



THUNDER TEAM

# ALGORITMIA

El algoritmo  
musical



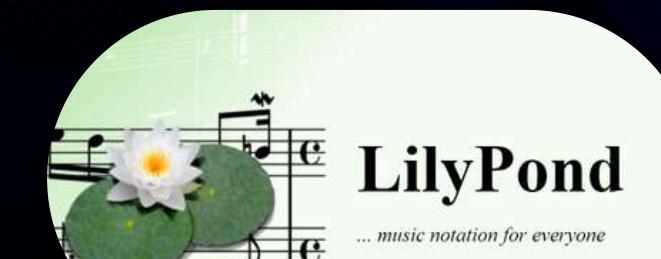


# Conozca a Algoritmia

1 Algoritmia es un lenguaje orientado a la composición musical algorítmica

2 LilyPond genera la partitura (PDF).  
Se crea un archivo MIDI desde el código musical.

3 FluidSynth o TiMidity++ convierten ese MIDI en audio WAV.





THUNDER  
TEAM

HOME

SERVICE

ABOUT US

CONTACT US

# Thunder Team



FRANK  
JAUREGUI



NARDY  
CONDORI



JESÚS  
MORALES



BRIGITTE  
BERNAL



# Objetivos :

1 Implementar un doble intérprete musical.

2 Generar partitura y reproducir melodías con lilyponds, timidity++ , Python y ANTLR.



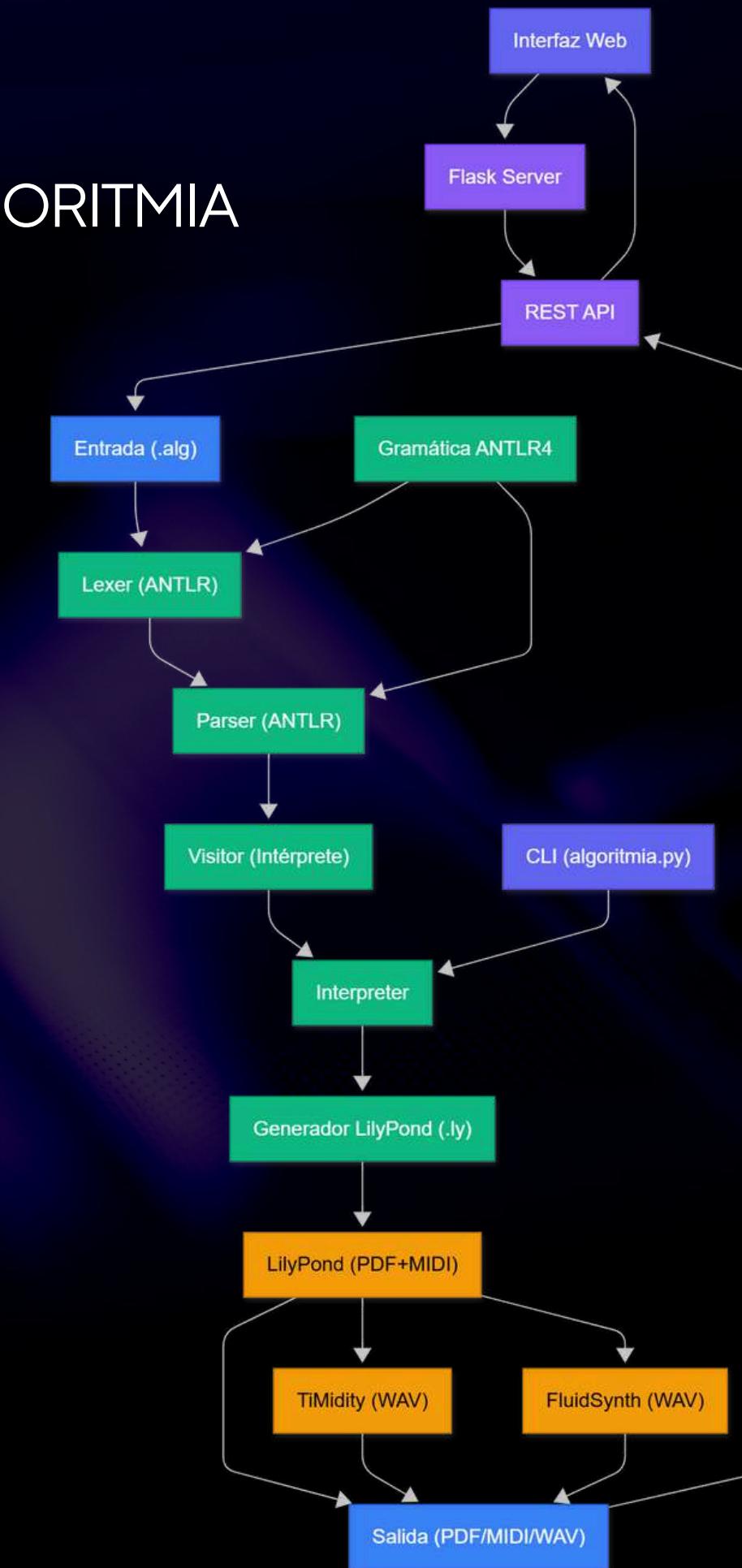


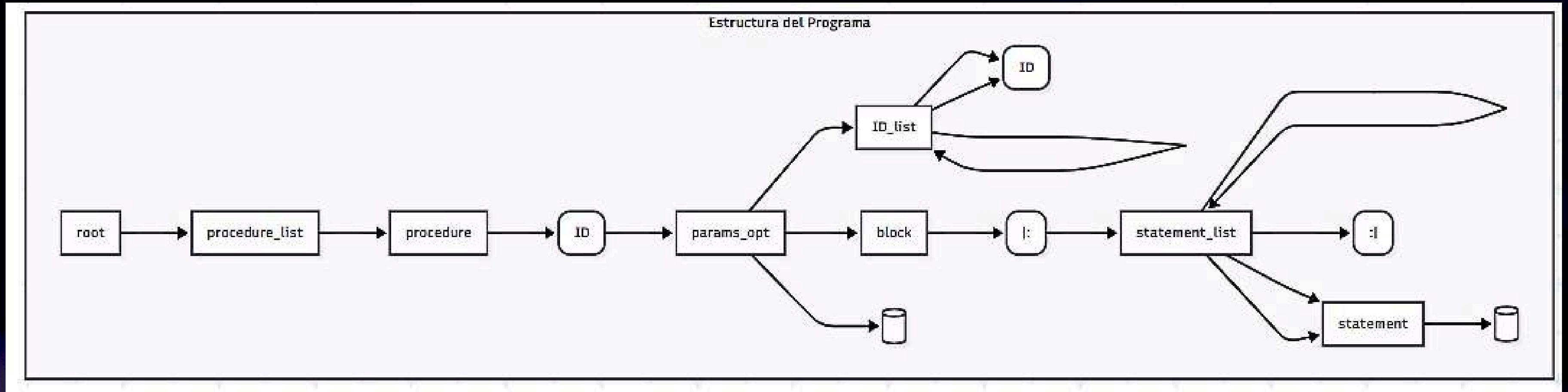
THUNDER  
TEAM

# Arquitectura

DIAGRAMA DE ARQUITECTURA DE ALGORITMIA

- FRONTEND (INTERFAZ WEB / CLI)
- BACKEND (FLASK / API)
- CORE (LEXER, PARSER, VISITOR, INTERPRETER, LILYGEN, GRAMÁTICA)
- HERRAMIENTAS EXTERNAS (LILYPOND, TIMIDITY, FLUIDSYNTH)
- STORAGE (ENTRADA/SALIDA DE ARCHIVOS)





