Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela Ciencias y Sistemas
Laboratorio OLC2
Sección A+

MANUAL DE TECNICO

Manuel Antonio Fuentes Fuentes
201213580
20 de Marzo del 2019

INDICE

DESCRIPCION **PAGINA**

1. Descripción de la Aplicación

- **Creator XML:** Es un generador de interfaces de usuario UI utilizando plantillas escritas en un lenguaje basado en XML. Este se integrará por medio del lenguaje de programación Funcion Script, este lenguaje es una versión parecida a el lenguaje de programación JavaScript.
  Este lenguaje nos servirá para poder generar, recolectar y almacenar los datos de cada una de las interfaces que se necesiten crear.

Componentes de la aplicación:

- **FuncionScript(FS):** FuncionScript es un lenguaje de programación basado en el paradigma fuctional programing, este es un paradigma de programación declarativo, es decir, los algoritmos son desarrollados únicamente usando expresiones (lógicas, aritméticas y relaciones). Este lenguaje permitirá utilizar el motor Creator XML para poder generar interfaces de usuario (UI) a partir de plantillas XML escritas en Generic XML. Funcional Script podrá transformar los datos ingresados en las interfaces de usuario (UI) generadas a partir de las plantillas XML, los datos procesados con FuncionScript serán convertidos de nuevo en un archivo XML. FuncionScript será capaz de leer los datos que han sido almacenados en archivos XML y a partir de estos permitirá la generación de reportes. 1XML es un meta lenguaje que se utiliza para almacenar datos de forma legible.

- **GenericXML (GX):** GenericXML será un lenguaje basado en XML, este se utilizará para definir las interfaces de usuario (UI). Cada interfaz de usuario tendrá también un archivo GenericXML que servirá para almacenar los datos que fueran capturados por medio de la interfaz gráfica.

- **ExtremeEditor (EE):** Este es el ide de Funcional Script, este cuenta con varias características importantes para la ejecución del lenguaje GenericXML y Funcion Script, en el ide podemos crear una pestaña con la cual vamos a poder ingresar el código que necesitemos analizar, teniendo la característica que cada lenguaje posee distintas palabras reservadas así como componentes se hace un énfasis en estas de manera que cada tipo de palabra cambia de color, dentro de las funcionalidades que vamos a poder realizar en el editor son las siguientes:

- Abrir Archivo
- Guardar
- Guardar Como
- Crear Pestaña

- Cerrar Pestaña
- Reporte Errores Léxico
- Reporte Errores Sintácticos
- Reporte Errores Semánticos

Este editor cuenta con dos secciones importantes, el área de pestañas y el área de consola, esta consola nos servirá para visualizar los valores que se imprimen para poder ver que esta sucediendo en la ejecución de los lenguajes.

