EXREGAN USAC

MANUAL TÉCNICO

Aldo SAÚL Vásquez MOREIRA

2023

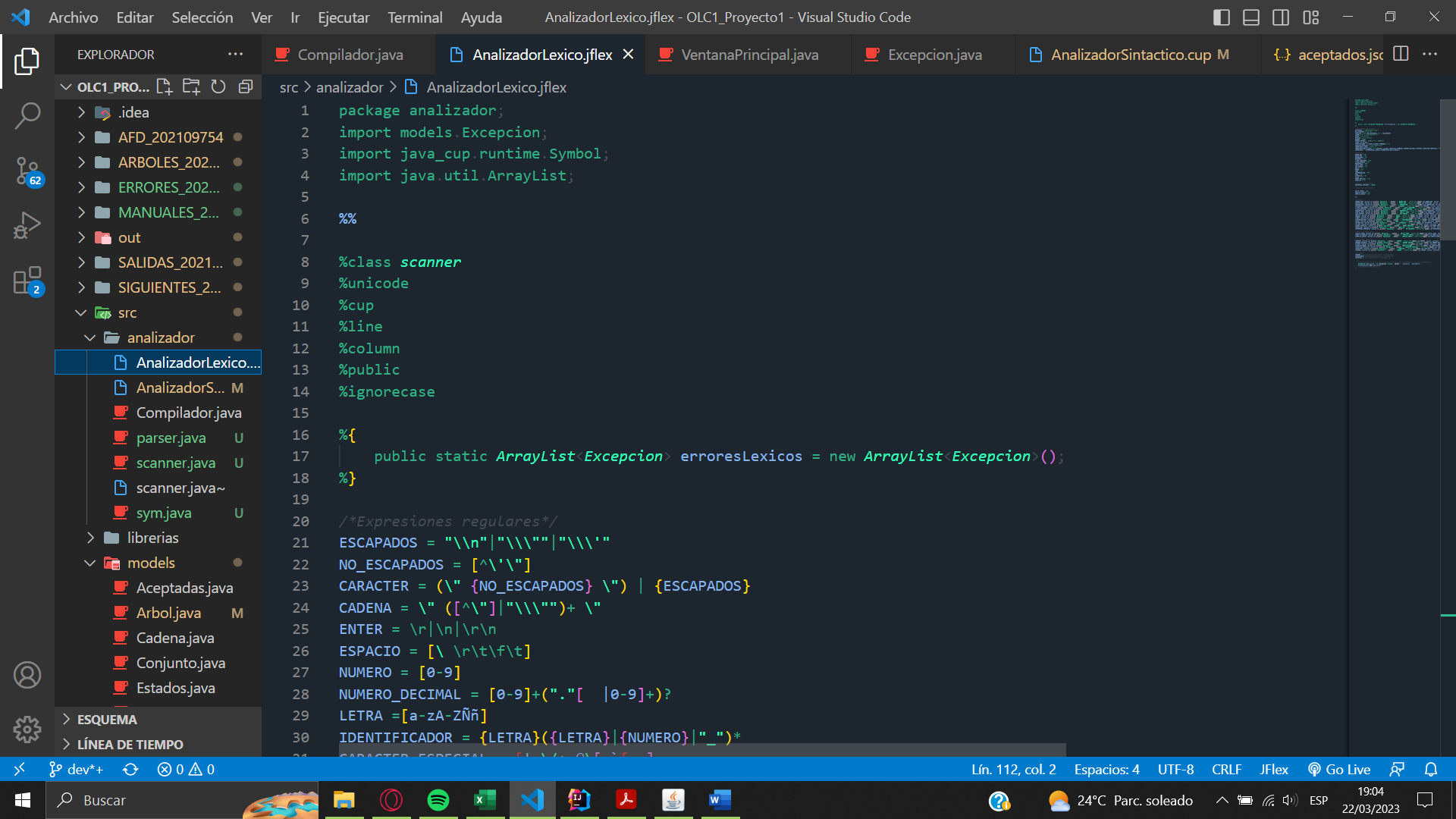
Para el desarrollo de este software se utilizó:

* Lenguaje de programación: JAVA JDK 8
* IDE: IntelliJ IDEA
* Paradigma de programación: Programación Orientada a Objetos
* Interfaz gráfica: JAVA Swing
* Generador de Lexer: JFlex
* Generador de Parser: CUP

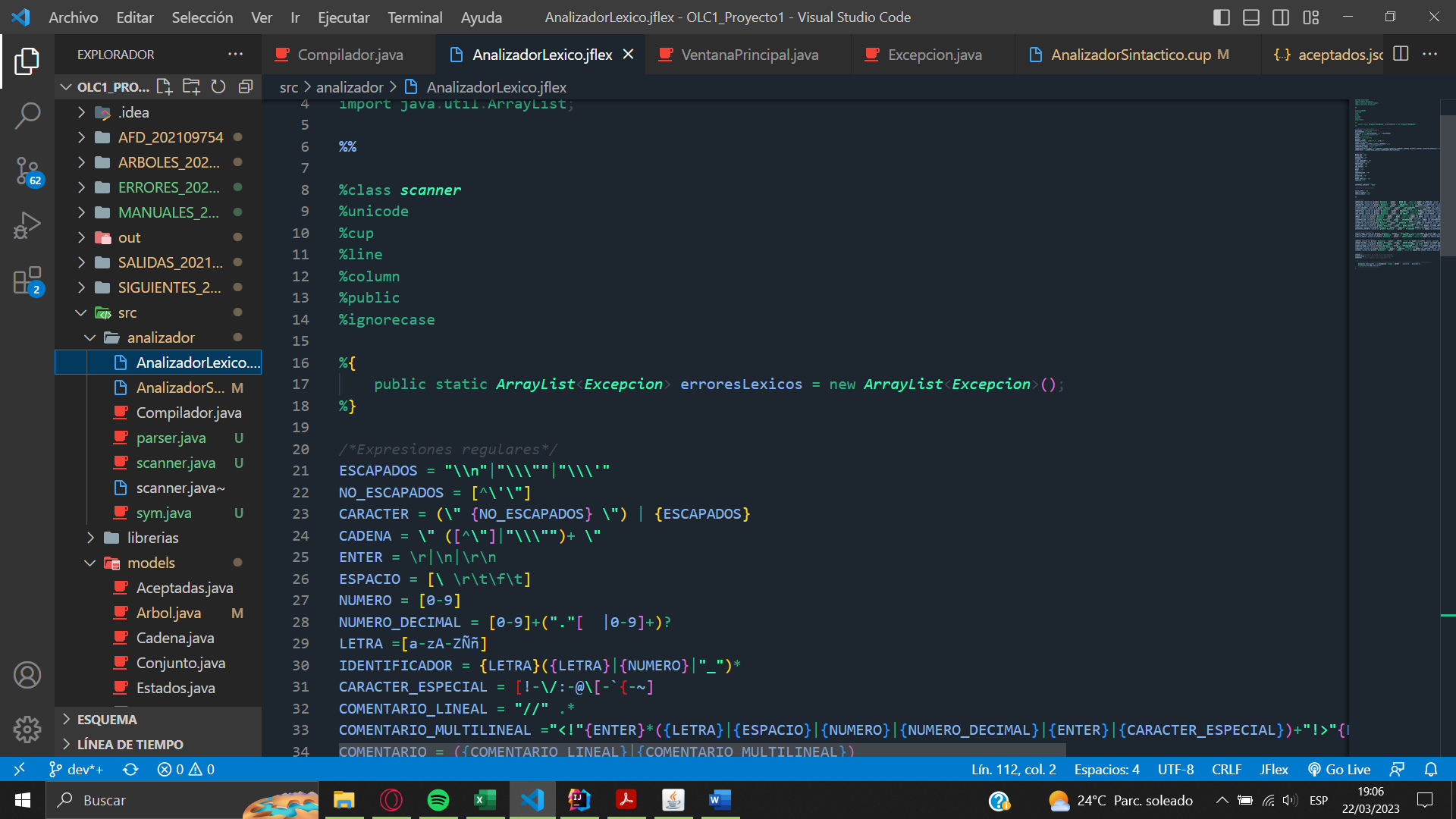
Sin embargo, en este manual nos centraremos en las clases principales así como en las herramientas utilizadas para generar el compilador, dado que las demás herramientas son más conocidas.