## DIAGRAMA DE SINTAXIS FERROVIARIA DE ALGORITMIA.G4

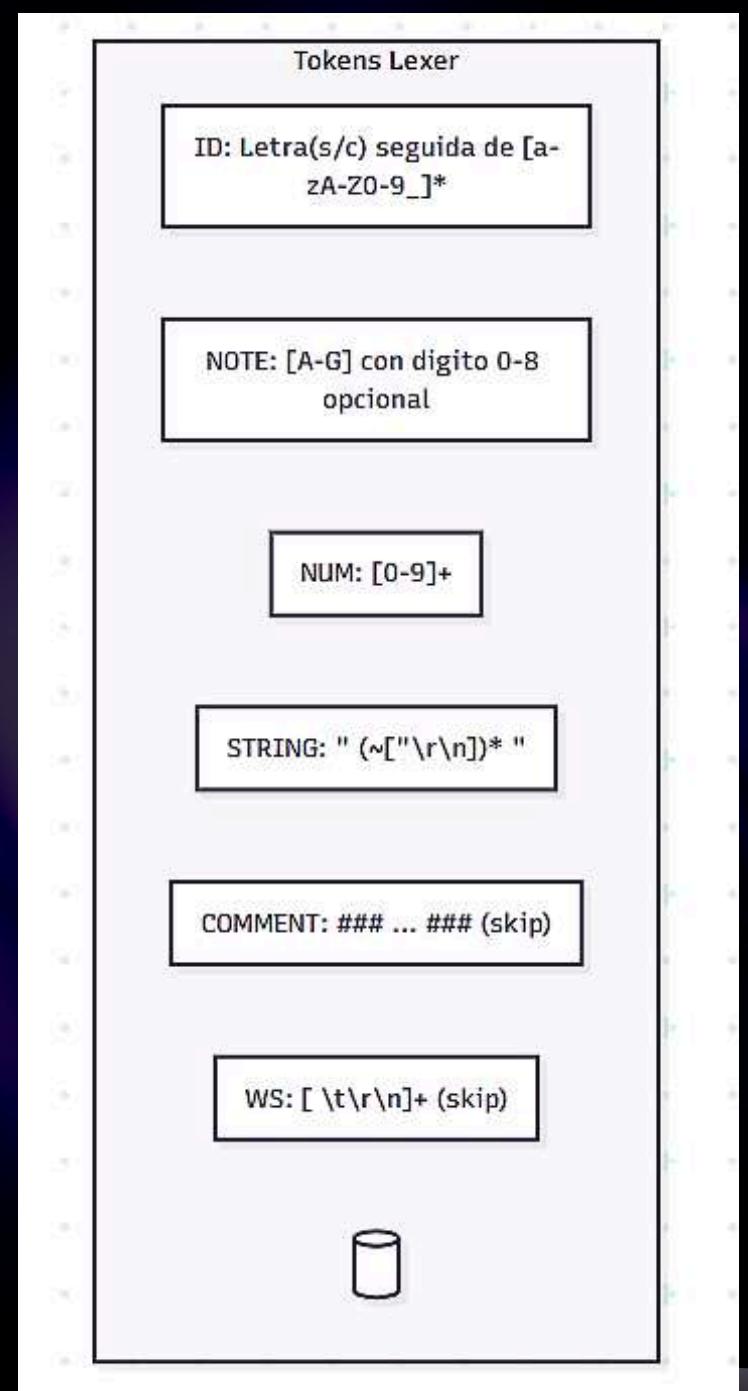
