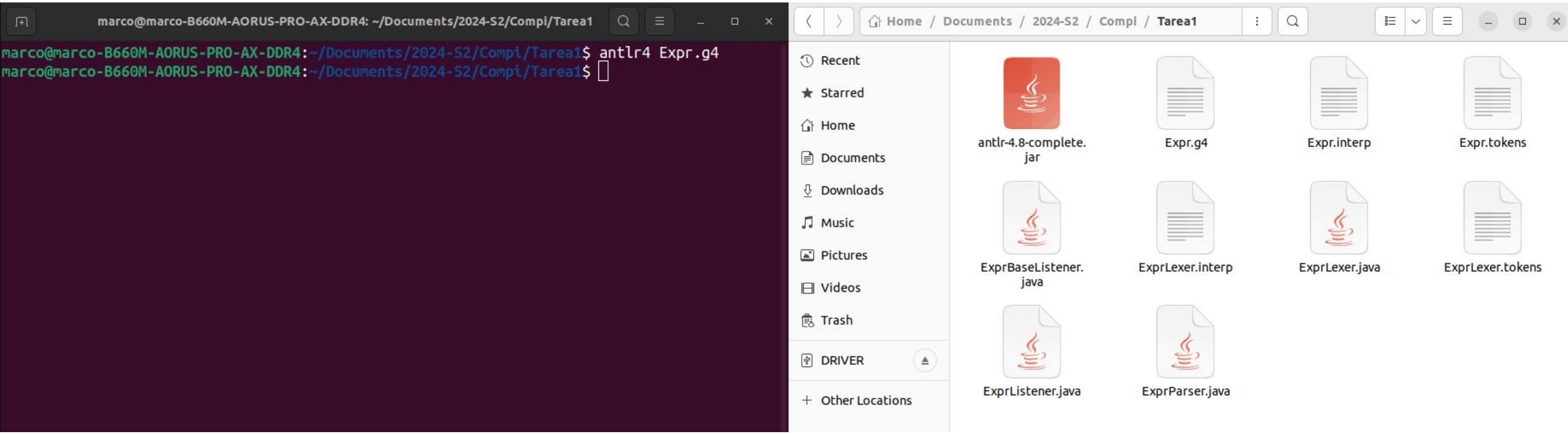
ANTLR

- **Ejemplo. Configurar con un alias para facilitar la ejecución de antlr4 y grun.**



