

AritIDE- Manual de técnico

Contenido

Descripción de aplicación y requerimientos mínimos	1
Herramientas y librerías utilizadas.....	2
Diagrama de flujo	2
Diagramas de paquetes principales	3
Diagrama de clases.....	4
Descripción de clases principales	5
ErrorHandler.....	5
NodeError	5
TypeError.....	5
NodeInfo.....	5
FileManager	5
AstNode.....	5
AritLanguage	5
Expression	5
Scope	5
FileScope	5
TypeFacade	5

Descripción de aplicación y requerimientos mínimos

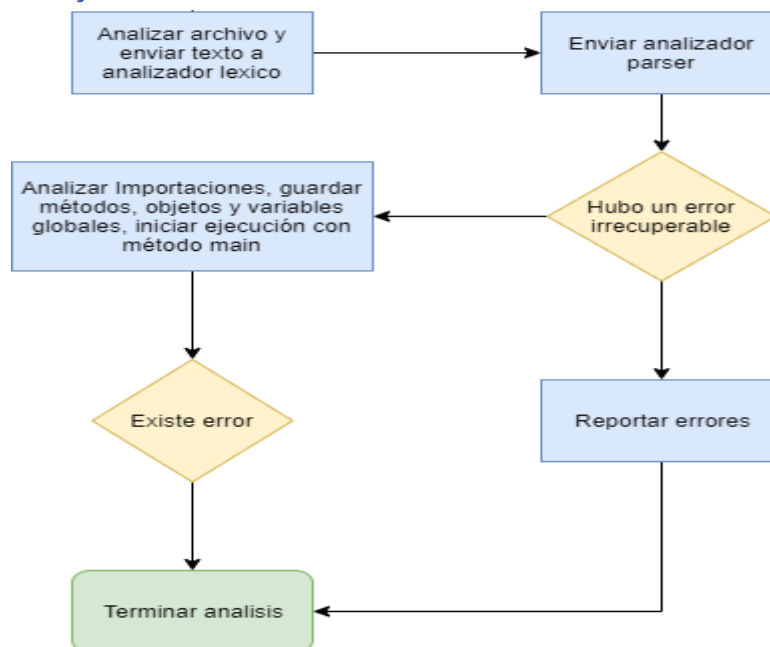
Arit Software es un entorno y lenguaje de programación con un enfoque en el análisis estadístico, el cual está formado por un conjunto de herramientas muy flexibles que pueden ampliarse fácilmente por nuestras propias funciones. Este software también cuenta con un entorno de desarrollo integrado (IDE) que proporciona servicios integrales para facilitarle al programador el desarrollo de aplicaciones. Para el funcionamiento correcto de la aplicación se necesita lo siguientes requerimientos mínimos:

- Sistema operativo: Windows 7 o posterior
- Tener instalado Java versión 1.8.
- Memoria: 1 GB de RAM.

Herramientas y librerías utilizadas

- **IntelliJ IDEA 2019.3.2:** IDE utilizado
- **JAVA 1.8:** Lenguaje utilizado
- **JavaFX:** es un conjunto de paquetes de gráficos y medios que permite a los desarrolladores diseñar, crear, probar, depurar e implementar aplicaciones de cliente enriquecido que operan de forma consciente en diversas plataformas.
- **JFlex 1.7.0:** JFlex es un generador de analizador léxico (también conocido como generador de escáner) para Java, escrito en Java.
- **Cup-11b:** CUP significa Construcción de analizadores útiles y es un generador de analizador LALR para Java.
- **JavaCC-7.0.5:** Java Compiler Compiler (JavaCC) es un generador de analizador para usar con aplicaciones Java.
- **commons-io-2.6:** Librería para el manejo de archivos y rutas sin importar el sistema operativo.
- **Richtextfx:** RichTextFX proporciona un área de texto de memoria eficiente para JavaFX que permite al desarrollador diseñar rangos de texto, mostrar objetos personalizados en línea (no más HTML editor) y anular el comportamiento predeterminado solo cuando sea necesario sin anular ninguna otra parte del comportamiento.