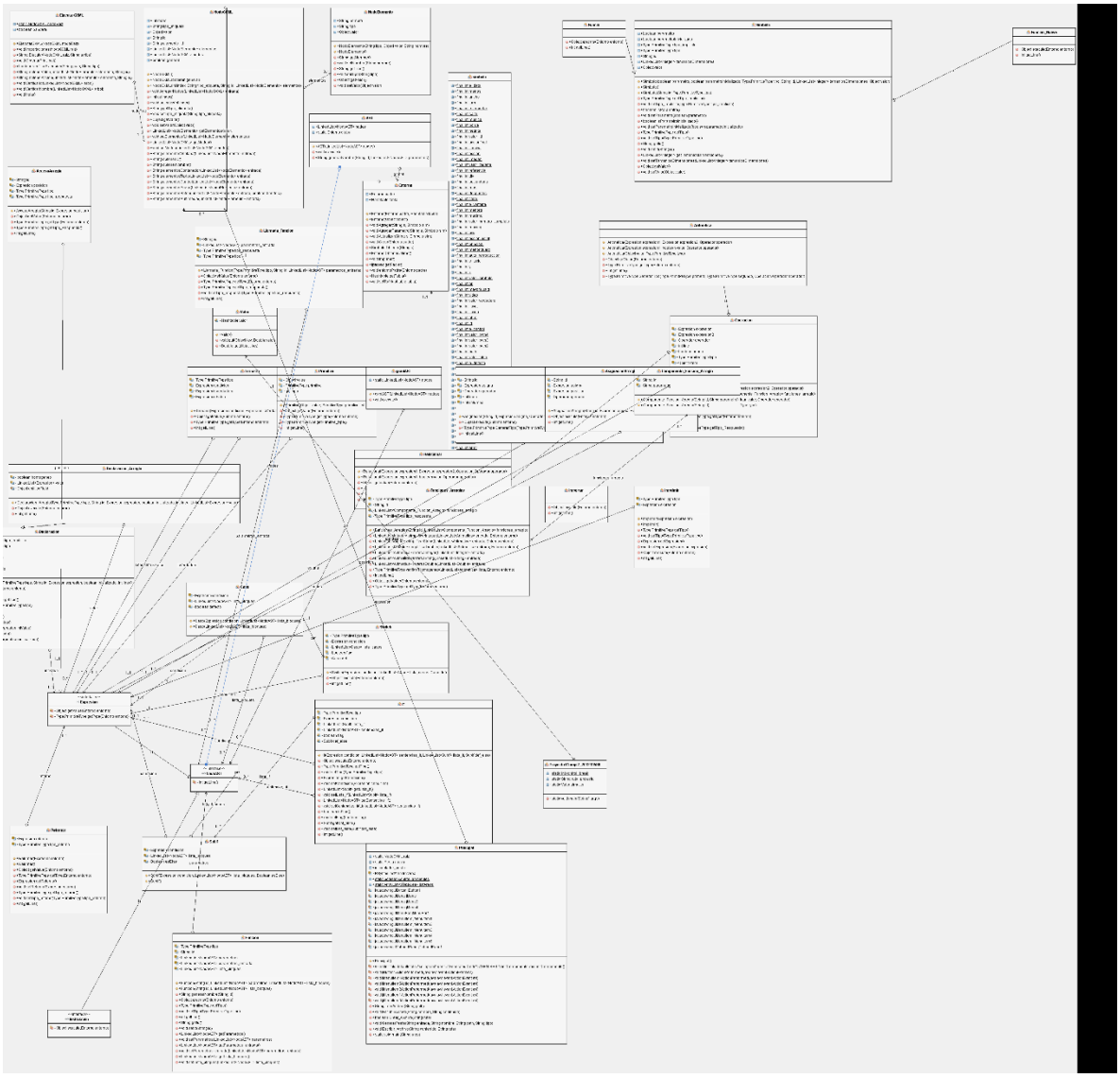
2. Requerimientos Mínimos

Los requerimientos mínimos del proyecto son funcionalidades del sistema que permitirán un ciclo de ejecución básica, para tener derecho a calificación se deben cumplir con lo siguiente:

