Objetivos generales

- Familiarizar al estudiante con la herramienta JFlex
- Familiarizar al estudiante con la herramienta CUP
- Aplicar conocimientos de análisis léxico y sintáctico en la construcción de un intérprete.

Objetivos específicos

- Creación de archivos de configuración para JFlex.
- Creación de archivos de configuración para CUP.
- Combinar la funcionalidad de JFlex y Cup en aplicaciones reales.
- Familiarizar al estudiante con el desarrollo de aplicaciones para Android.

Descripción de la actividad

Los dispositivos móviles hoy en día se han convertido en la herramienta a la que más personas tienen acceso, dada esta característica como profesional de sistemas es importante conocer el proceso de desarrollo de soluciones en este tipo de plataformas.

Por otro lado, los diagramas de flujo son herramientas fundamentales en el diseño de algoritmos y la programación estructurada.

Descrito lo anterior se le pide al estudiante realizar una aplicación para Android que permita representar pseudocodigo con un diagramas de flujo.

Estructura del lenguaje

Nuestro lenguaje consta de 2 secciones importantes y los comentarios (que pueden estar definidos en cualquier parte del lenguaje), separadas por la expresión

%%%%

Las dos secciones se describen a continuación:

Sección de definición de algoritmo

Esta es una sección en la cual podremos definir un algoritmo en pseudocódigo que será representado en un diagrama de flujo. Ejemplo:

```
INICIO
    VAR a = 10
    VAR b = 20
    SI (a < b) ENTONCES
        MOSTRAR "a es menor que b"
    FINSI
    MIENTRAS (a < 15) HACER
        a = a + 1
        MOSTRAR a
    FINMIENTRAS
    MOSTRAR "Fin del programa"
FIN
```

Sección de configuración de diagrama de flujo

En esta sección se pueden configurar aspectos relacionados con la creación del diagrama de flujo que representa al algoritmo.

A través de expresiones se pueden definir formas, colores, tamaño de letra, entre otros, para cada figura que forma parte del diagrama de flujo. Ejemplo:

```
%DEFAULT=1
%COLOR_TEXTO_SI=12,45-5,1|1
%FIGURA_MIENTRAS=CIRCULO|1
%DEFAULT=3
```

Definición del lenguaje en pseudocodigo

Definición de operadores aritméticos

Dentro del lenguaje se podrán utilizar expresiones aritméticas que incluyen literales numéricas e identificadores de variables.

Variables y asignación de variables

Dado que es un lenguaje muy simple, las variables se definen usando la palabra reservada VAR y sólo pueden almacenar valores numéricos incluido enteros y decimales.

La asignación se hace mediante el operador =

VAR a
a = 25 * 1.1

Definición de comentarios

Los comentarios de texto son muy importantes en cualquier lenguaje por lo que podremos definir comentarios en cualquier parte de nuestro archivo, estos comentarios serán de una línea y empezaran con el símbolo #, el comentario puede tener cualquier tipo de texto. ejemplo:

esto es un comentario!! :)

Definición de instrucciones de flujo

- `INICIO`: Marca el inicio del diagrama.
- `FIN`: Marca el final.
- `SI (condición) ENTONCES`: Representa a una condicional
- `MIENTRAS (condición) HACER`: Representa a un ciclo.
- `MOSTRAR "texto"`: Muestra un mensaje en consola.
- `LEER variable`: Representa la lectura de un valor para una variable.

Importante: No se pueden anidar ciclos ni condiciones.