Inicialmente, para el análizador léxico tenemos el archivo “AnalizadorLexico.jflex” en el cual se definieron los tokens que se envían al analizador sintáctico o parser.

Cuenta con la siguiente configuración, la cual indica el nombre del archivo, que tomará en cuenta las líneas y columnas y que no será sensible a las mayúsculas y minúsculas, ya que así lo requiere el lenguaje definido.

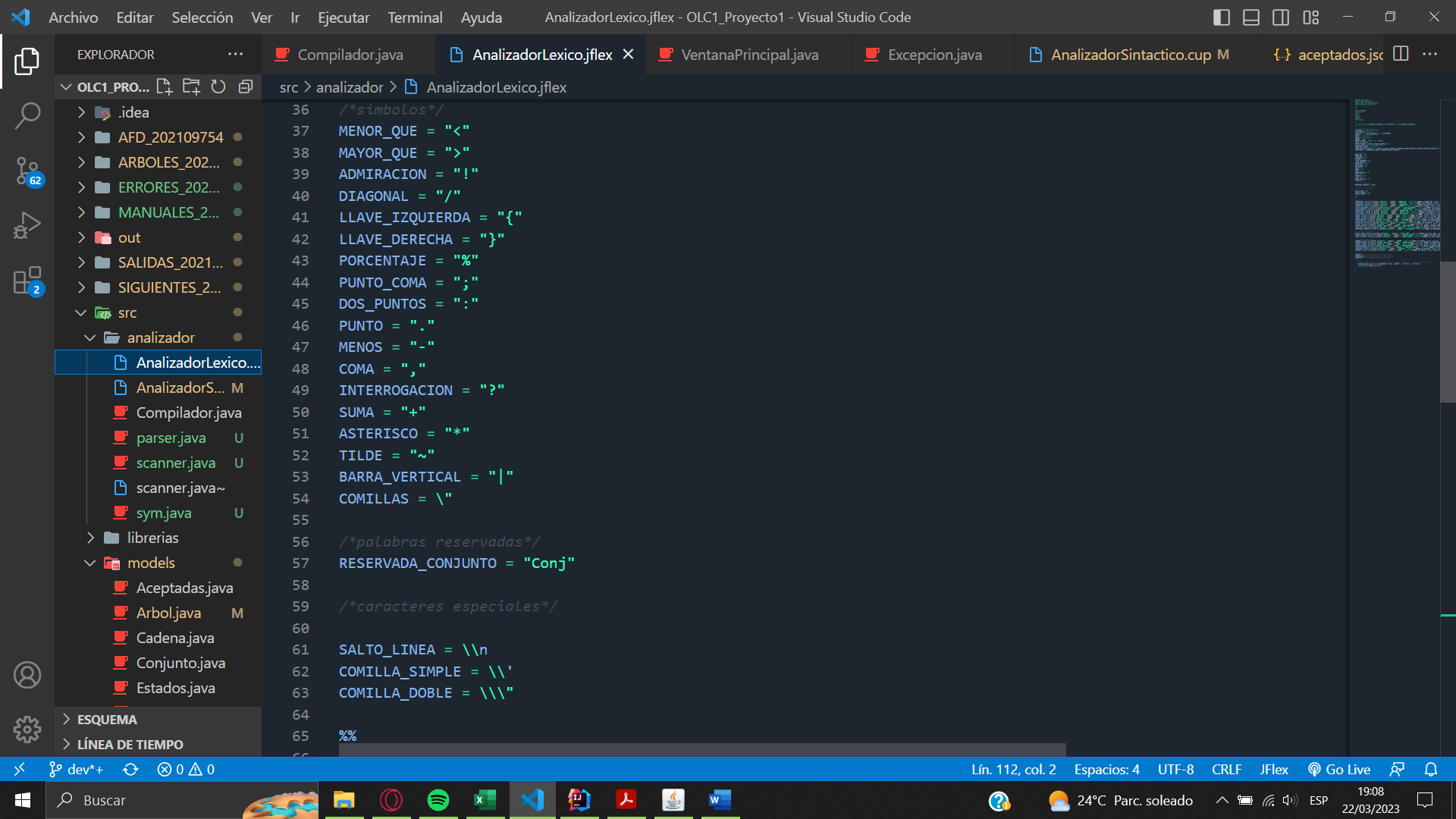


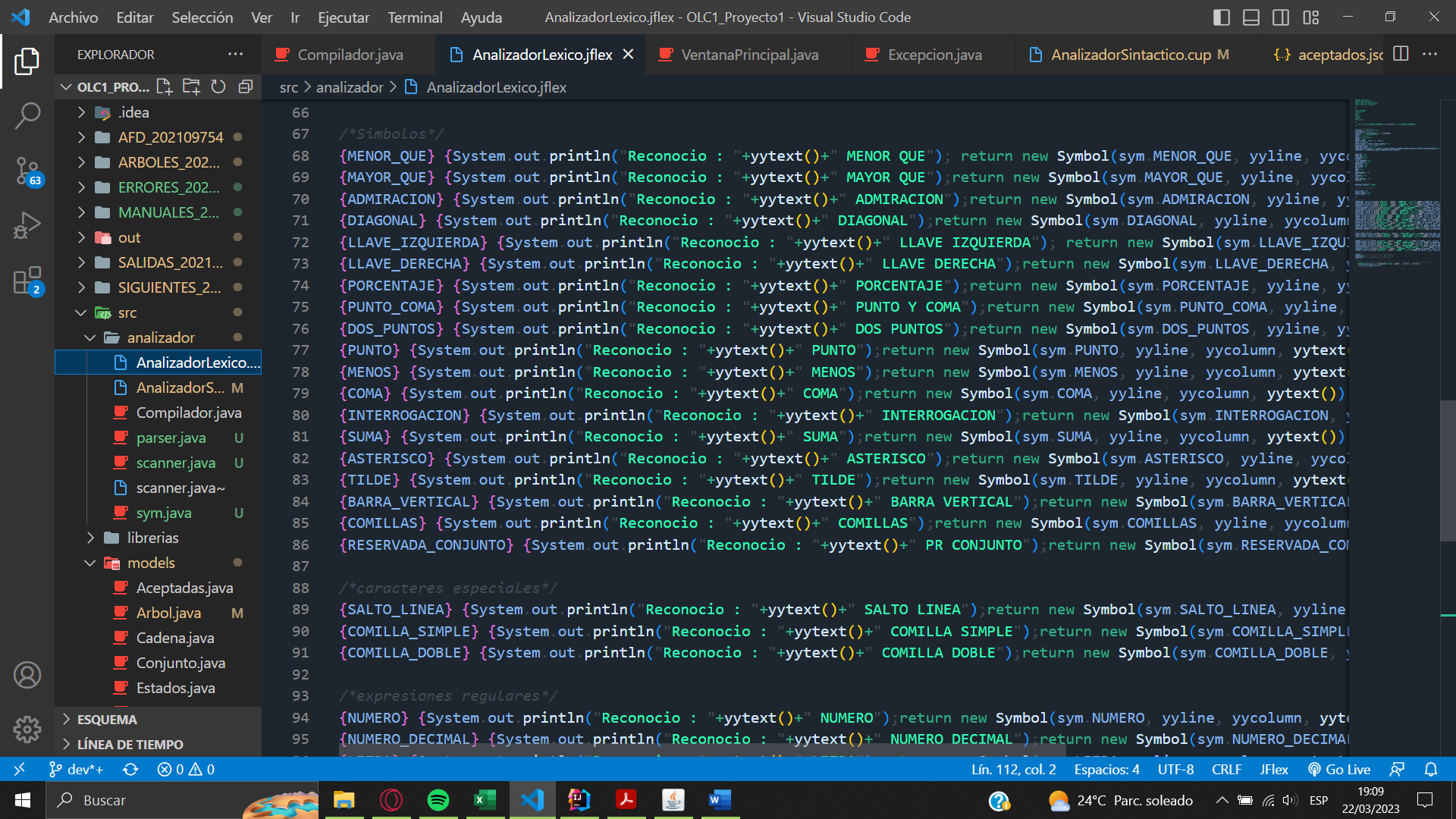
Luego en el área que permite código JAVA se instanció un ArrayList denominado “erroresLexicos” el cual será de utilidad para el almacenamiento de los errores léxicos encontrados en el archivo.



Seguidamente, se encuentran cada una de las expresiones regulares necesarias en el lenguaje, así como el resto de tokens:



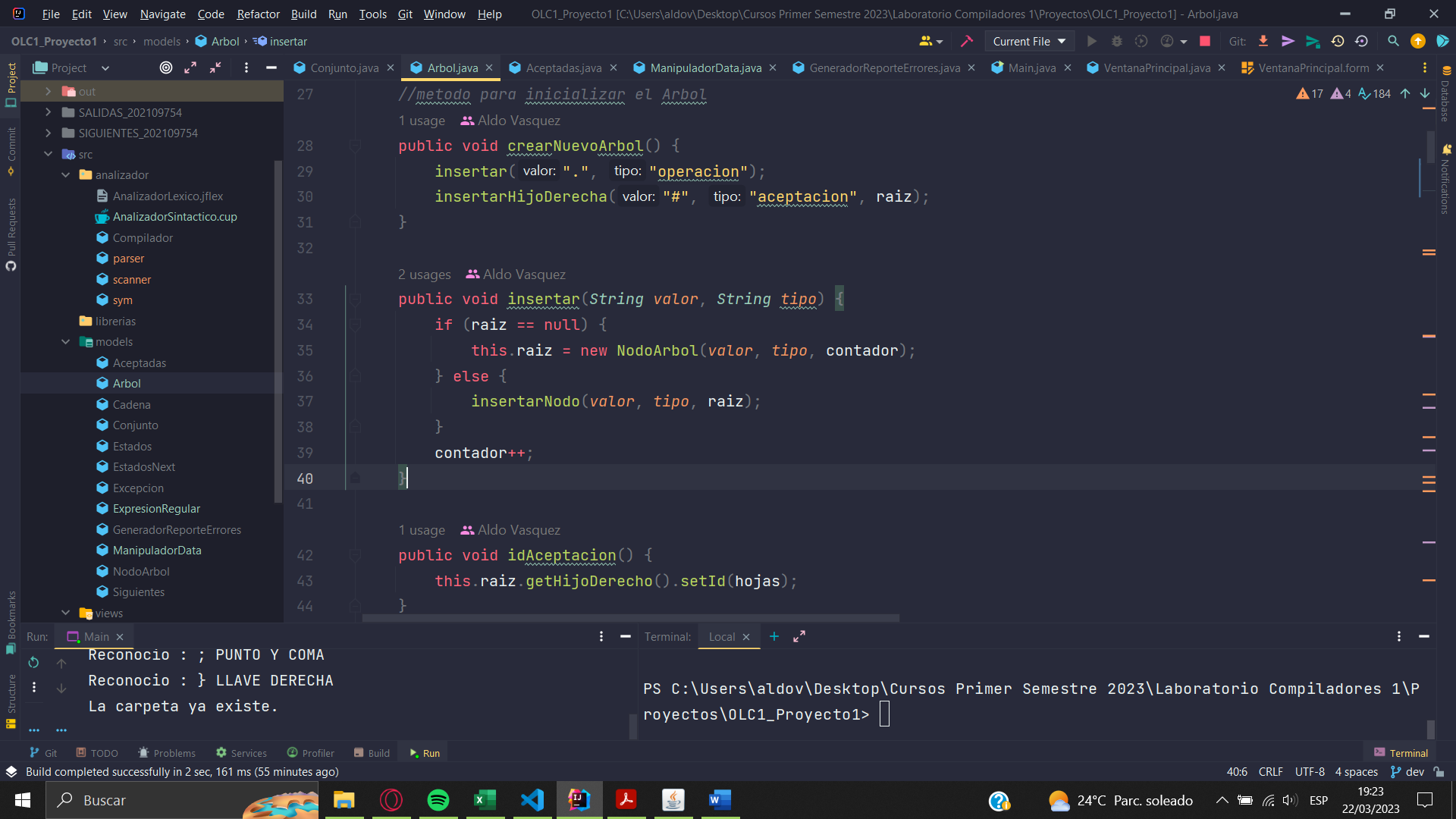


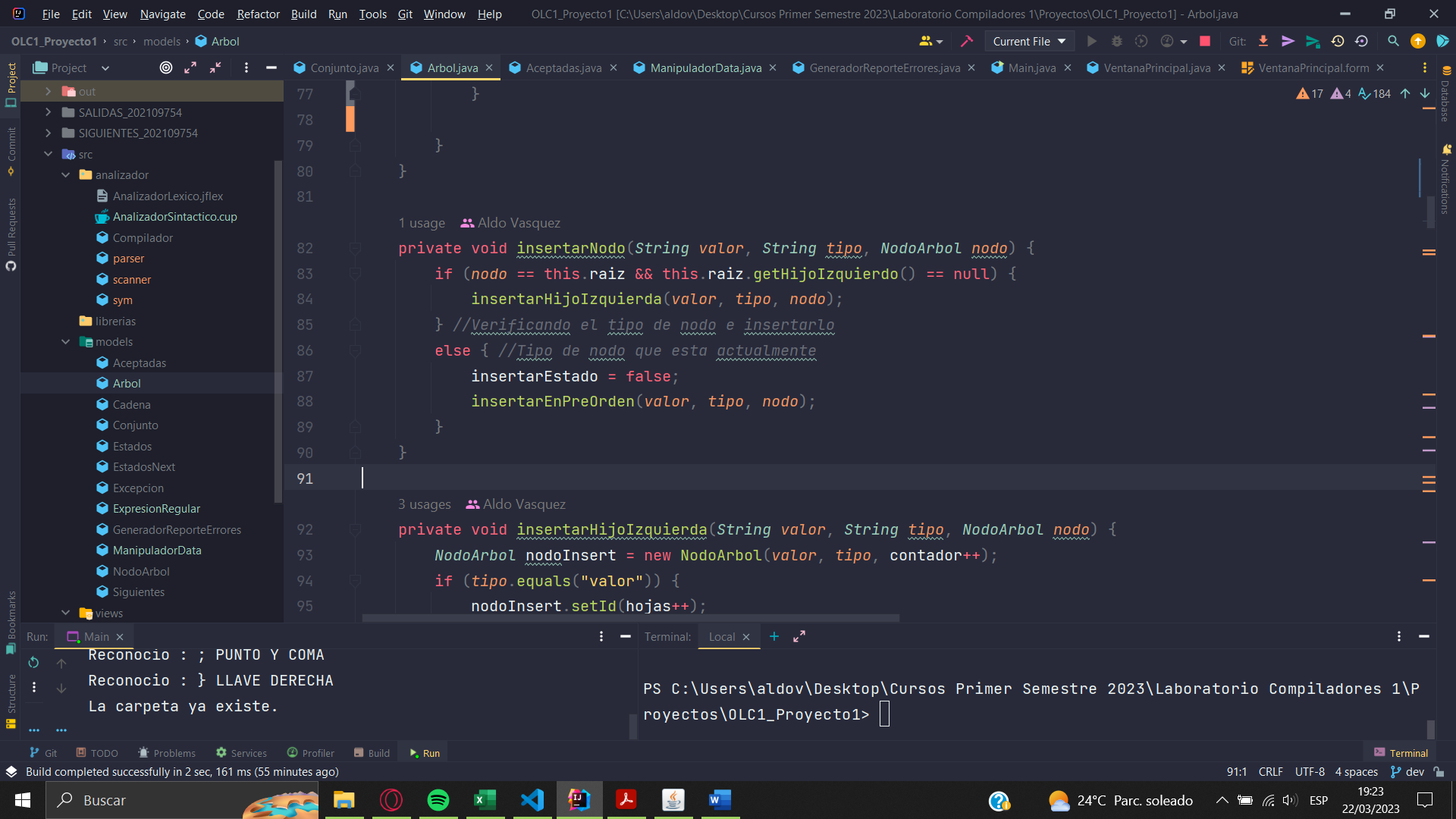
Luego de esto se tiene el área que devuelve los tokens y los retorna al