

# Manual Técnico

NoteBook es una aplicación para Android que permite a los usuarios crear y editar texto con formato Markdown, además de ejecutar código Python para realizar operaciones aritméticas. Esta herramienta está diseñada para brindar una experiencia enriquecida en la creación de notas estructuradas con contenido matemático y programático.

# Tecnologías Usadas

## Lenguaje de Programación

- Kotlin:** Lenguaje de programación para funcionalidad y diseño de la aplicación Android.
- Java 17:** Lenguaje de programación para la lógica de analizadores y estructura de la aplicación.



# Analizadores de java

- **JFlex:** Generador de analizadores léxicos para generar Tokens, expresiones regulares para Texto y Código.
- **CUP:** Herramienta para generar analizadores sintácticos por medio de gramática y producciones.

The JFlex logo is a white rectangular box with a thin black border. Inside the box, the word "JFlex" is written in a black, sans-serif font, centered horizontally and vertically.

JFlex

The JavaCup logo features the word "JavaCup" in a stylized, italicized, serif font. The "C" is large and prominent. To the right of the "C" is a small, dark gray icon of a coffee cup. The entire logo is set against a light gray rectangular background.

JavaCup

# Ide utilizado para la creación de la aplicación

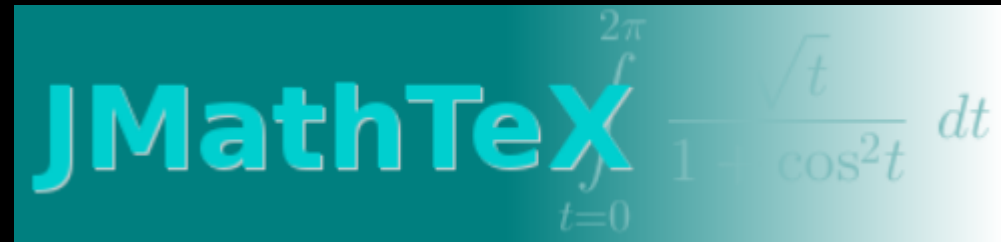
**Android Studio:** Entorno de desarrollo integrado (IDE) utilizado para la creación y compilación de la aplicación.

android  
studio




# Librerías y Dependencias

- Markdown**: Lenguaje de marcado ligero para el formato de texto.
- JLaTeXMath**: Biblioteca para la representación de expresiones matemáticas en formato LaTeX.



# Instalación y Configuración

- Android Studio instalado.
- JDK 17 configurado.
- Dependencias necesarias en build.gradle:



```
1 dependencies {
2     implementation("com.github.vbmacher:java-cup-runtime:11b-20160615") // JAVACUPRUNTIME
3     implementation("ru.noties:jlatexmath-android:0.2.0") // Jlatex
4     implementation("ru.noties:jlatexmath-android-font-cyrillic:0.2.0") // for Cyrillic symbols
5     implementation("ru.noties:jlatexmath-android-font-greek:0.2.0") // for Greek symbols
6     implementation("com.vladsch.flexmark:flexmark-all:0.64.8") // MARKDOWN
7 }
```

# Gramática de Jflex para Código

```
-- TOKENS DE SIGNO --  
POTENCIA = ^  
MAS = +  
MENOS = -  
POR = *  
DIV = /  
PAR1 = (  
PAR2 = )  
IGUAL = =  
COMA = ,  
PUNTO = .
```

```
-- TOKENS PALABRAS RESERVADAS --  
PRINT = print  
FORMAT = format  
PLOT = plot  
REPORTE = reportes  
ERRORES = errores  
OPERADORES = OPERADORES
```

```
-- TOKENS USANDO EXPRESION REGULAR --  
BLANCOS = [\ \r\t\f\n]+  
ENTERO = [0-9]+  
DECIMAL = [0-9]+ "." [0-9]+  
IDENTIFICADOR = [a-zA-Z][a-zA-Z0-9_]*  
CADENA = [""]([^\"])*[""]
```

