

MANUAL DE USUARIO PROYECTO2

Selvin Orlando Hernandez
201700603

ING. KEVIN LAJPOP
AUX. SANDRA EUNICE JIMENEZ

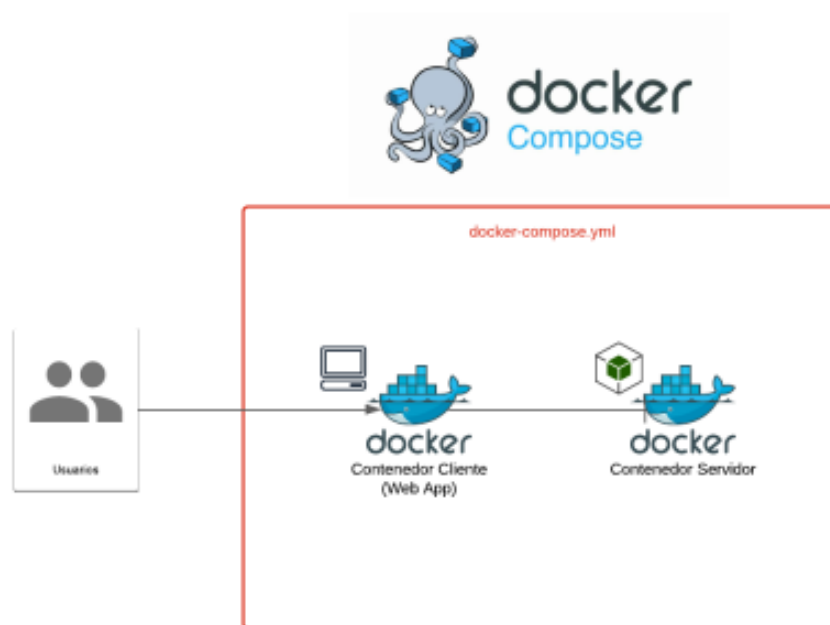
ORGANIZACION DE LENGUAJES Y COMPILADORES 1C

Objetivos de la aplicación:

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 Computación1, 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.

Arquitectura implementada

Arquitectura Cliente-Servidor con Docker



Interfaz de la aplicación web



1. **Compilar:** Este botón se encarga de llamar al interprete el cual se hará cargo del análisis léxico, sintáctico y semántico en la aplicación.
2. **T_Simbolos:** Este botón tiene la función de mostrar la tabla de símbolos en el área de reportes de la aplicación.
3. **Reporte Errores:** Este botón tiene el trabajo de mostrar la tabla de errores sintácticos, léxicos y semánticos en el área de reportes.
4. **AST:** Este botón tiene la función de mostrar el Árbol de análisis sintáctico en el área de reportes.
5. **Guardar:** Este botón se encarga de almacenar el contenido del Editor (7) en un archivo con extensión sc
6. **Seleccionar Archivo:** Este bonton se encarga de mostrar un cuadro de dialogo que permite elegir un archivo con extensión sc para cargarlo al editor

7. **Editor:** En esta sección se puede agregar el código a analizar, se puede agregar mediante el botón Seleccionar Archivo (6), o copiando y pegando directamente en el editor.
8. **Consola:** En este apartado aparecerá el resultado del código compilado, o bien los errores sintácticos o semánticos.

Descripción del lenguaje: Este no distingue entre mayúsculas y minúsculas

```
// Ejemplo

int a=0;
INt A=0;

//Debe dar error la declaración de "A" ya que la variable "a" ya existe previamente
//int es lo mismo que INt
```

- Comentario de una línea (1)
- Comentario multilinea (2)

```
// Este es un comentario de una línea 1

/*
    Este es un comentario
    Multilínea
    Para este lenguaje
*/ 2
```

Tipos de Datos

TIPO	DEFINICION	DESCRIPCION	EJEMPLO	OBSERVACIONES	DEFAULT
Entero	Int	Este tipo de datos aceptará solamente números enteros.	1, 50, 100, 25552, etc.	Del -2147483648 al 2147483647	0
Doble	Double	Admite valores numéricos con decimales.	1.2, 50.23, 00.34, etc.	Se manejará cualquier cantidad de decimales	0.0
Booleano	Boolean	Admite valores que indican verdadero o falso.	True, false	Si se asigna un valor booleano a un entero se tomará como 1 o 0 respectivamente.	True
Caracter	Char	Tipo de dato que únicamente aceptará un único carácter, y estará delimitado por comillas simples. ''	'a', 'b', 'c', 'E', 'Z', '1', '2', '^', '%', ' ', '=', '!', '&', '/', '\\', 'n', etc.	En el caso de querer escribir comilla simple escribir se escribirá \ y después comilla simple \, si se quiere escribir \ se escribirá dos veces \\, existirá también \n, \t, \r, \".	'\u0000' (carácter 0)
Cadena	String	Es un grupo o conjunto de caracteres que pueden tener cualquier carácter, y este se encontrará delimitado por comillas dobles. ""	"cadena1", "-- ** cadena 1"	Se permitirá cualquier carácter entre las comillas dobles, incluyendo las secuencias de escape: \" comilla doble \\ barra invertida \\n salto de línea \\r retorno de carro \\t tabulación	"" (string vacío)

Secuencias de escape

SECUENCIA	DESCRIPCION	EJEMPLO
\\n	Salto de línea	"Hola\\nMundo"
\\	Barra invertida	"C:\\miCarpeta\\Personal"
\"	Comilla doble	"\"Esto es una cadena\""
\\t	Tabulación	"\\tEsto es una tabulación"
\"	Comilla Simple	"\"Estas son comillas simples\""

Símbolos del Sistema

Símbolo	Descripción	Tipo
+	Este símbolo permite realizar sumas o concatenaciones	Aritmética
-	Este símbolo permite realizar restas	
*	Este símbolo permite realizar multiplicaciones	
/	Este símbolo permite realizar divisiones	

^	Este símbolo permite realizar potencias	
%	Este símbolo permite encontrar el residuo de una división	
==		RELACIONAL
!=	Compara si dos expresiones son iguales	
<	compara si la expresión izquierda es menor que la derecha	
<=	compara si la expresión de la izquierda es menor o igual que la derecha	
>	compara si la expresión de la izquierda es mayor que la derecha	
>=	compara si la expresión de la izquierda es mayor o igual que la derecha	
	OR: compara expresiones lógicas y si al menos una es verdadera retorna verdadero en otro caso retorna falso	LOGICA
&&	AND: compara expresiones lógicas, retorna verdadero siempre que las dos expresiones sean verdaderas en otro caso retorna falso	
!	NOT: devuelve el valor inverso de una expresión lógica, si su valor es verdadero devuelve falso y si es falso devuelve verdadero	

Requisitos de instalación/uso	
Navegador	Opera, Firefox, Chrome, Safari
Sistema Operativo	Windows, Mac, Linux
Puerto de conexión	Localhos:3000