analizador sintáctico:  
  


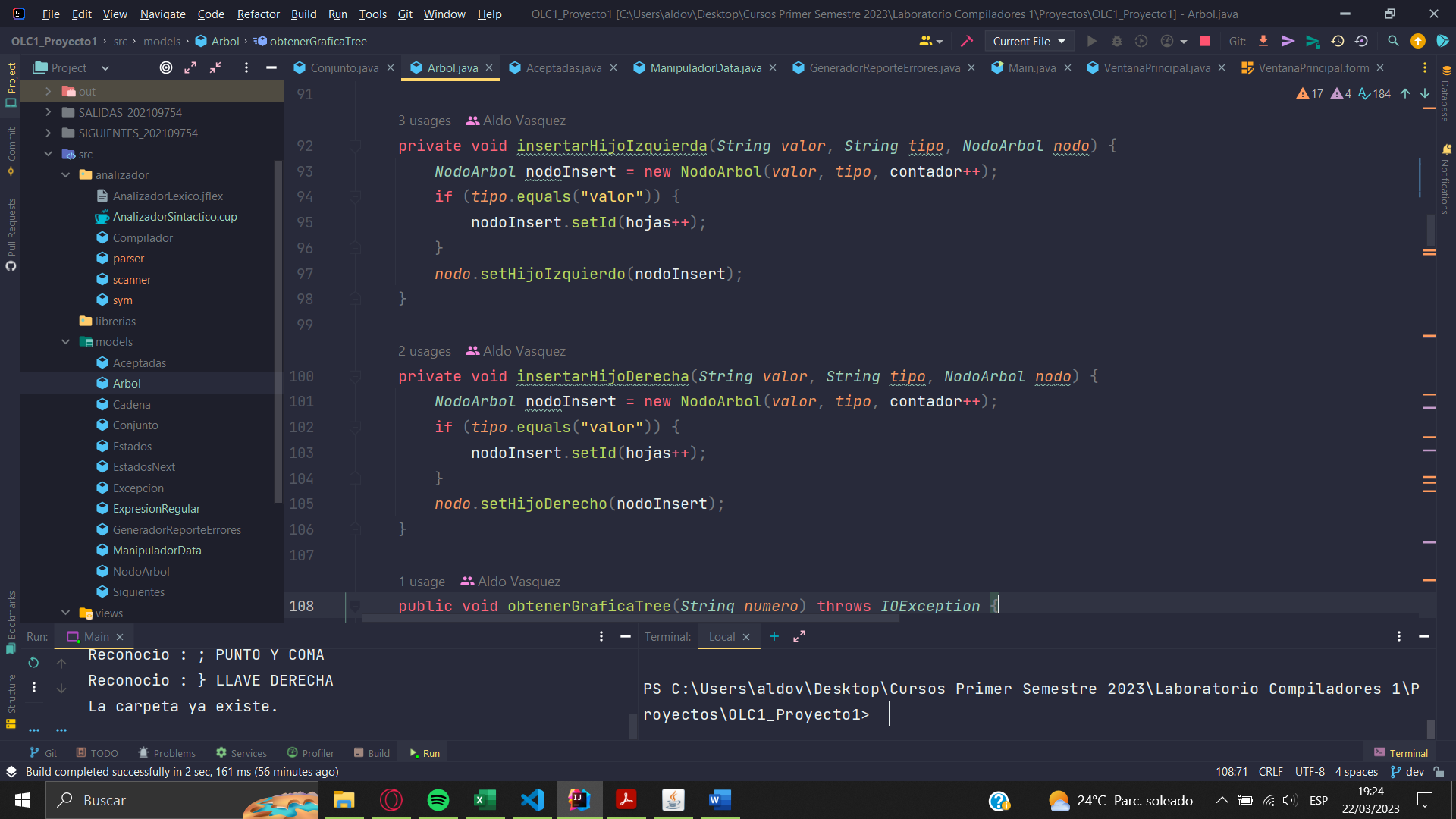
La parte del analizador sintáctico se explicará de mejor manera en el archivo de gramática.