# Gramática de Cup para Código

```
-- TERMINALES --  
MAS, MENOS, POR, DIV, PAR1, PAR2, UMENOS,  
POTENCIA, IGUAL, COMA, PUNTO PRINT,  
FORMAT, PLOT, REPORTE, ERRORES, OPERADORES,  
CADENA, ENTERO, DECIMAL, IDENTIFICADOR  
  
-- NO TERMINALES --  
INICIO, INSTRUCCIONES, INSTRUCCION, IMPRESION,  
ASIGNARDECLARACION, EXPRESION, FORMATEAR,  
PLOTTEAR, REPOERROR, REPOPE
```

```
-- PRECEDENCIA --  
MAS, MENOS  
POR, DIV  
POTENCIA  
PAR1, PAR2  
UMENOS
```



-- GRAMATICA --

INICIO -> INSTRUCCIONES

INSTRUCCIONES -> INSTRUCCION

| INSTRUCCIONES INSTRUCCION

INSTRUCCION -> IMPRESION

| ASIGNARDECLARACION

| FORMATEAR

| PLOTEAR

| REPOERROR

| REPOPE

| error

IMPRESION -> PRINT PAR1 EXPRESION PAR2

ASIGNARDECLARACION -> IDENTIFICADOR IGUAL EXPRESION

FORMATEAR -> FORMAT PAR1 EXPRESION PAR2

PLOTEAR -> PLOT PAR1 EXPRESION COMA EXPRESION COMA EXPRESION PAR2

REPOERROR -> REPORTE PUNTO ERRORES PAR1 PAR2

REPOPE -> REPORTE PUNTO OPERADORES PAR1 PAR2

EXPRESION -> MENOS EXPRESION

| EXPRESION MAS EXPRESION

| EXPRESION MENOS EXPRESION

| EXPRESION POR EXPRESION

| EXPRESION DIV EXPRESION

| EXPRESION POTENCIA EXPRESION

| ENTERO

| DECIMAL

| CADENA

| IDENTIFICADOR

| PAR1 EXPRESION PAR2

# Gramática de Jflex para Texto

```
-- TOKENS CON EXPRESION REGULAR --  
ITALIC = \*[^*]+\*  
BOLD = \*\*[^\*]+\*\*  
BOLD_ITALIC = \*\*\*[^\*]+\*\*\*  
HEADER = \#{1,6} +[^\\n#][^\\n]*  
LISTA = \+ .*  
NUM_PUNTO = [0-9]+\.[^\\n]*  
PARRAFO = [a-zA-Z][^\\n]*  
BLANCOS = [\\ \\r\\t\\f\\n]+
```

# Gramática de Cup para Código

```
-- TERMINALES --  
HEADER, BOLD_ITALIC, BOLD, ITALIC  
LISTA, NUM_PUNTO, PARRAFO  
  
-- NO TERMINALES --  
INICIO, INSTRUCCIONES, INSTRUCCION  
TITULO, TEXTEAR, PBI, PB, PBI  
HEAD, CONTENIDO, LINEA  
LISTMAS, LISTNUM, LT, LN
```

```
-- GRAMATICA --  
INICIO -> INSTRUCCIONES  
  
INSTRUCCIONES -> INSTRUCCION  
                | INSTRUCCIONES INSTRUCCION  
  
INSTRUCCION -> TITULO  
              | TEXTEAR  
              | LISTMAS  
              | LISTNUM  
              | CONTENIDO  
              | error
```