Bloque de código

Un bloque de código es un conjunto de una o mas instrucciones consecutivas. Las instrucciones son: VAR, asignación, MOSTRAR o LEER

Definición de operadores relacionales

Símbolo	**Descripción**
==	igualdad
!=	diferente
>	mayor que
<	menor que
>=	mayor o igual que
<=	menor o igual que

Definición de operadores lógicos

Símbolo	**Descripción**
&&	And
||	Or
!	Not

Ejemplo de un algoritmo en pseudocodigo

```
INICIO
    VAR a = 10
    VAR b = 20
    SI (a < b) ENTONCES
        MOSTRAR "a es menor que b"
    FIN SI
    MIENTRAS (a < 15) HACER
        a = a + 1
        MOSTRAR a
    FIN MIENTRAS
    MOSTRAR "Fin del programa"
FIN
```

Definición del lenguaje de configuración de diagrama de flujo

El propósito principal de este apartado es definir formas, colores, tamaño de letra, entre otros, para cada figura que forma parte del diagrama de flujo.

Esta sección debe tener al menos una instrucción.

Definición de instrucciones de configuración

Para cada instrucción siguiente se puede especificar un índice, el cual indica el elemento que se verá afectado. si el índice no corresponde con algún objeto entonces se ignora. El indice inicia en 1.

Instrucción DEFAULT

%DEFAULT=<indice>

Se aplican los estilos y formatos por defecto para el objeto indicado en el indice. Queda a criterio del desarrollador definir los estilos y formatos por defecto.

Instrucción COLOR_TEXTO_SI

%COLOR_TEXTO_SI=<rgb> o <hexadecimal>|<indice>

Aplica el color al texto de la figura condicional

Instrucción COLOR_SI

%COLOR_SI=<rgb> o <hexadecimal>|<indice>

Aplica el color al fondo de la figura condicional

Instrucción FIGURA_SI

%FIGURA_SI=<NOMBRE_FIGURA>|<indice>

Aplica la figura seleccionada a la figura que representa al condicional

Instrucción LETRA_SI

%LETRA_SI=<NOMBRE_LETRA>|<indice>

Aplica la letra seleccionada al texto de la figura condicional

Instrucción LETRA_SIZE_SI

%LETRA_SIZE_SI=<expresion decimal>|<indice>

Aplica el tamaño de letra seleccionada al texto de la figura condicional

Instrucción COLOR_TEXTO_MIENTRAS

%COLOR_TEXTO_MIENTRAS=<rgb> o <hexadecimal>|<indice>

Aplica el color al texto de la figura condicional del elemento MIENTRAS

Instrucción COLOR_MIENTRAS

%COLOR_MIENTRAS=<rgb> o <hexadecimal>|<indice>

Aplica el color al fondo de la figura condicional del elemento MIENTRAS

Instrucción FIGURA_MIENTRAS

%FIGURA_MIENTRAS=<NOMBRE_FIGURA>|<indice>

Aplica la figura seleccionada a la figura que representa al condicional del elemento MIENTRAS

Instrucción LETRA_MIENTRAS

%LETRA_MIENTRAS=<NOMBRE_LETRA>|<indice>

Aplica la letra seleccionada al texto de la figura condicional del elemento MIENTRAS

Instrucción LETRA_SIZE_MIENTRAS

%LETRA_SIZE_MIENTRAS=<expresion decimal>|<indice>

Aplica el tamaño de letra seleccionada al texto de la figura condicional del elemento MIENTRAS

Instrucción COLOR_TEXTO_BLOQUE

%COLOR_TEXTO_BLOQUE=<rgb> o <hexadecimal>|<indice>

Aplica el color al texto de la figura BLOQUE

Instrucción COLOR_BLOQUE

%COLOR_BLOQUE=<rgb> o <hexadecimal>|<indice>

Aplica el color al fondo de la figura BLOQUE

Instrucción FIGURA_BLOQUE

%FIGURA_BLOQUE=<NOMBRE_FIGURA>|<indice>

Aplica la figura seleccionada a la figura que representa a un BLOQUE

Instrucción LETRA_BLOQUE

%LETRA_BLOQUE=<NOMBRE_LETRA>|<indice>

Aplica la letra seleccionada al texto de la figura BLOQUE

Instrucción LETRA_SIZE_BLOQUE

%LETRA_SIZE_BLOQUE=<expresion decimal>|<indice>

Expresión de colores

Los colores pueden ser representados por medio de rgb o hexadecimal.
Un color rgb está representado por tres expresiones enteras separadas por coma.
Ejemplos:

12,74,25+6-8
251,12,85/2

Un color hexadecimal está representado por H seguido de 6 dígitos hexadecimales.
Ejemplos:
H124AA44
HFF0012A

Expresiones numéricas

Una expresión numérica no es más que una literal entera o decimal, u operaciones aritméticas entre literales enteras y decimales.
Ejemplos:
85
12 - 85
25 - 2 * 5 / 4
8.25 - 25 * 2.2

Si en alguna instrucción se requiere una expresión entera y la expresión genera un valor decimal, entonces la instrucción sólo usará el valor entero y truncará la parte decimal.

Símbolo	**Descripción**	**Precedencia (de menor a mayor)**
+	Suma	1
-	Resta	1
*	Multiplicación	2
/	División	2
( )	Paréntesis	3

Figuras

Las figuras disponibles son:
ELIPSE
CIRCULO
PARALELOGRAMO
RECTANGULO
ROMBO
RECTANGULO_REDONDEADO

Tipos de letras

Las letras disponibles son:
ARIAL
TIMES_NEW_ROMAN
COMIC_SANS
VERDANA

Reportes

Después de la compilación de las instrucciones, los siguientes reportes deben estar disponibles para su visualización en forma tabular:

Reporte de ocurrencias de operadores matemáticos

Operador	Línea	Columna	Ocurrencia
Suma	1	15	12 + 2
Resta	1	28	) - 25
División	2	35	8 / 4
Multiplicación	2	50	5 * 44

Reporte de estructuras de control

Objeto	Linea	condicion
SI	5	a > b
MIENTRAS	24	b != 5

Reporte de errores

Lexema	Línea	Columna	Tipo	Descripción
$	2	13	Léxico	Símbolo no existe en el lenguaje
*	3	1	Sintáctico	Se esperaba ’numero’

Importante: Si existen errores al momento de la compilación entonces solo el reporte de errores debe ser accesible, y si no existen errores de compilación entonces el resto de los reportes deben ser accesibles.

Solo se debe mostrar el diagrama de flujo si no hay errores.

Ejemplo de archivo de entrada

```
INICIO
    VAR a = 10
    VAR b = 20
    SI (a < b) ENTONCES
        MOSTRAR "a es menor que b"
    FIN SI
    MIENTRAS (a < 15) HACER
        a = a + 1
        MOSTRAR a
    FIN MIENTRAS
    MOSTRAR "Fin del programa"
FIN
%%%%
%DEFAULT=1
%COLOR_TEXTO_SI=12,45-5,1|1
%FIGURA_MIENTRAS=CIRCULO|1
%DEFAULT=3
```

CENTRO UNIVERSITARIO DE OCCIDENTE
DIVISIÓN DE CIENCIAS DE LA INGENIERÍA
ORGANIZACIÓN DE LENGUAJES Y COMPILADORES 1
PRIMER SEMESTRE 2026
PRACTICA 1

Importante

- La práctica debe ser desarrollada para plataforma Android usando lenguaje de programación Kotlin
- **Usar herramientas Jflex y Cup para cualquier tipo de análisis/proceso léxico y sintáctico.**
- Práctica obligatoria para tener derecho al siguiente proyecto.
- Las copias y uso de IA obtendrán nota de cero y se notificará a coordinación.

Entrega

- La fecha de entrega es el día 26 de febrero a las 14:00. Los componentes a entregar utilizando un repositorio git son:
- Código fuente
- Archivo apk
- Manual técnico: Detalle de la organización de su proyecto, análisis de gramática para analizador léxico y gramática para analizador sintáctico, diagrama de clases.
- Manual de usuario.

Calificación

Pendiente de definir.