```python
from dataclasses import dataclass
from typing import List, Optional, Any

# Clase base para todos los nodos del AST
class ASTNode:
    pass

@dataclass
class Program(ASTNode):
    statements: List[ASTNode]

@dataclass
class Command(ASTNode):
    name: str
    arguments: List[Any]

@dataclass
class IfStatement(ASTNode):
    condition: ASTNode
    then_branch: List[ASTNode]
    else_branch: Optional[List[ASTNode]] = None

@dataclass
class WhileStatement(ASTNode):
    condition: ASTNode
    body: List[ASTNode]

@dataclass
class ForStatement(ASTNode):
    variable: str
    start: ASTNode
    end: ASTNode
    step: Optional[ASTNode] = None
    body: List[ASTNode]

@dataclass
class RepeatStatement(ASTNode):
    times: ASTNode
    body: List[ASTNode]

@dataclass
class BinaryOp(ASTNode):
    left: ASTNode
    op: str
    right: ASTNode

@dataclass
class UnaryOp(ASTNode):
    op: str
    operand: ASTNode
```

```
@dataclass
class Variable(ASTNode):
    name: str

@dataclass
class Literal(ASTNode):
    value: Any

@dataclass
class Assignment(ASTNode):
    variable: str
    value: ASTNode

@dataclass
class FunctionCall(ASTNode):
    name: str
    arguments: List[ASTNode]
```