TITULO -> PBI

| PB

| PI

PI = ITALIC

PB = BOLD

PBI = BOLD\_ITALIC

TEXTEAR -> HEAD

HEAD -> HEADER

LISTMAS -> LT

LT -> LISTA

LISTNUM -> LN

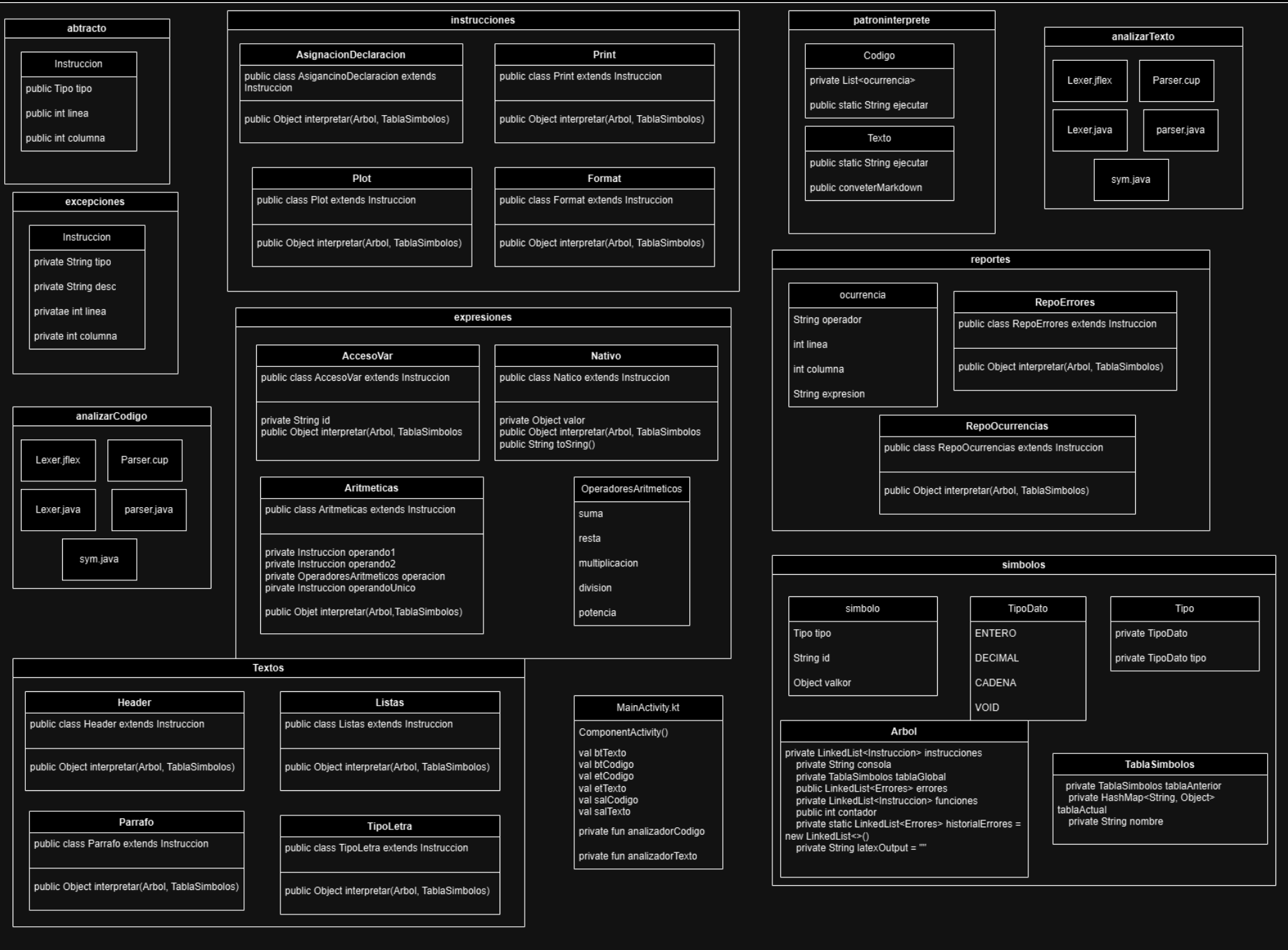
LN -> NUM\_PUNTO

CONTENIDO -> LINEA

LINEA -> PARRAFO

# DIAGRAMAS

main.java.com.example.notebook



# Funcionamiento

- **Creación de Notas:** El usuario puede escribir texto con formato Markdown.
- **Ejecutar Código Python:** Se permite la ejecución de fragmentos de código Python en las notas.
- **Visualización de Fórmulas:** Las expresiones matemáticas escritas en LaTeX se renderizan usando JLaTeXMath.