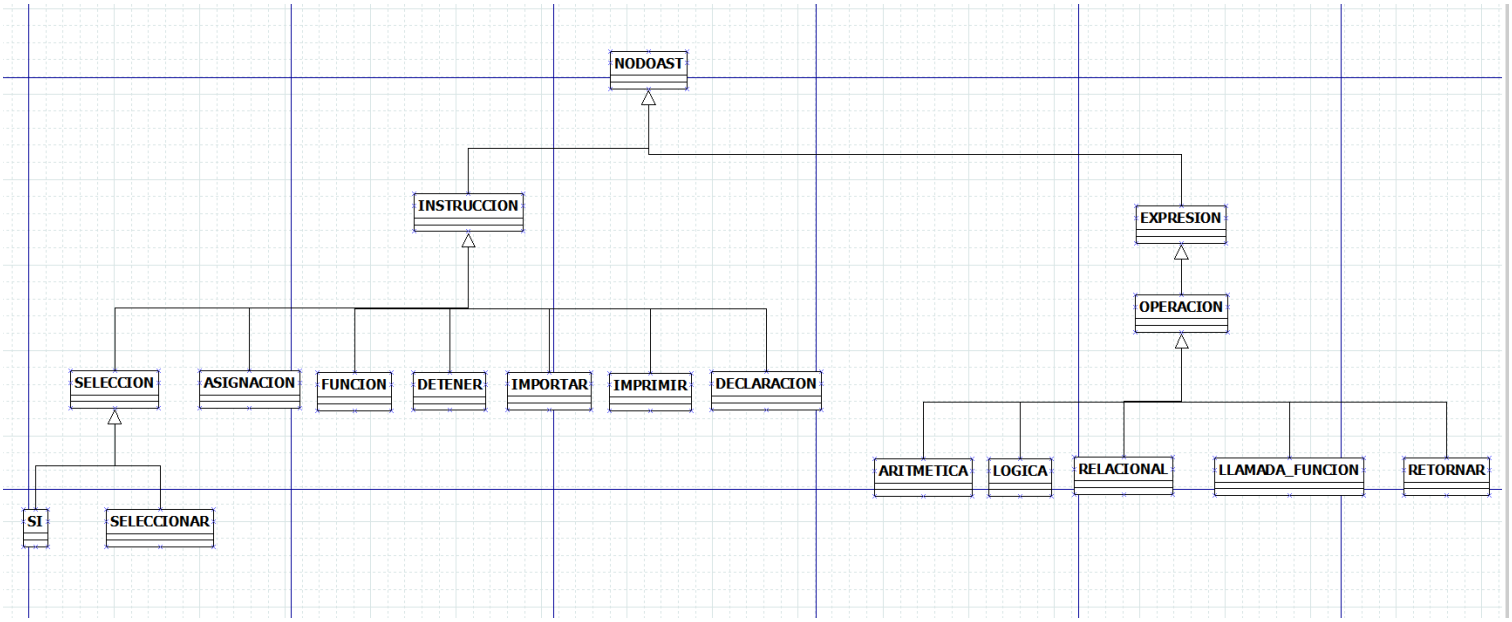
- Extreme Editor (Todos los componentes del IDE)
- GenericXML
- Etiqueta Texto
- Etiqueta dato
- Etiqueta Lista de datos
- Etiqueta enviar
- Etiqueta Botón
- Etiqueta Ventana
- FuncionScript
- Declaración de variables
- Asignación de variables
- Operaciones aritméticas
- Operaciones relacionales
- Operaciones lógicas
- Arreglos
- Imprimir
- Sentencias de selección
- Sentencias de transferencia
- Llamada a procedimientos y funciones
- Funciones con arreglos propias de función Script
- Ordenamiento descendente
- Ordenamiento ascendente
- Máximo
- Mínimo
- Funciónfiltrar
- Funciónbuscar
- Función map
- Función reduce
- Función todos

- Función algunos
- Funciones nativas de la interfaz
- LeerGxml
- Obtener por etiqueta
- ObtenerporID
- ObtenerPorNombre
- CrearTexto
- CrearVentana
- CrearContenedor
- CrearCajaTexto
- CrearAreaTextooEventos
- Al Clic

3. Diagrama de Clases



4. Diagrama de Herencia



5. Descripción de herramientas utilizadas

- **NetBeans:** Es un ide de desarrollo hecho principalmente para el lenguaje de programación Java, este se encuentra disponible para el sistema operativo Windows o Linux, NetBeans es un proyecto de código abierto de gran éxito con una gran base de usuarios, una comunidad en constante crecimiento.

Versión Utilizada: NetBeans 8.2

Dirección Descarga: https://netbeans.org/downloads/8.0.2/

- **Java:** Java es un lenguaje de programación de propósito general, concurrente, orientado a objetos, que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible.

Versión Utilizada: JAVA 1.8.0
Dirección Descarga: https://www.java.com/es/download/

- **Jflex:** Un generador de analizadores léxico toma como entrada una especificación con un conjunto de expresiones regulares y acciones correspondientes. Genera un programa que lee la entrada, compara la entrada con las expresiones regulares en el archivo de especificaciones y ejecuta la acción correspondiente si una expresión regular coincide.

Versión Utilizada: Jflex 1.6.1
Direccion Descarga: https://www.jflex.de/download.html

- **Cup:** CUP significa Construcción de analizadores útiles y es un generador de analizador LALR para Java. Implementa la generación de analizadores LALR (1) estándar .

