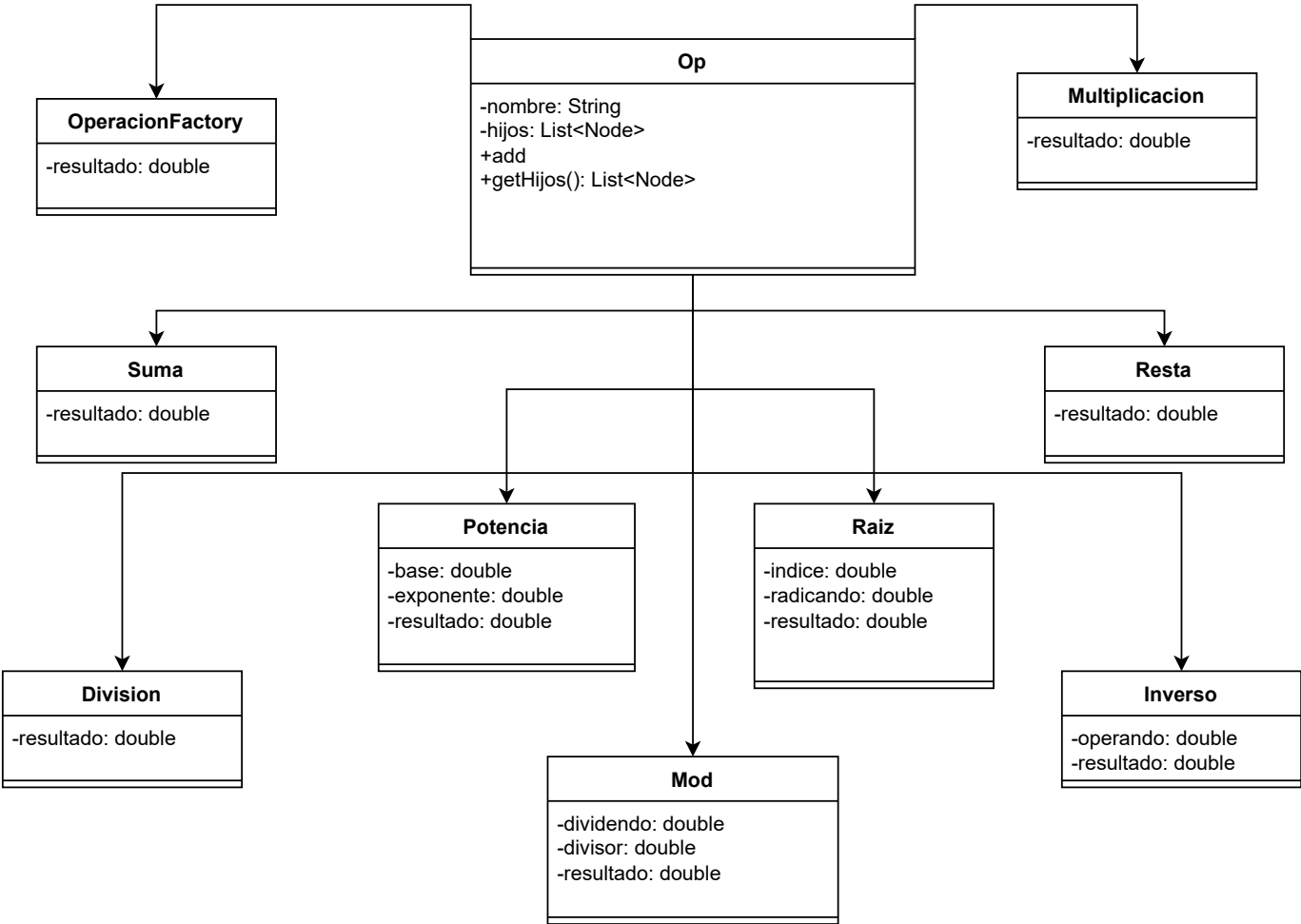


Lexer
- input: String - idx: int - line: int - col: int
+ Lexer(String entrada) + analizar(): List<Token> - siguienteCaracter(): char - retroceder() - agregarToken(String tipo, String valor) + obtenerTokens(): List<Token> + obtenerErrores(): List<String>

Num
-value: double +evaluate(): double +render(): String
+ Num(double valor) + getValor(): double + setValor(double nuevoValor) + getTipoDato(): String + toString(): String // Devuelve el número como cadena.

Token
+ type: TokenType + lexeme: String + line: int + columna: int
+ Token(String tipo, String valor, int fila, int columna) + getTipo(): String + getValor(): String + getFila(): int + getColumna(): int + toString(): String

Num
- tokens: list<Token> - pos: int + parsePrograma(): list<Node> - parseElemento(): node
+ Parser(List<Token> tokens) + parsear(): Node - expresion(): Node - termino(): Node - factor(): Node - numero(): Node + obtenerErrores(): List<String>



ArbolGrafico
-rutaDot: String -rutalmagen: String -contadorNodos: int
+ generarDot(nodoRaiz: Node, rutaSalida: Path)

Analizar
-parserAdapter: ParserAdapter -parser: Parser -tokens: List<Token> -arboles: List<Node>
+ parsearArboles(tokens: List<lexer.Token>): List<ast.Node>

OperacionResultado
- tipoOperacion: String - operandos: List<Double> - resultado: Double - mensaje: String
+ getResumen(): String

Main
-rutaEntrada: String -rutaSalida: String -lexer: Lexe -analizador: Analizar -htmlReport: HtmlReport -arbolGrafico: ArbolGrafico
+ main(args: String[])

InfixPretty
-expresion: String
+ convertir(node: Node): String + limpiarExpresion(): String

HtmlReport
-rutaSalida: String -nombreArchivo: String -errores: List<String> -resultados: List<OperacionResultado> -fechaGeneracion: LocalDateTime
+ HtmlReport(String rutaDestino) + generarReport(List<OperacionResultado> resultados) + agregarFila(String descripcion, String valor) + guardarArchivo()

AppFrame
-txtEntrada: JTextArea -btnAnalizar: JButton -btnReporte: JButton -lblEstado: JLabel -analizador: Analizar -htmlReport: HtmlReport -arbolGrafico: ArbolGrafico
+ inicializarComponentes() + ejecutarAnalisis() + mostrarResultados()

GuiMain
-ventanaPrincipal: AppFrame
+ main(args: String[])