# Árvore sintática abstrata

<algoritmo> ::= "inicio-algoritmo" <comando> "fim-algoritmo"

<smt\_enquanto> ::= "enquanto" "(" <cond> ")" "faca" <comando> "fim"

<smt\_se> ::= "se" "(" <cond> ")" "faca" <comando> "fim" (<smt\_se\_nao> | <smt\_ou\_entao>)?

<smt\_se\_nao> ::= "se nao" "(" <cond> ")" "faca" <comando> "fim" (<smt\_se\_nao> | <smt\_ou\_entao>)?

<smt\_ou\_entao> ::= "ou entao" <comando> "fim"

<smt\_decl\_variavel> ::= "declarar" <ident> "=" <expr>

<smt\_atr\_variavel> :: = <ident> "=" <expr>

<comando> ::= <smt\_enquanto>

| <smt\_se>

| <smt\_decl\_variavel>

| <smt\_atr\_variavel>

<expr> ::= <expr> "+" <expr>

| <expr> "-" <expr>

| <expr> "\*" <expr>

| <expr> "/" <expr>

| <expr> "^" <expr>

| <expr> "%" <expr>

| <cond>

| <termo>

<termo> ::= "(" <expr> ")"

| <numero>

| <ident>

<cond> ::= <expr> "ou" <expr>

| <expr> "e" <expr>

| <expr> ">" <expr>

| <expr> "<" <expr>

| <expr> ">=" <expr>

| <expr> "<=" <expr>

| <expr> "==" <expr>

| <expr> "!=" <expr>

<número> ::= [0-9]+

<ident> ::= [a-zA-Z\_][a-zA-Z0-9\_]\*

# Exemplo

inicio-algoritmo

se (x > 10) faca declarar y = 5 fim

fim-algoritmo

<algoritmo>

├── "inicio-algoritmo"

├── <comando>

│ └── <smt\_se>

│ ├── "se"

│ ├── "("

│ ├── <cond>

│ │ └── <expr> ">" <expr>

│ │ ├── <ident> (x)

│ │ └── <numero> (10)

│ ├── ")"

│ ├── "faca"

│ ├── <comando>

│ │ └── <smt\_decl\_variavel>

│ │ ├── "declarar"

│ │ ├── <ident> (y)

│ │ ├── "="

│ │ └── <expr> (5)

│ └── "fim"

└── "fim-algoritmo"