

Emiliano José Alexander Velásquez Najera

Carne: 201903848

# **Manual Técnico**

## **JPR Editor**

## Clase Instrucción

Clase Abstracta de la cual heredan todas las instrucciones del analizador.

Método	Parámetros	Descripción
Constructor	Fila, columna	Método con el cual se inicia la instrucción
Interpretar	tree, table	Método para interpretar la instrucción.
getNodo		Método que retorna el nodo en el árbol ast.

## Clase NodoAst

Clase que define los nodos del reporte para el árbol ast.

Método	Parámetros	Descripción
Constructor	Hijos, valor	Declara un nuevo nodo ast.
setHijos	hijos	Igual el valor del parámetros hijos con el del atributo hijos de la clase.
agregarHijo	hijo	Agrega un hijo a la lista de hijos del objeto.
agregarHijos	Hijos	Agrega una lista de hijos a la lista de hijos del objeto.
agregarHijoNodo	hijo	Agrega un nodo hijo al árbol de AST

## Clases que Heredan de Instrucción:

- AccesoArreglo
- Aritmetica
- Casteo
- Identificador
- Lógica
- Primitivos
- Read
- Relacional
- Asignación
- Break
- Case
- Continue
- DeclaracionArr1
- Declaración

- Decrecimiento
- Default
- For
- Función
- If
- Imprimir
- Incremento
- Llamada
- Main
- ModificarArreglo
- Return
- Símbolo
- Switch
- While

```
from abc import ABC, abstractmethod

class Instruccion(ABC):
    def __init__(self, fila, columna):
        self.fila = fila
        self.columna = columna
        self.arreglo = False
        super().__init__()

    @abstractmethod
    def interpretar(self, tree, table):
        pass

    @abstractmethod
    def getNodo(self):
        pass
```

Clase Instrucción

```
class NodoAST():
    def __init__(self, valor):
        self.hijos = []
        self.valor = valor

    def setHijos(self, hijos):
        self.hijos = hijos

    def agregarHijo(self, valorHijo):
        self.hijos.append(NodoAST(valorHijo))

    def agregarHijos(self, hijos):
        for hijo in hijos:
            self.hijos.append(hijo)

    def agregarHijoNodo(self, hijo):
        self.hijos.append(hijo)

    def agregarPrimerHijo(self, valorHijo):
        self.hijos.insert(0, NodoAST(valorHijo))

    def agregarPrimerHijoNodo(self, hijo):
        self.hijos.insert(0, hijo)

    def getValor(self):
        return str(self.valor)

    def setValor(self, valor):
        self.valor = valor

    def getHijos(self):
        return self.hijos
```

Clase NodoAST

## Clase Main

Clase donde se ejecuta la gramática y la interfaz grafica.

Método	Parámetros	Descripción
Constructor		Construye la ventana principal de la interfaz grafica.
agregar_menu		Agrega el menú archivo a la ventana principal
ejecutar		Ejecuta el interprete.
salir		Termina la ejecución de la aplicación
reporteError		Abre el reporte de errores
reporteTabla		Abre el reporte de Tabla de Simbolos
guardarComo		Guardar el archivo con el nombre indicado por el usuario
guardarArchivo		Guardar el archivo con el nombre actual.
abrir		Abre un nuevo archivo.

## Clase Grammar

Clase que define y parsea la gramática.

Método	Parámetros	Descripción
run		Ejecuta el parseo de la gramática.

Para más información de la gramática revisar el documento dedicado a la misma.