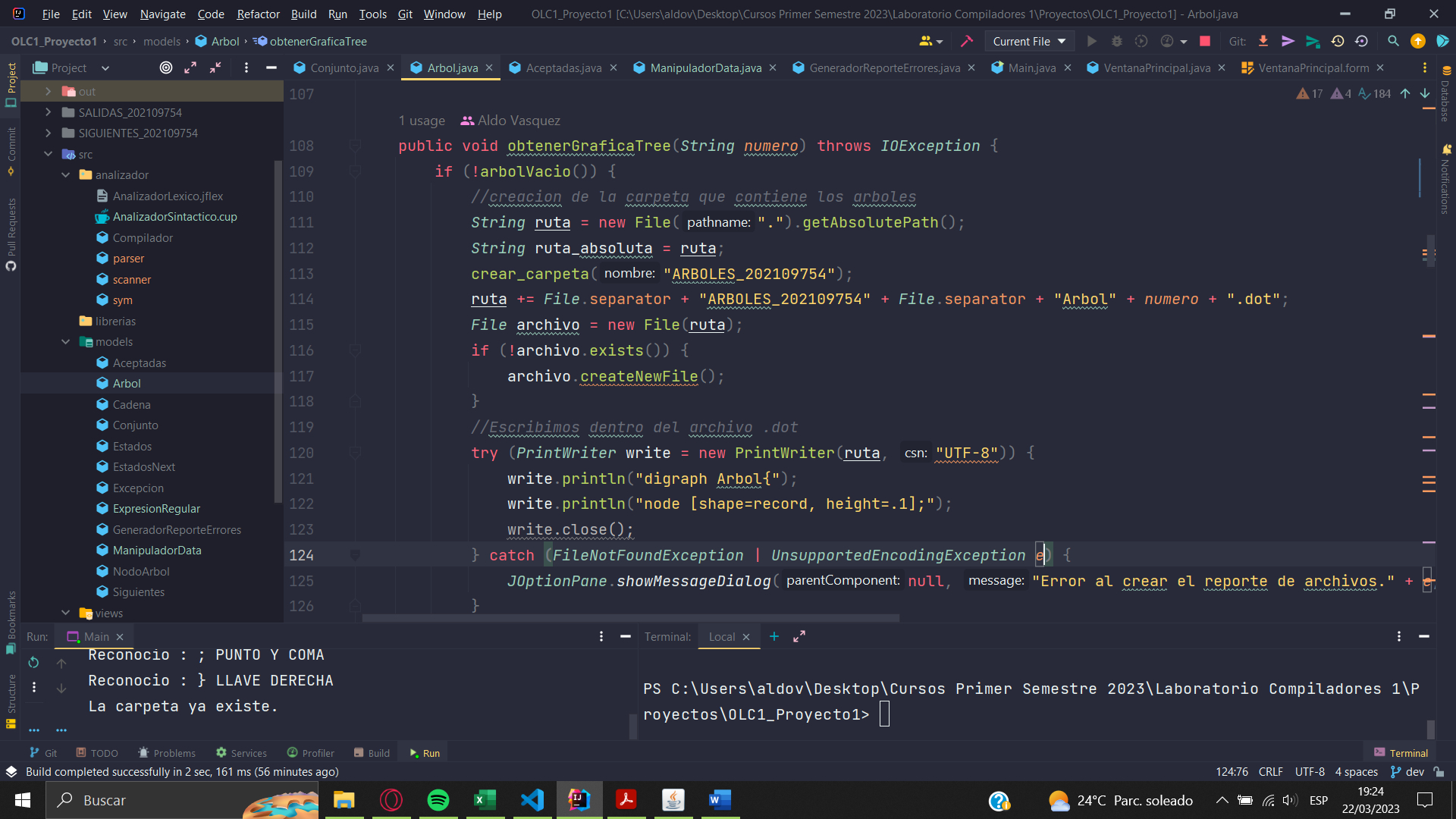
Las clases principales para el desarrollo del sistema se denominan:

* Arbol: Esta clase es de mucha importancia ya que instancia el resto de clases que tienen que ver con el árbol. Por lo cual, obtiene los datos del árbol tal como, los siguientes, los estados, si los nodos son anulables o no, etc. Así como la generación del grafo y diversas funciones más.