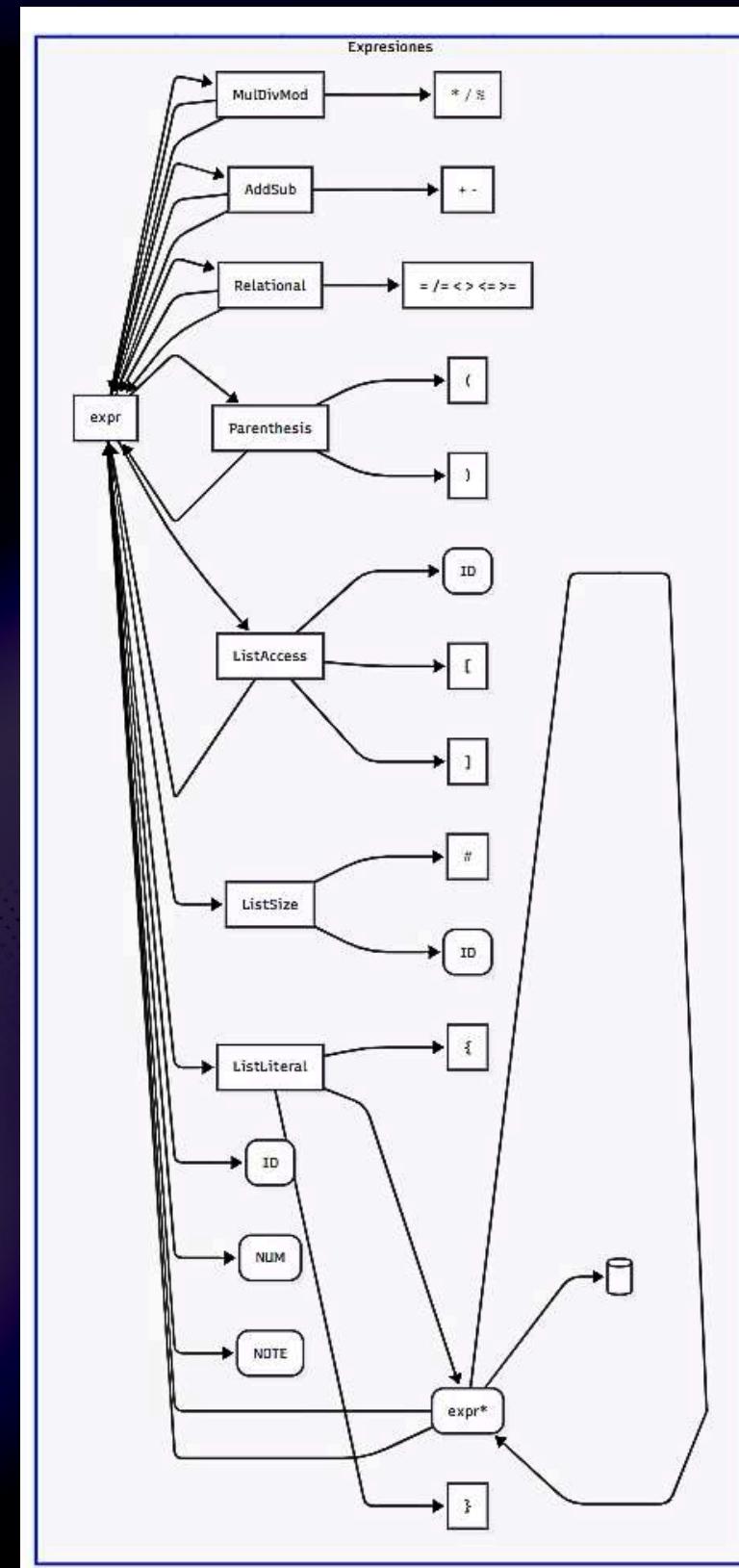
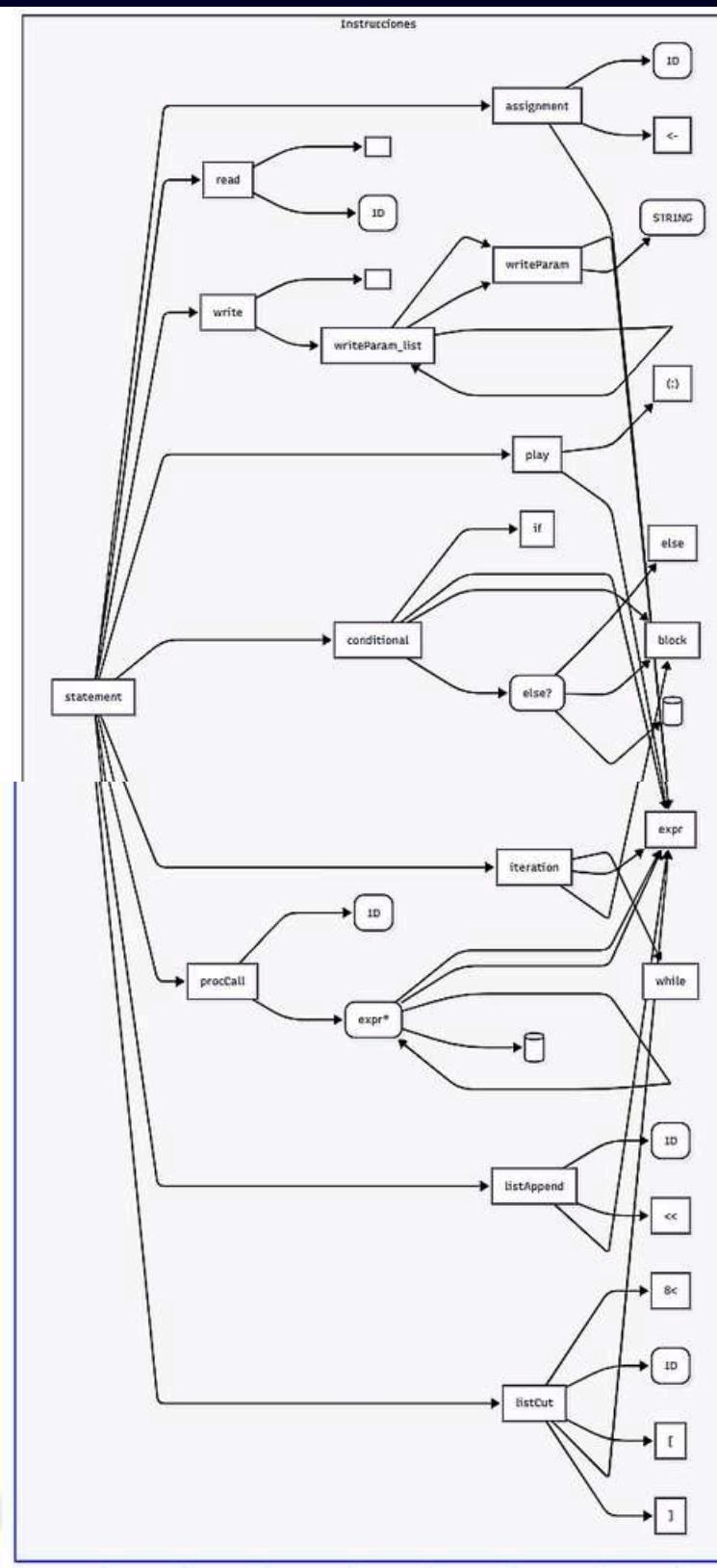
### DEFINE:

- QUÉ PALABRAS SON VÁLIDAS (TOKENS)
- CÓMO SE COMBINAN (SINTAXIS)
- QUÉ ESTRUCTURAS SE PERMITEN (PROCEDIMIENTOS, BLOQUES, EXPRESIONES)
- LA PRECEDENCIA DE OPERADORES

# Algoritmia.g4



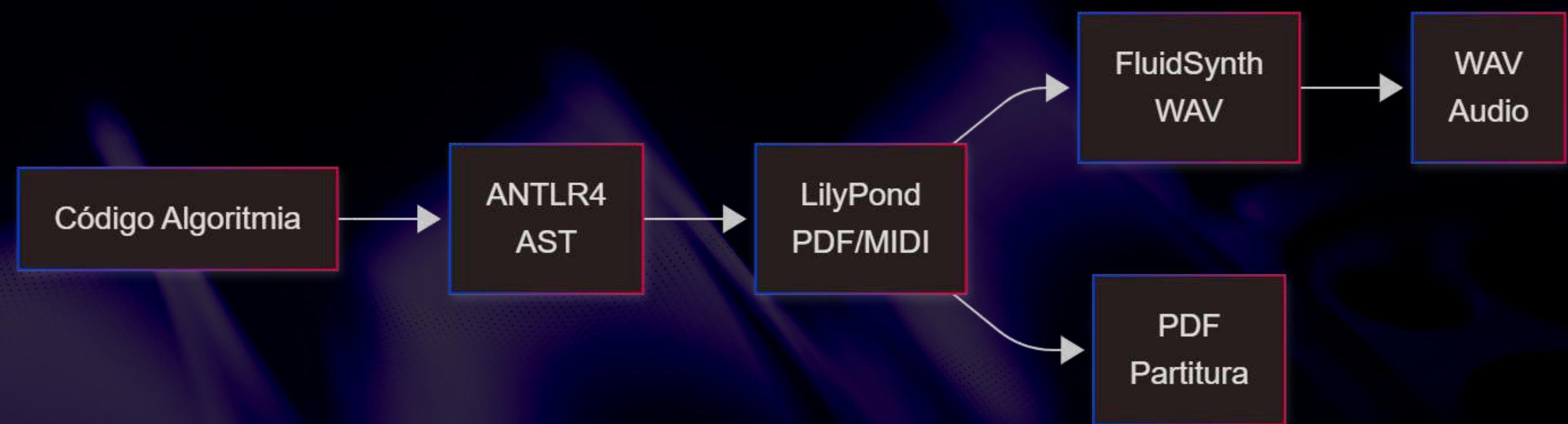
# DIAGRAMA DE SINTAXIS FERROVIARIA DE ALGORITMIA.G4:

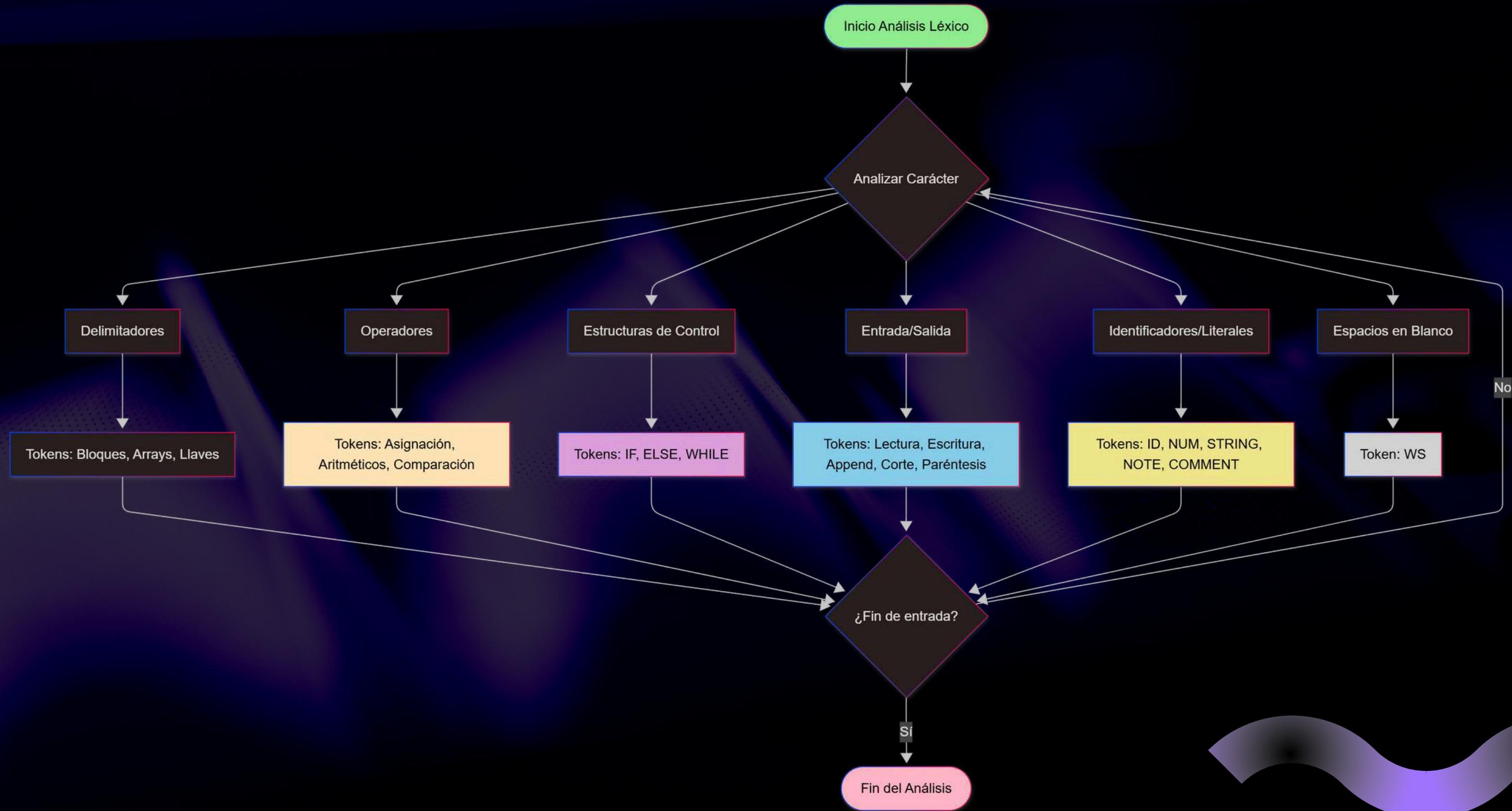


- INSTRUCCIONES
  - EXPRESIONES
  - TOKEN LEXER



# AlgoritmiaParser.py







# Análisis sintáctico

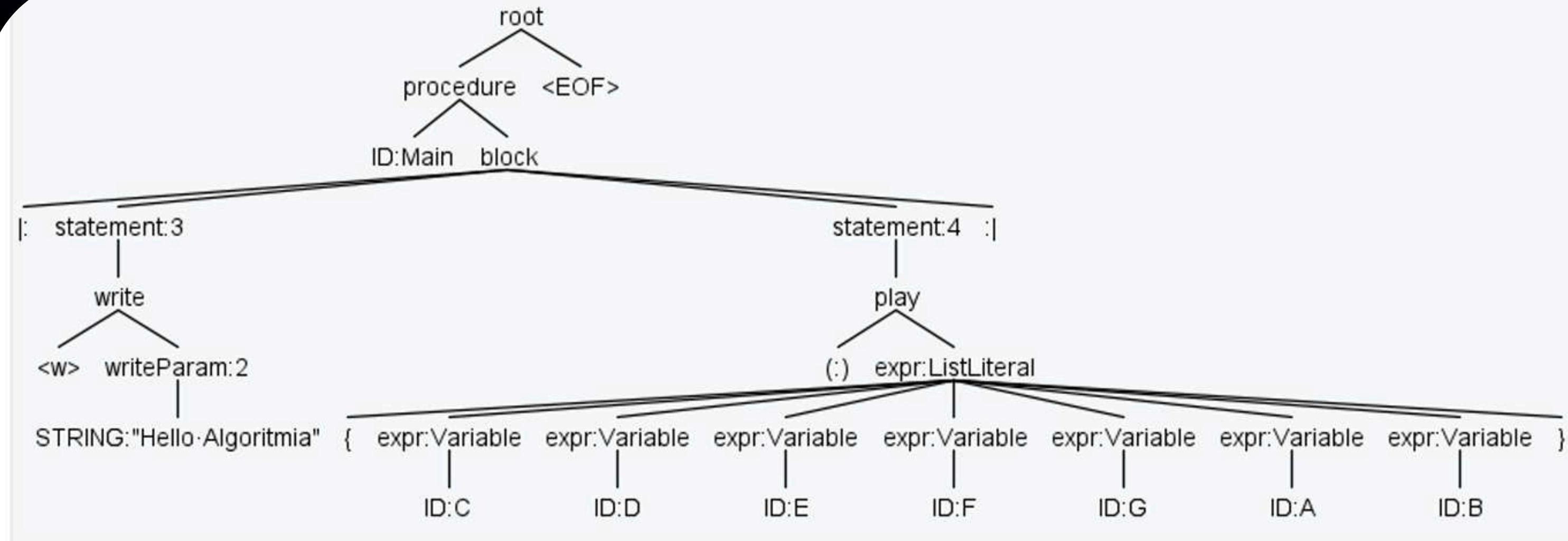