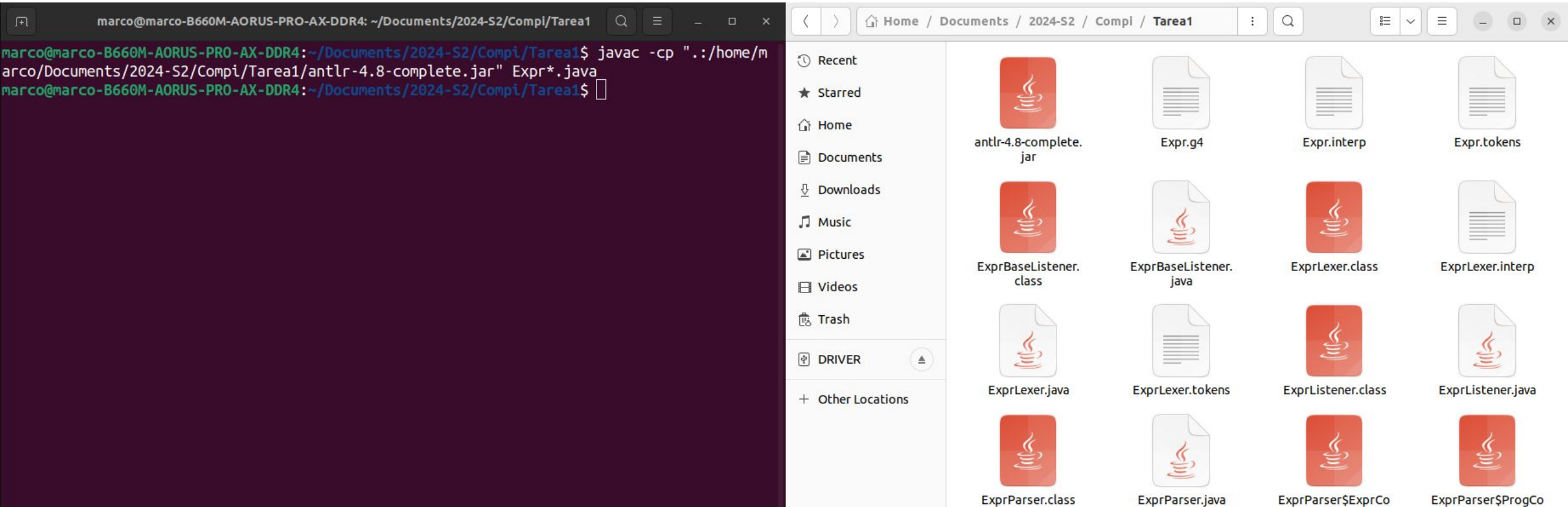
ANTLR

- **Ejemplo. Este comando genera los archivos .java necesarios (como ExprLexer.java y ExprParser.java) a partir del archivo de gramática Expr.g4.**



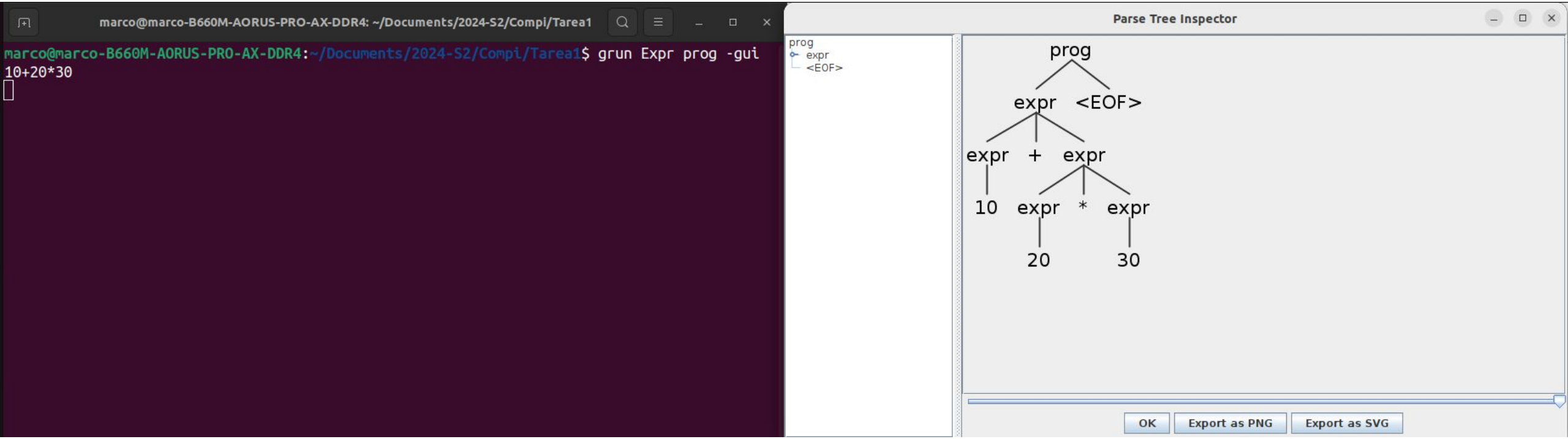
ANTLR

- **Ejemplo. Este comando compila los archivos .java generados por ANTLR utilizando el archivo JAR de ANTLR.**



ANTLR

- **Ejemplo. Después de ejecutar el comando, ingresa una expresión matemática (por ejemplo, 10+20*30), presiona Enter, y luego Ctrl + D para indicar el final de la entrada.**



ANTLR

- **Bibliografía**

https://www.antlr.org/
https://github.com/antlr/antlr4/blob/master/doc/index.md