

PROYECTO 2

FRANCISCO MAGDIEL ASICONA MATEO 201801449

ORGANIZACIÓN DE LENGUAJES Y COMPILADORES 2

Manual de técnico

Frontend

En los siguientes componentes, todos realizan peticiones al servidor.

Módulos

Nombre	Detalle
App.tsx	Modulo principal para ejecutar los otros componentes
Compilador.tsx	Componente donde está el área de código y salida
Home.tsx	Componente de inicio, donde está la información de proyecto
ReporteAST.tsx	Componente donde se visualiza el árbol ast generado
ReporteError.tsx	Componente donde se visualiza la tabla de errores
TablaSimbolo.tsx	Componente donde se visualiza la tabla de símbolos

Backend

En los siguientes módulos, en algunos se recibe una petición del frontend y responde con un json.

Módulos

Nombre	Detalle
Instrucción.ts	Es una clase abstracta para heredarlo, y ahorrar Código
Nodo_Ast.ts	Nodo del árbol ast
Errores.ts	Plantilla para guardar los errores con sus atributos
Indentificador.ts	Clase para guardar un identificador y validar
Lógica.ts	Clase para validar la parte lógica del compilador
Nativo.ts	Clase donde se valida si un tipo de dato es nativo
Operación.ts	Clase donde se realizan las operaciones y se validan
Relaciones.ts	Clase donde se realiza las validaciones de las relacionales
Analizador.json	Modulo donde se construye la gramática
Instrucción_AccesoLista.ts	Clase donde se valida en acceso a una lista
Instrucción_AccesoVector.ts	Clase donde se valida el acceso a un vector
Instrucción_AgregarLista.ts	Clase donde se agrega un dato una lista y se valida
Instrucción_Asignacion.ts	Clase donde se valida la asignación
Instrucción_AsignacionLista.ts	Clase donde se asigna una lista o se crea y se valida
Instrucción_AsignacionVector.ts	Clase donde se asigna un vector o se crea y se valida

Instrucción_Break.ts	Clase donde se valida el break
Instrucción_Casteo.ts	Clase donde se valida y realiza el casteo
Instrucción_Continue.ts	Clase donde se valida el continue
Instrucción_Declaracion.ts	Clase donde se valida y realiza la declaración
Instrucción_DeclaracionLista.ts	Clase donde se valida la declaración de una lista
Instrucción_DeclaracionVectores.ts	Clase donde se valida y realiza la declaración de un vector
Instrucción_Decremento.ts	Clase donde se valida y se realiza el decremento
Instrucción_Funciones.ts	Clase donde se valida una función
Instrucción_FunNativas.ts	Clase donde se valida si es una función nativa
Instrucción_Incremento.ts	Clase donde se valida y realiza el incremento
Instrucción_LlamadaFunMetodos.ts	Clase donde se valida la llamada de un método
Instrucción_Metodos.ts	Clase donde se valida un método
Instrucción_Print.ts	Clase donde se valida el método print
Instrucción_Println.ts	Clase se valida el método print pero con un salto de línea
Instrucción_Return.ts	Clase donde se valida el return
Instrucción_Run.ts	Clase donde se valida el run y se ejecuta
Ciclo_DoWhile.ts	Clase donde se valida y se realiza el ciclo dowhile
Ciclo_For.ts	Clase donde se valida y se realizar el ciclo for
Ciclo_While.ts	Clase donde se valida y se realiza el ciclo while
Condición_If.ts	Clase donde se valida la condición if y se asigna
Condición_IfTernario.ts	Clase donde se valida la condición if ternario y se asigna
Condición_Switch.ts	Clase donde se valida la condifcion switch
Condición_SwitchCase.ts	Clase donde se valida cada caso del switch
Condición_SwitchDefault.ts	Clase donde se valida el valor si no va a ningún caso
Reporte_CambiarTipo.ts	Clase para cambiar el tipo de dato
Reporte_Graficar.ts	Para graficar el árbol ast
Reporta_ReporteTabla.ts	Para generar la tabla de símbolos
Símbolo_Arbol.ts	Clase que contiene la estructura del árbol ast
Símbolo_Simbolo.ts	Plantilla para una array cuando se guarda los símbolos
Símbolo_TablaSimbolo.ts	Clase para generar la tabla de símbolos
Símbolo_Tipo.ts	Clase para retornar el tipo de método
IndexControllers.ts	Clase que gestiona todas las peticiones que se realizan desde el front