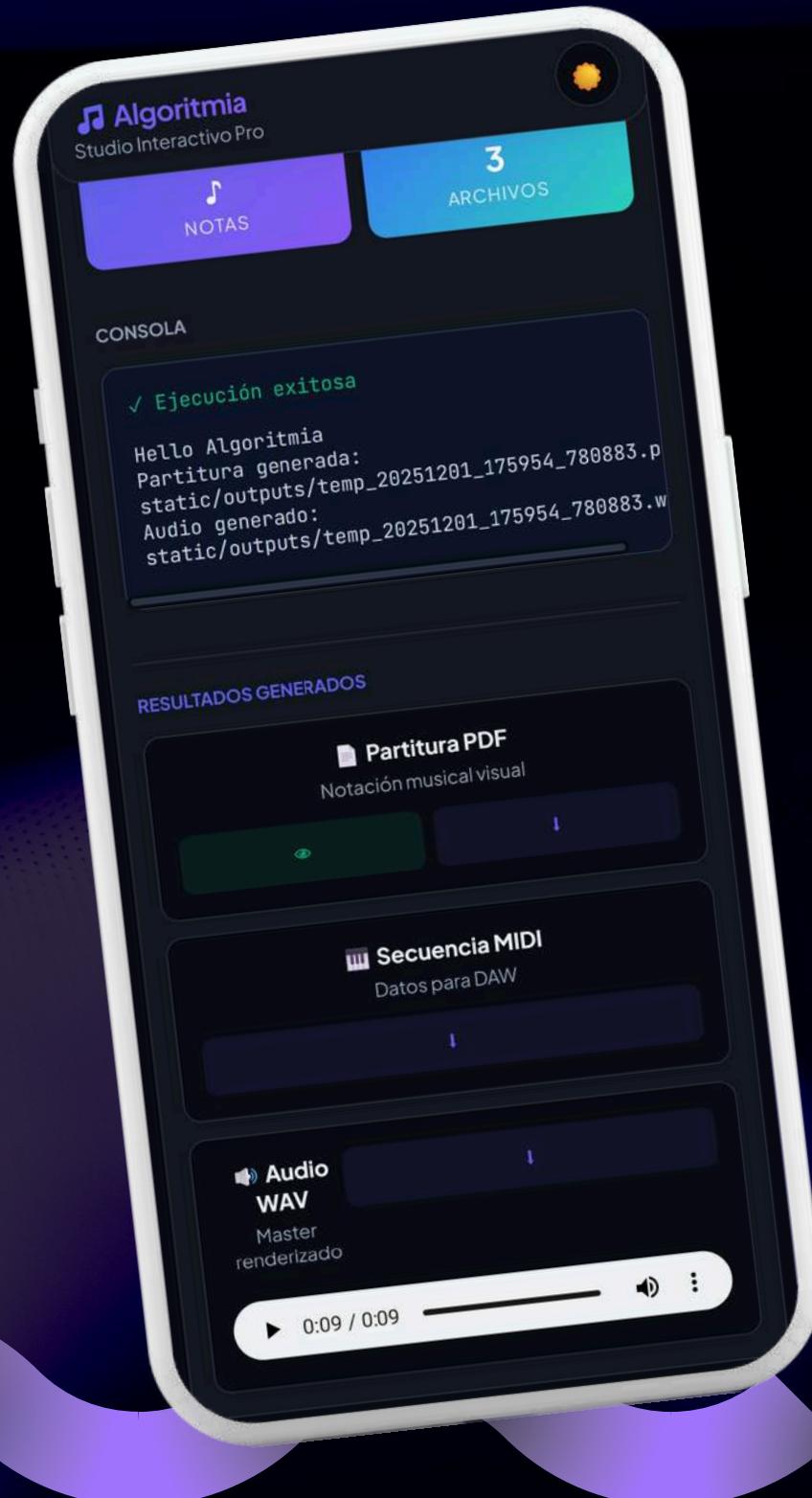
[HOME](#)[SERVICE](#)[ABOUT US](#)[CONTACT US](#)

Figura 3.3: Ejemplo de una Melodía Simple (Hello Algoritmia) en un árbol de análisis sintáctico (AST)



