

Universidad de San Carlos de Guatemala

Facultad de Ingeniería

Escuela de Ciencias y Sistemas

Introducción a la programación y Computación 1

Segundo Semestre 2022

Catedrático: Kevin Adiel Lajpop AjpacajaTutor

académico: Moises Gonzalez Fuentes

Proyecto 1

Manual técnico

Nombre: Josue Daniel Solis Osorio

Carnet: 202001574

Fecha: 03/11/2022

Introducción

El presente documento describe los aspectos técnicos informáticos del sistema de información. El documento familiariza las personas que utilicen el programa para su correcta ejecución.

Objetivo de los procedimientos

El curso de Organización de Lenguajes y Compiladores 1, ha puesto en marcha un nuevo proyecto, requerido por la Escuela de Ciencias y Sistemas de la Facultad de Ingeniería, que consiste en crear un lenguaje de programación para que los estudiantes, del curso de Introducción a la Programación y Computación 1, aprendan a programar y tener conocimiento de todas las generalidades de un lenguaje de programación. Cabe destacar, que este lenguaje será utilizado para generar sus primeras prácticas de laboratorio del curso antes mencionado.

Por lo tanto, a usted, que es estudiante del curso de Compiladores 1, se le encomienda realizar el proyecto llamado MFMScript, dado sus altos conocimientos en temas de análisis léxico, sintáctico y semántico.,

Arquitectura General del proyecto

Hoy en día, se ha dado gran importancia al uso de tecnologías de contenedores, lo que otorga como ventajas; rapidez en el despliegue y facilidad de mantenimiento en un servidor. Para el presente proyecto, se le propone manejar una arquitectura Cliente-Servidor, con el objetivo de que pueda separar los servicios administrados por el intérprete, de la aplicación cliente que se mostrará al usuario final.

Requisitos mínimos

Windows

- *Windows 10 (8u51 y superiores)
- *Windows 8.x (escritorio)
- *Windows 7 SP1
- *Windows Vista SP2
- *Windows Server 2008 R2 SP1 (64 bits)
- *Windows Server 2012 y 2012 R2 (64 bits)
- *RAM: 128 MB
- *Espacio en disco: 124 MB para JRE; 2 MB para Java Update
- *Procesador: Mínimo Pentium 2 a 266 MHz
- *Exploradores: Internet Explorer 9 y superior, Firefox

Mac OS X

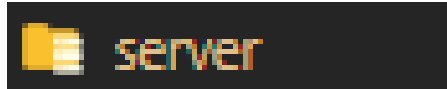
- *Mac con Intel que ejecuta Mac OS X 10.8.3+, 10.9+
- *Privilegios de administrador para la instalación
- *Explorador de 64 bits
- *Se requiere un explorador de 64 bits (Safari, por ejemplo) para ejecutar Oracle Java en Mac.

Linux

- *Oracle Linux 5.5+1
- *Oracle Linux 6.x (32 bits), 6.x (64 bits)2
- *Oracle Linux 7.x (64 bits)2
(8u20 y superiores)
- *Red Hat Enterprise Linux 5.5+1 6.x (32 bits), 6.x (64 bits)2
- *Red Hat Enterprise Linux 7.x (64 bits)2
(8u20 y superiores)
- *Suse Linux Enterprise Server 10 SP2+, 11.x
- *Suse Linux Enterprise Server 12.x (64 bits)2
(8u31 y superiores)
- *Ubuntu Linux 12.04 LTS, 13.x
- *Ubuntu Linux 14.x (8u25 y superiores)
- *Ubuntu Linux 15.04 (8u45 y superiores)
- *Ubuntu Linux 15.10 (8u65 y superiores)

Lógica del programa

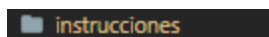
- Servidor: Aquí se encuentra el back end



- SRC: Aquí se encuentran las instrucciones, gramáticas, código fuente para el back end



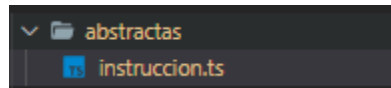
- Instrucciones: Aquí se encuentran las instrucciones que se realizan al momento de ejecutar



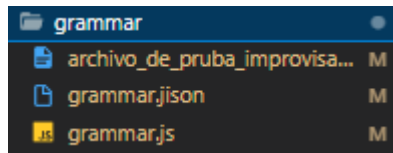
- Elementos en instrucciones: Estas son todas las ejecuciones que realiza el programa al momento de ejecutar dándole así un sentido, su nombre describe la función que este realiza

acceso_vector_1D.ts	M	lista_case.ts	M	string.ts	M
acceso_vector_2D.ts	M	lista_identificadores.ts	M	switch.ts	M
asignacion.ts	M	lista_instrucciones.ts	M	ternario.ts	M
bloque.ts	U	lista_parametros.ts	M	typeof.ts	M
break.ts	M	lista_s1.ts	M	upper.ts	M
case.ts	M	lista_valores.ts	M	vector_declarado_1D.ts	M
casteo.ts	M	llamada_sin_parametros.ts	M	vector_declarado_2D.ts	M
chararray.ts		llamada.ts	M	vector_modificar_1D.ts	M
ciclo.ts		lower.ts	M	vector_modificar_2D.ts	M
continue.ts	M	metodo_void.ts	M	while.ts	M
declaracion_lista_vector_1D.ts	M	metodo.ts	M		
declaracion_lista_vector_2D.ts	M	operacionBinaria.ts	M		
declaracion.ts	M	parametros.ts	M		
default.ts	M	pop.ts	M		
do_until.ts	M	primitivos.ts	M		
do_while.ts	M	print.ts	M		
elif.ts	M	println.ts	M		
else.ts	M	push.ts			
for.ts	M	return.ts	M		
funcion.ts	M	round.ts	M		
if_lista.ts	M	run_sin_parametros.ts			
if.ts	M	run.ts			
incremento.ts	M				
length.ts					

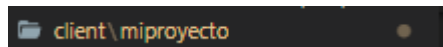
- Clase Padre: Ya que para este proyecto se utiliza herencia esta es la clase padre



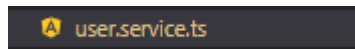
- Grammar: Aquí se trabajo la gramática usando la librería jison



- Client: Aquí se encuentra toda la parte del front – end para este proyecto se utilizo angular



- User.Services : Aquí se aceptan las peticiones del back end



- APP.DATA: Aquí se mandan las petición hacia el front end



ENLACE A GITHUB:

<https://github.com/JosueSolis121201/OLC1-202001574.git>