Versión Utilizada: CUP 0.11
Dirección Descarga: http://www2.cs.tum.edu/projects/cup/

- **RSyntaxArea:** RSyntaxTextArea es un componente de texto de plegado de código y resaltado de sintaxis para Java Swing. Extiende JTextComponent para que se integre completamente con el paquete estándar javax.swing.text. Es rápido y eficiente, y puede usarse en cualquier aplicación que necesite editar o ver el código fuente.

Version Utilizada: RSyntaxTextArea 2.6.1
Direccion Descarga: Http://bobbylight.github.io/RSyntaxTextArea/

6. Descripción de clases y métodos principales

Clases y Metodos Generic XML:

Clase NodoGXML: Esta clase es el nodo que forma el árbol AST que nos sirve para la generación de código Función Script, dentro de los atributos que tenemos en esta clase son: Índex, tipo_respuesta, valor, id, elemento_id, elementos, nodos, general. Los atributos elementos y nodos son de tipo lista y nos ayudaran a poder almacenar otros nodos u otros elementos para poder armar el Árbol AST.

- **Metodo NodoGXML**: Este es el constructor de la clase, este tiene la característica que hace uso del polimorfismo y nos permite definir varios constructores para poder inicializar cada uno de los atributos de la clase de manera correcta.

- **Metodo AgregarNodos**: este método nos permite agregar los nodos de otra clase - NodoGXML conservando los que tenemos actualmente.

- **Metodo elementosVentana:** este método nos permite verificar los parámetros de un elemento ventana.

- **Metodo obtenerID:** nos permite obtener un id de algún nodo que este dentro de la estructura del árbol AST.

Clase EjecutarGXML: Esta clase nos permite realizar la traducción del lenguaje de etiquetas GenericXML al lenguaje Funcion Script, este es la parte principal de la ejecución.

- **Método Importaciones:** Este método ejecuta las importaciones de los archivos GXML y Función Script para su posterior traducción.

- **Método Ejecutar:** En este método es donde se realiza la traducción de cada uno de los elementos al lenguaje Función Script, es un método recursivo ya que se tiene que hacer un recorrido de profundidad al árbol AST que se elaboró en la gramática de GXML.

- **Métodos Verificar**: Estos métodos nos permiten verificar si cada una de las etiquetas cumple con los elementos mínimos y obligatorios para poder realizar la ejecución.

Función Script:

Entorno: Esta clase es la que maneja los entorno y tabla de símbolos del análisis de Función Script, este tiene los atributos Padre y Tabla de Símbolos, el atributo padre es un puntero a otro entorno que seria el entorno que este mas arriba de la ejecución.

- **Método Agregar:** Este método nos permite agregar un símbolo nuevo a la tabla de símbolos, con la característica que verifica si el símbolo ya existe.
- **Método Obtener:** Este método nos devuelve un símbolo en base a una comparación con el id que le enviemos.
- **Método Actualizar:** Este método nos permite actualizar el contenido de un símbolo que este dentro del entorno.
- **Método Obtener Ultimo:** Este método nos permite obtener el entorno que este mas arriba, el entorno global.

NodoAST: Esta clase es el nodo del árbol AST que se crea para la ejecución de Función Script, esta es una interfaz de la cual las clases instrucción y expresión van a implementar sus métodos.

Instrucción: Esta es una interfaz que implementa los métodos que se definen en NodoAst con la diferencia que esta interfaz va a ser la que van a heredar clases como Si, Selecciona, Función. Todas estas clases tienen las características que no retornan un valor en específico, sino que solo ejecutan una serie de acciones.

- **Método Ejecutar:** Este método se tiene presente en todas las clases que hereden de la clase instrucción por lo que aquí se van a realizar las acciones correspondientes para la ejecución de todo el flujo de instrucciones.

Expresión: Esta es una interfaz que implementa los métodos que se definen en NodoASt con la diferencia que de esta clase van a heredar clases como Aritmetica, Logica, Relacional,

Operación, ya que estas son clases que van a retornar un valor en especifico por lo que hay que saber cual es su valor de retorno y su tipo de retorno.

- **Método getValue:** Dentro de este método vamos a ejecutar las acciones necesarias para poder enviar como retorno un valor primitivo.
- **Método getType:** En este método luego de enviar el valor vamos a poder retornar el tipo del cual es la llamada.