Diagrama de flujo



Diagramas de paquetes principales

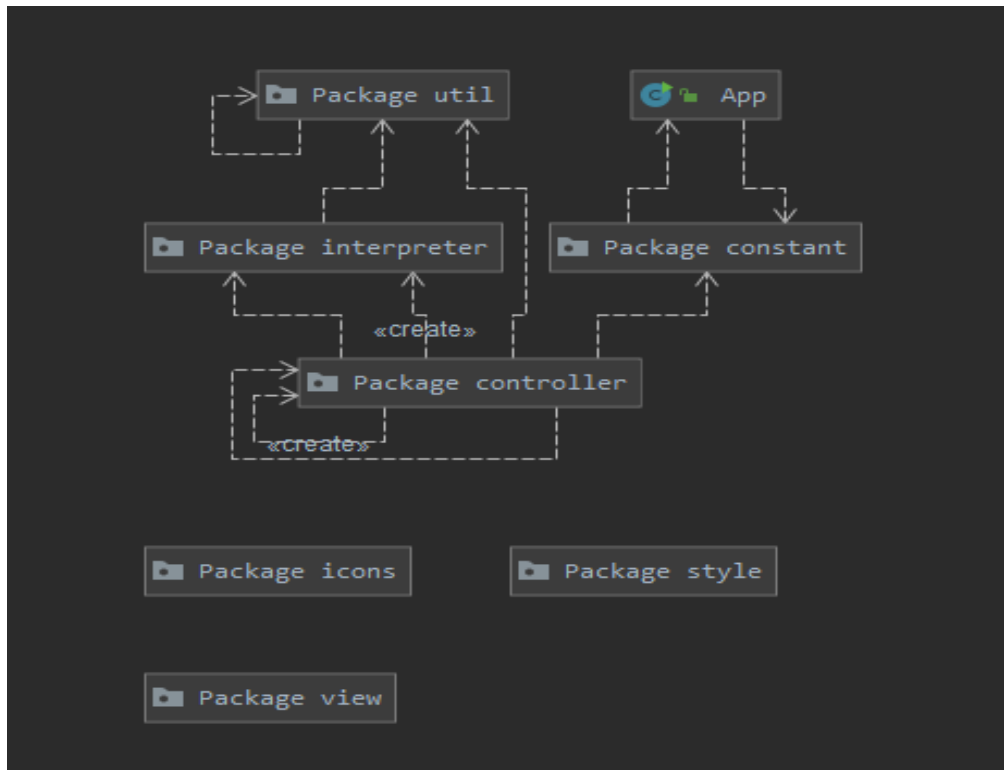
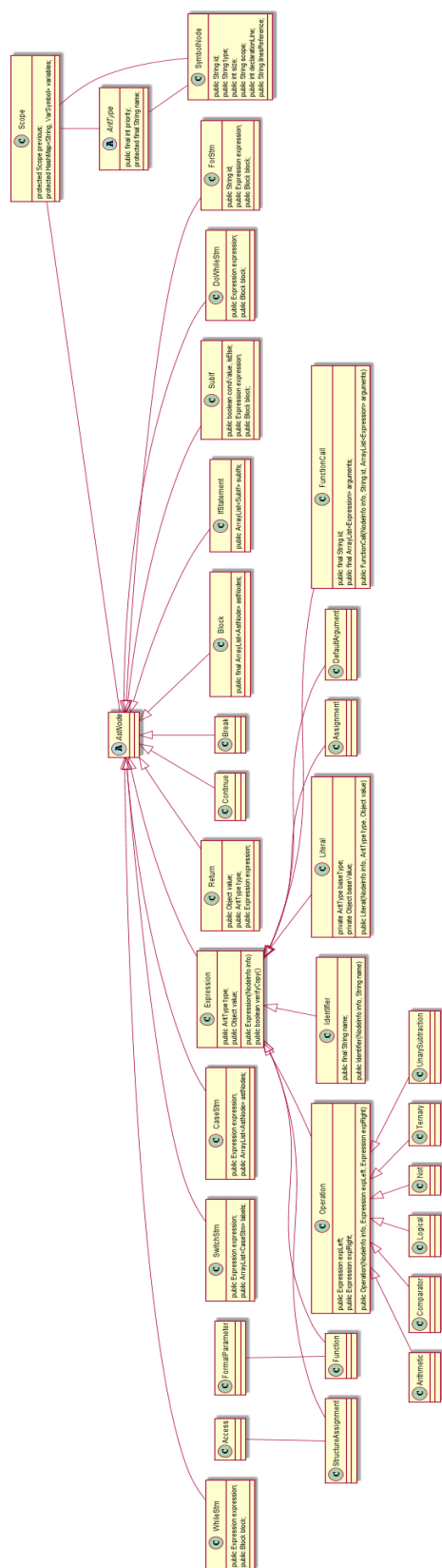


Diagrama de clases



Descripción de clases principales

ErrorHandler

Clase que se encarga de ir guardando los errores encontrados durante el análisis en una lista pública y estática.

NodeError

Clase nodo que guarda todos los campos necesarios para indicar por qué ocurrió el error.

TypeError

Clase donde se especifica el tipo de error (léxico, sintáctico, semántico).

NodeInfo

Clase nodo que guarda la línea, columna y nombre del archivo que está analizando actualmente.

FileManager

Clase encargada del manejo de archivos y directorios. (Crear, eliminar, sobrescribir, etc.).

AstNode

Clase nodo, se utiliza para generar el árbol de análisis sintáctico. Esta clase es abstracta ya que lo hereda cada una de las clases necesarias para la ejecución del programa.

AritLanguage

Clase que tiene una lista de nodos que son recorridos para ejecutar el programa.

Expression

Clase abstracta encargada de calcular todo tipo de operaciones encontradas durante el análisis.

Scope

Clase encargada de guardar las variables encontradas (Tabla de símbolos).

FileScope

Clase que hereda de Scope, esta clase guarda los métodos, fusión y variables globales.

TypeFacade

Controlador de tipos del lenguaje