THUNDER  
TEAM

# Resultados



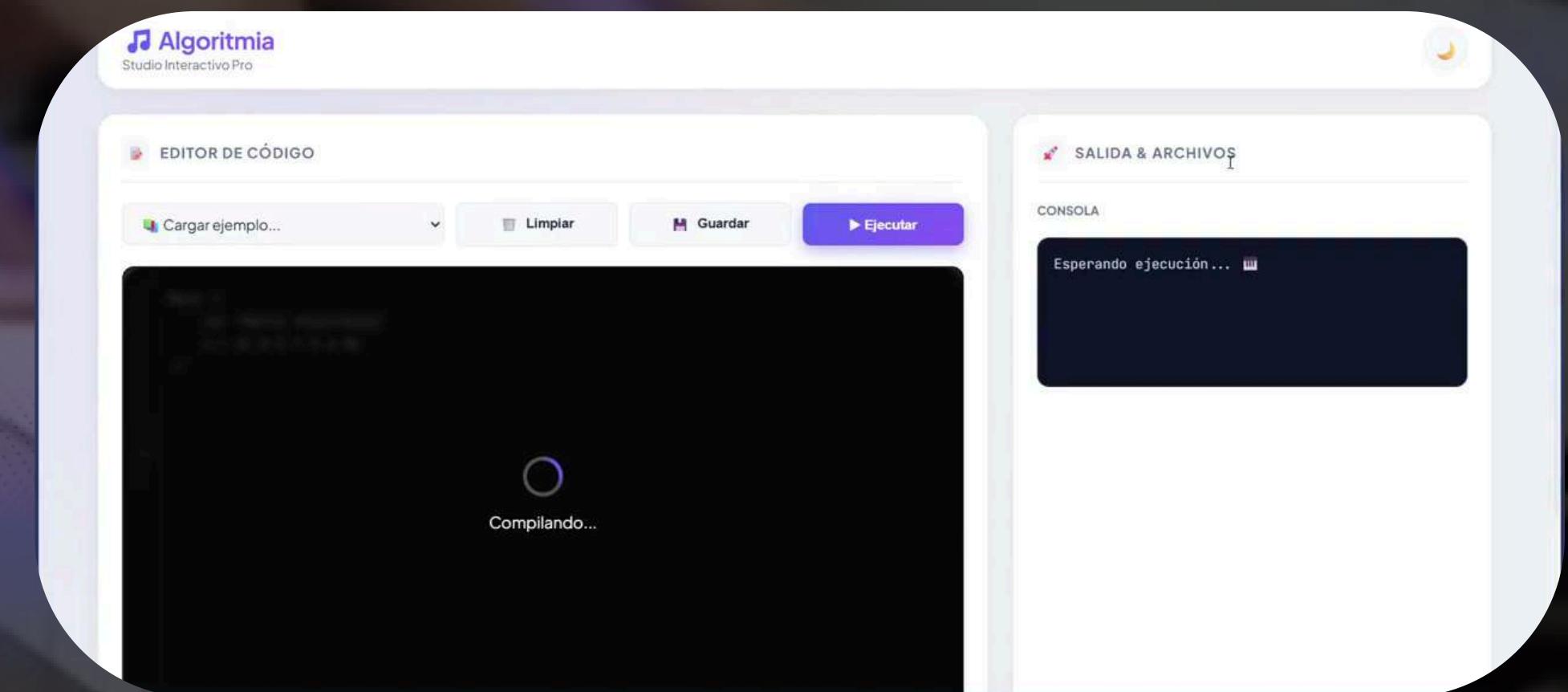
- **MODO OSCURO**
- **COMPATIBLE EN MÓVILES**
- **DESCARGAS EN FORMATO PDF, MIDI Y WAV**





# Conclusiones

- Desarrollo exitoso de un intérprete web funcional con generación de partituras PDF, archivos MIDI y audio WAV de calidad
- Dificultades técnicas, especialmente en la generación de archivos WAV y en la ejecución de código complejo
- Mejoras futuras: integración de bancos de sonidos, sistema de depuración robusto y nuevas estructuras de control musical





# BIBLIOGRAFÍA

- Parr, T. (2013). The Definitive ANTLR 4 Reference. Pragmatic Bookshelf. Recuperado de: <https://dl.icdst.org/pdfs/files3/a91ace57a8c4c8cdd9f1663e1051bf93.pdf>
- ANTLR Official Documentation (2024). ANTLR 4 GitHub Repository. Recuperado de: <https://github.com/antlr/antlr4>
- Parr, T. (2020). Language Implementation Patterns. Pragmatic Bookshelf. Recuperado de: <https://nlogn.art/wp-content/uploads/2023/04/Language-Implementation-Patterns.pdf>
- LilyPond Development Team (2024). LilyPond Notation Reference. Recuperado de: <https://lilypond.org/doc/v2.24/Documentation/learning>
- LilyPond Development Team (2024). Automated Engraving: The LilyPond Approach. Recuperado de: <https://lilypond.org/doc/v2.24/Documentation/essay-big-page.html>
- Timidity++ Development Team (2024). Timidity++ Documentation. Recuperado de: <https://timidity.sourceforge.net>
- SoundFont Technical Specification (2024). SoundFont® 2.04 File Format Specification. Recuperado de: <https://soundfont3.pages.dev/routes/2%20SF%202.04%20Spec/README.html>
- Grune, D., van Reeuwijk, K., Bal, H. E., Jacobs, C. J., & Langendoen, K. (2012). Modern Compiler Design. Springer. Recuperado de: <https://dpviprcollege.ac.in/wp-content/uploads/2023/01/Modern.Compiler.Design.2nd.pdf>





# MUCHAS GRACIAS