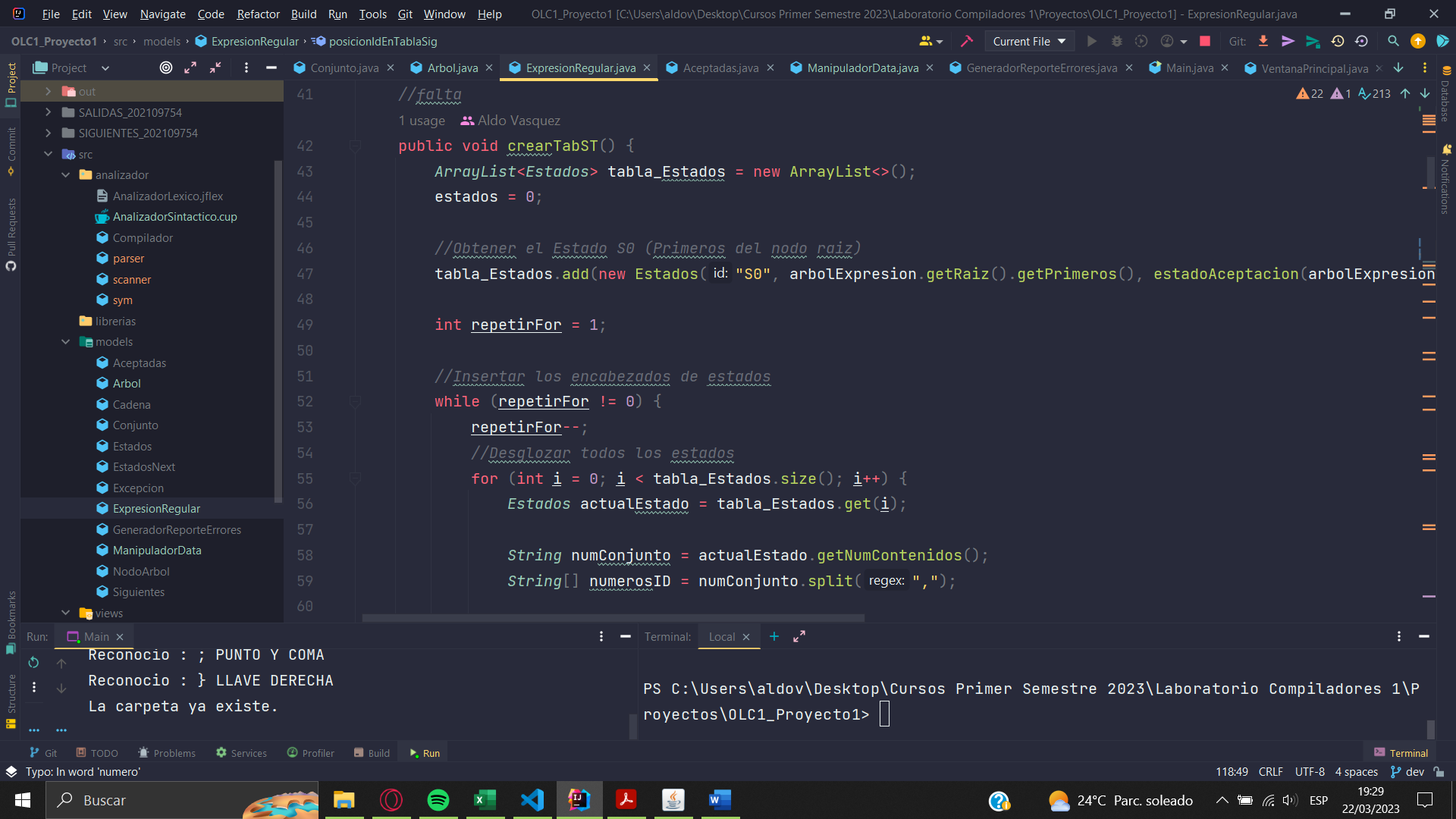


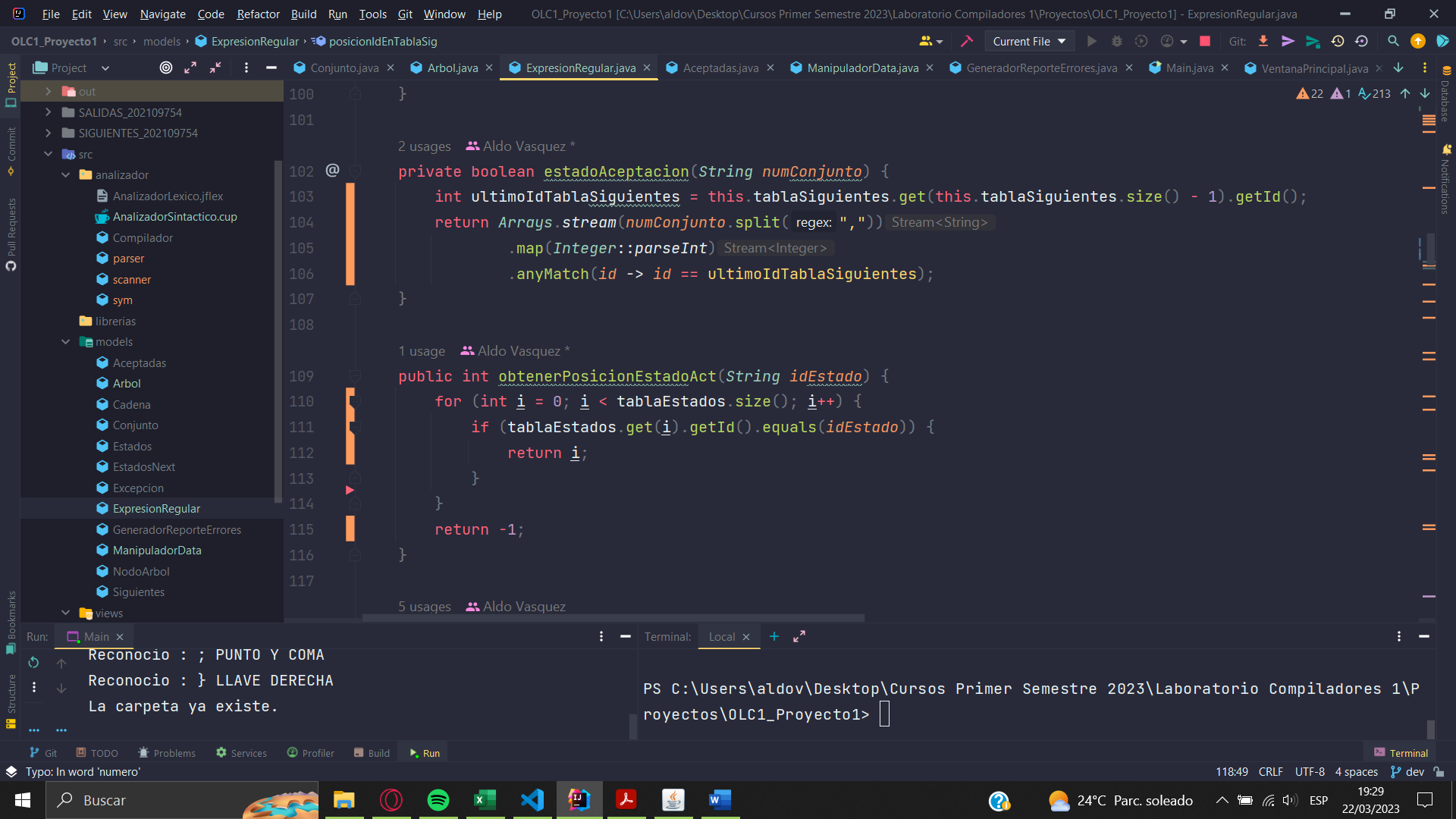


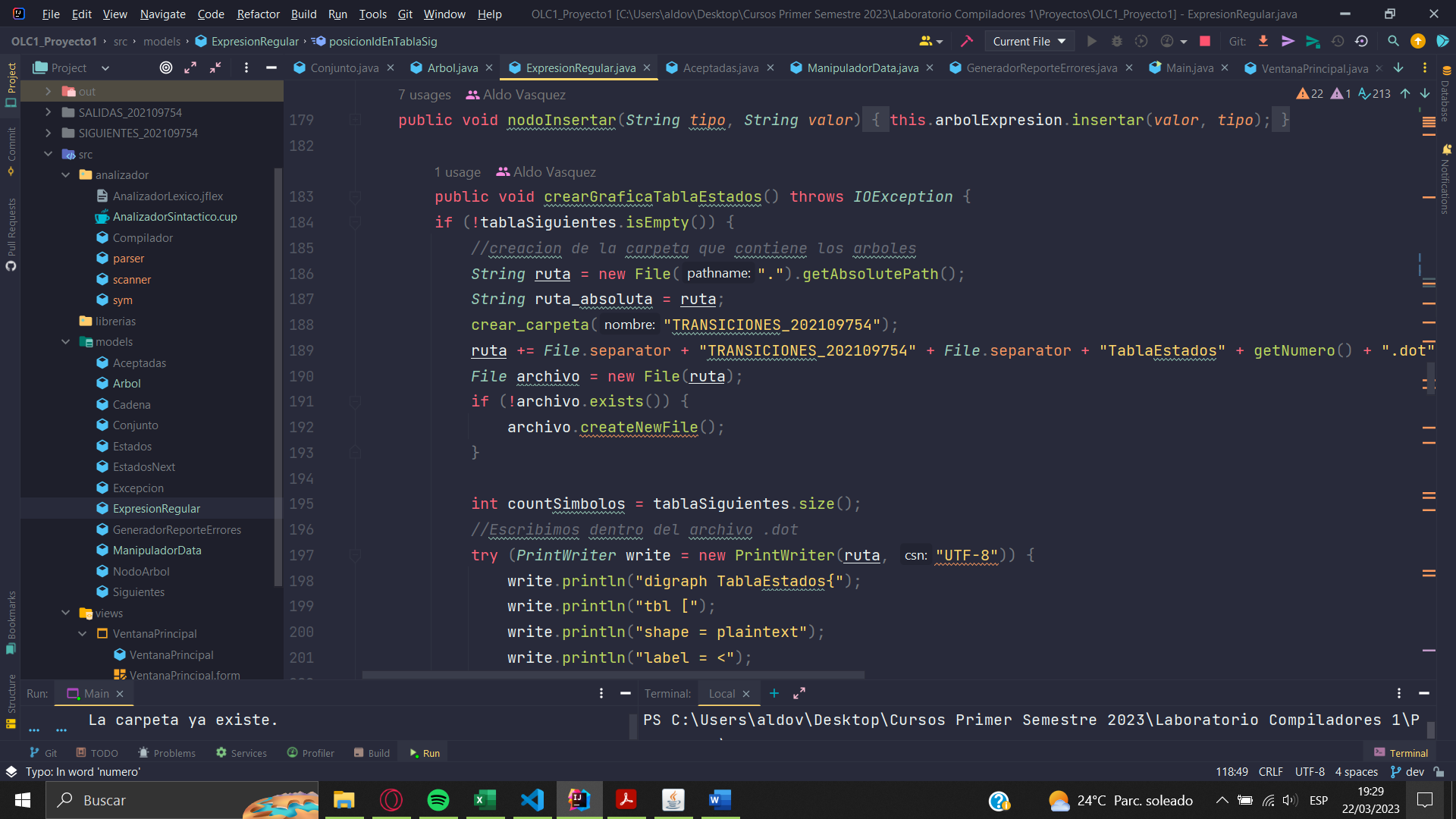


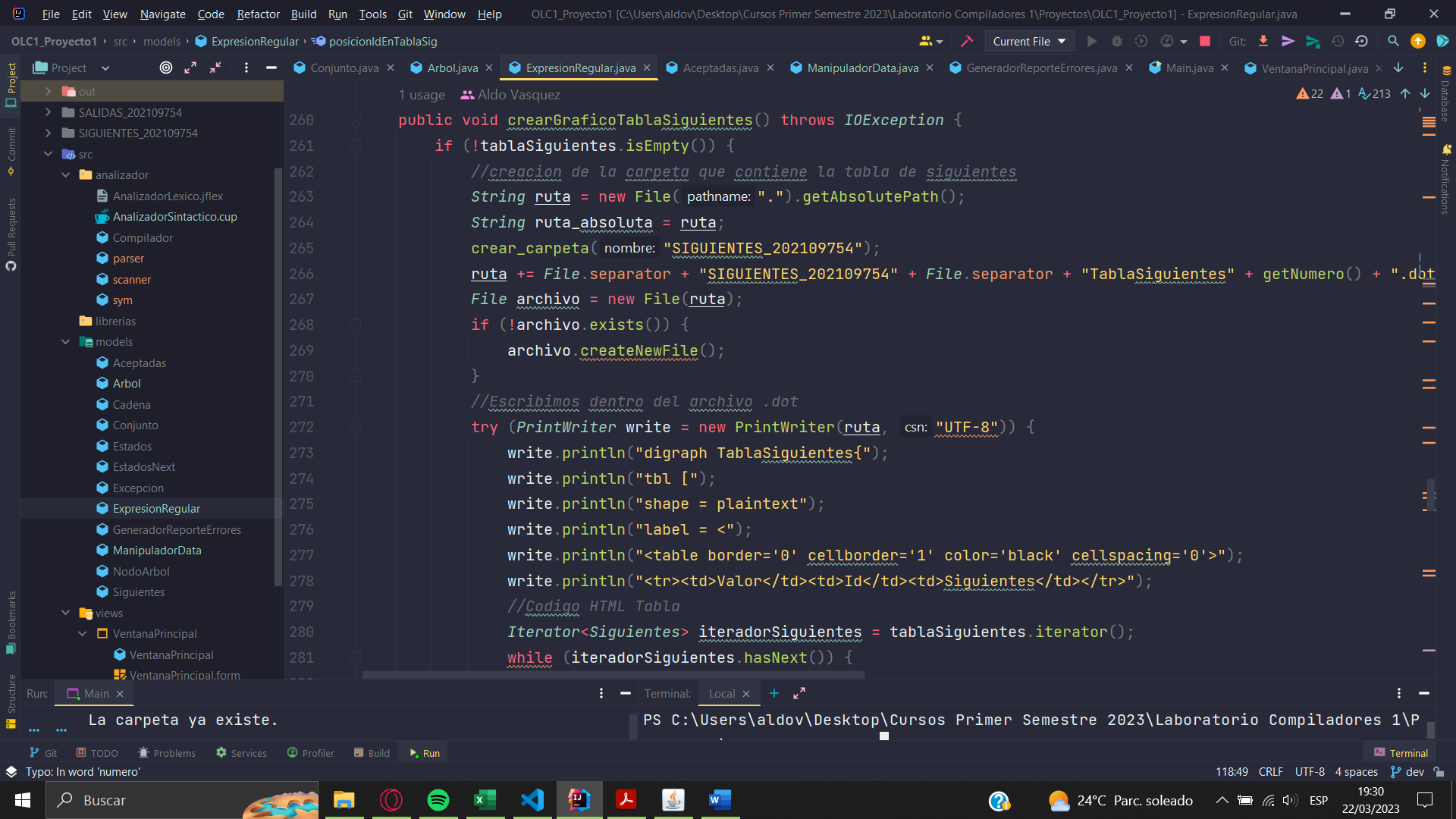
* NodoArbol: Declara las propiedades de un nodo de un autómata finito determinista.
* ExpresionRegular: Esta contiene el funcionamiento principal ya que implementa la clase del árbol, así como las que la clase Arbol instancia. Esto dado que al analizar la expresión regular crea el AFD. Asimismo, obtiene cada uno de los conjuntos declarados, identifica la notación y obtiene los elementos pertenecientes a dicho conjunto. Al obtener los elementos, verifica las expresiones regulares e implementa las funciones de manipulación de información que se encuentran en la clase ManipuladorData para validación de las cadenas exitosamente.

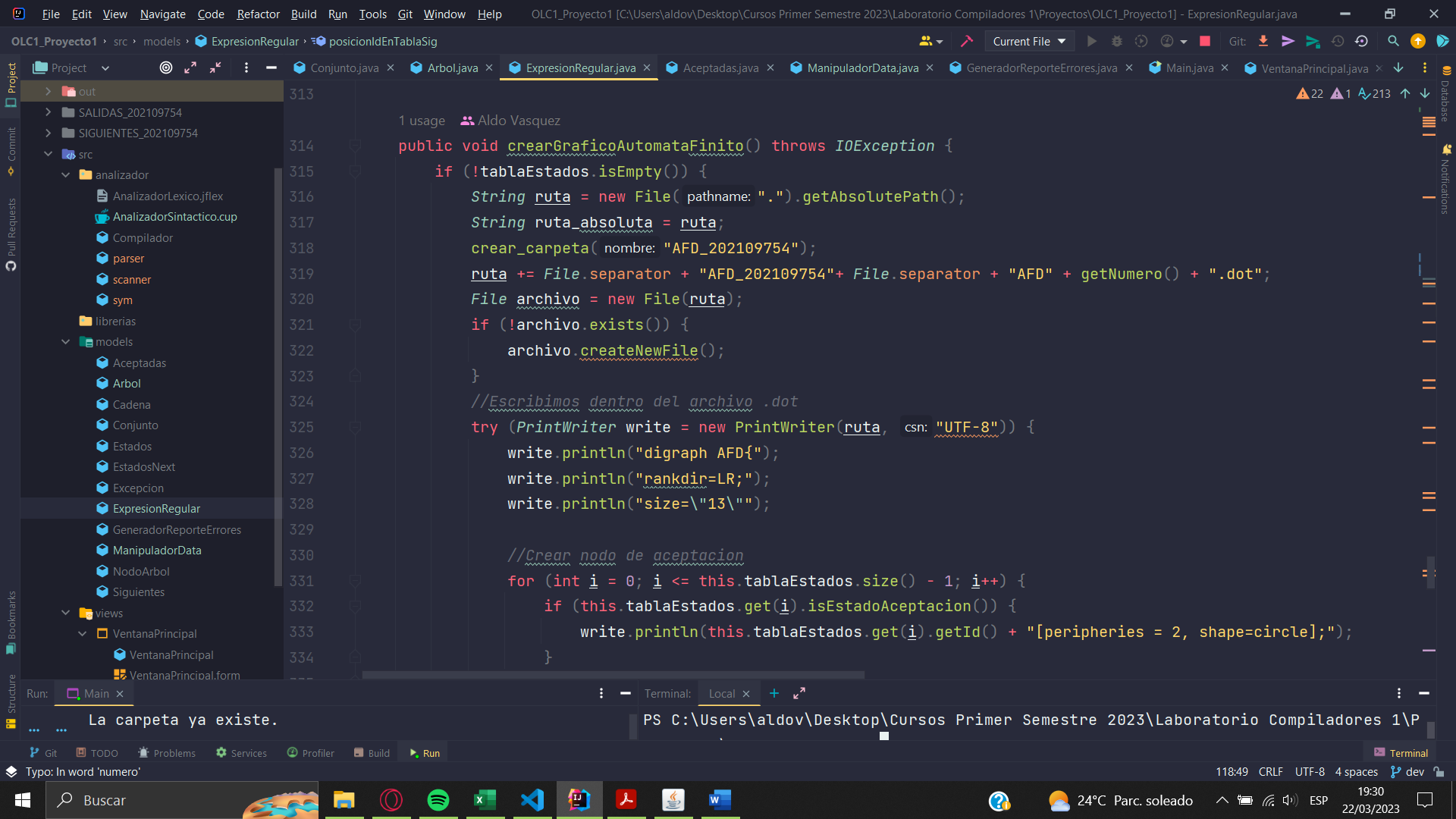
Algunos de los métodos principales de esta clase son:











* Conjunto: Se encarga de declarar las propiedades y funciones principales de un conjunto.
* Estados: Obtiene los estados para el AFD.
* Siguientes: Obtiene los siguientes de un nodo.
* ManipuladorData: Implementa todas las funciones para establecer el funcionamiento básico del programa. Por lo cual, dicha clase se instancia en la ventana principal.
* Cadena: Básicamente se encarga de establecer las propiedades de importancia para la validación de las cadenas.

Se tienen otras clases, sin embargo, su funcionamiento es secundario ya que únicamente cumplen funciones complementarias o de menor importancia en el sistema.

* Excepcion: Esta clase define las propiedades de un error del lenguaje, ya sea léxico o sintáctico.
* GeneradorReporteErrores: Obtiene la lista de errores léxicos y sintácticos, luego genera el reporte web.