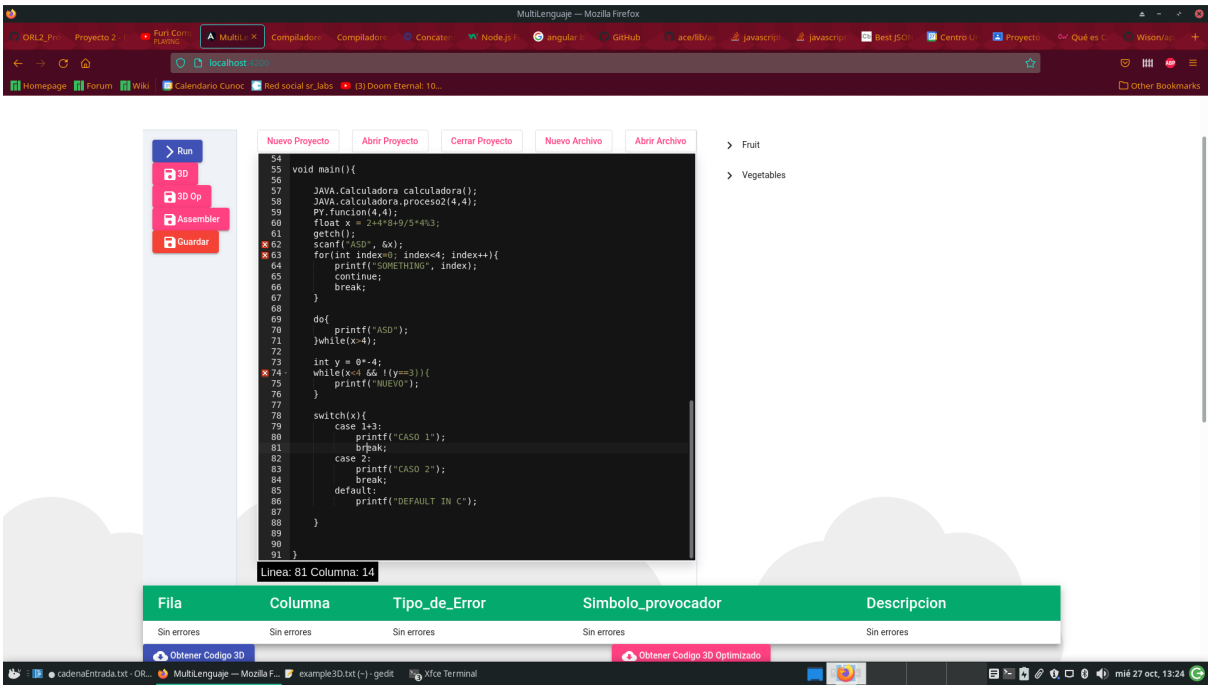
MANUAL DE USUARIO

El multilenguaje es una aplicación dónde puedes trabajar tres lenguajes distintos (JAVA; PYTHON, C), dónde los primeros dos manejan únicamente cierta serie de reglas, como las estructuras básicas de programación, cómo el uso de funciones, en el caso de JAVA se puede heredar clases de otras, siempre y cuando la clase a la que herede esté encima o en posición más arriba que la clase que la quiere extender.

Por lo tanto, se producirá el código 3 direcciones del programa para ver el resultado. El lenguaje JAVA y el lenguaje Python manejan únicamente:

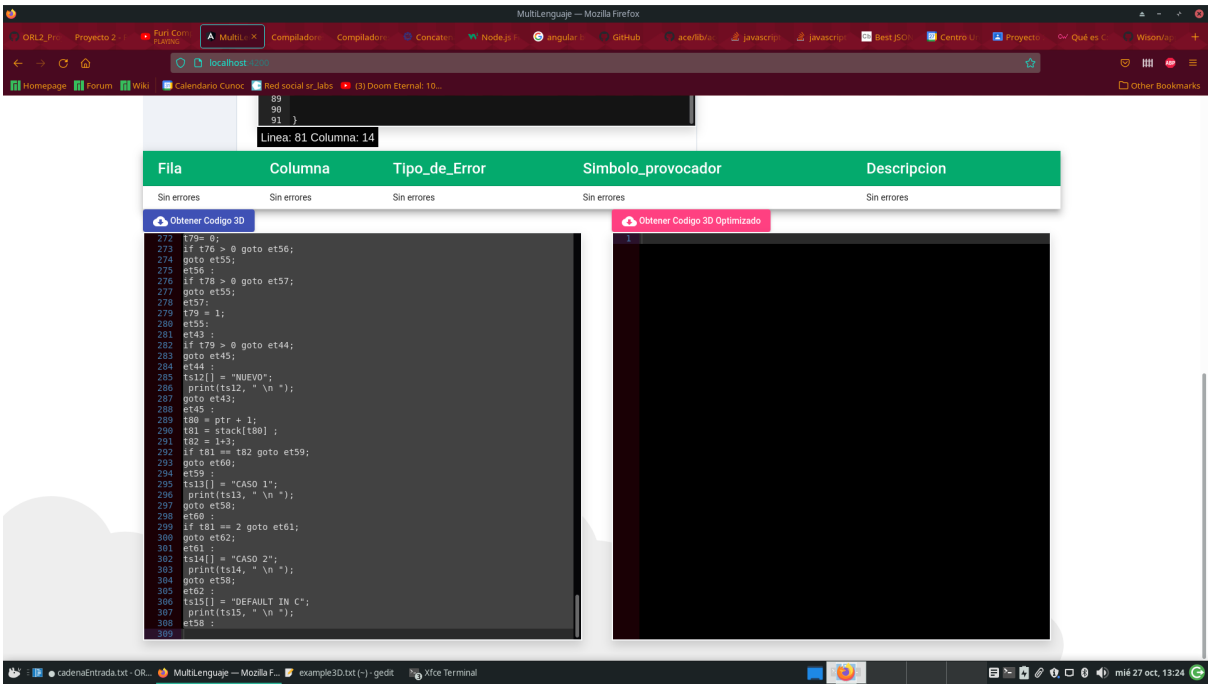
• mensajes a pantalla: se usará la palabra print y println tanto para python como en
java, y la sintaxis es print(valores_separados_por_coma)
● solicitud de datos a usuarios: intinput, floatinput, charinput, ej. vardestino = intinput()
● ciclos (for i, while, do while), incluyendo continue y break.
● En python se usará el ciclo for con range: for x in range(int_exp)
● manejo de condiciones (if, switch)
● declaración de variables
● asignaciones variables
● Funciones
● Procedimientos
● Clases (Java)
● El paso de parametros cumple con Pass-by-object-reference
● La cadena es considerada como un tipo primitivo.
● Por simplicidad los metodos Python no pueden invocar a otros metodos python.
● Por simplicidad los metodos en clases de Java solo pueden invocar a metodos de la misma clase y de sus clases padre.

APLICACIÓN



Esa es el área de texto, tras escribir el código, seleccionaremos run para verificarlo semánticamente, si existieran errores la tabla nos avisa que tipo de errores son, como su posible solución (no todos los casos).

Tras correr el código y no existieran errores, en el apartado de abajo encontraremos dos secciones dónde podemos descargar el código 3 direcciones generado.



Especificaciones del programa principal

Solo el programa principal es capaz de invocar funciones y clases de python y java,

Tipos de datos

Los tipos de datos que se utilizarán son:

- int
- char
- float

NOTA: La variable puede o no ser iniciada al momento de definirla.

Arreglos

Arreglos de N dimensiones de cualquier tamaño. Los arreglos pueden ser de cualquier tipo.

```
Tipo arreglo[dim1];
Tipo arreglo[dim1][dim2][dim3];
```

NOTA: Al momento de declarar un arreglo, las dimensiones pueden ser expresadas usando
literales enteras, constantes o expresiones aritméticas sobre literales y constantes.
Al momento de usar un arreglo, dentro de las dimensiones podrán venir expresiones
aritméticas, arreglos, funciones, constantes, etc. Ejemplo:

```
int arreglo[25+4][CONSTANTE_ENTERA];
x = vector[5*4/7+8*9(4+2)][matriz[1][2*7]];
```

Constantes

Definición de constantes:

```
const tipo nombre_constante = valor;
const float pi = 3.14159;
const char c = 'X'; // X es una constante tipo char
const int X = 10; // X es un tipo int
```

Operadores

Todos los operadores aritméticos, números enteros y reales ambos pueden ser negativos,
los operadores aritméticos pueden ser: +, -, *, /, % (modulo ó residuo), = asignación y
paréntesis.
Ejemplo

```
var = ( 1 + 2 – 3 * 4 / 5 ) % (5 * -3);
```

Comentarios

Los comentarios de línea o de bloque que se hagan deberán de ser pasados a código tres
direcciones y al código ensamblador.

```
//Comentario de línea
/*Comentario
```

de bloque*/

Sentencia if-else

En la instrucción If, dentro de la condición pueden venir expresiones aritméticas y
relacionales (<, >, = =, !=), esto implica operadores && (and), || (or), !(not); con o sin Else.

```
if ( condición )
{
SENTENCIAS;
}
else
{
SENTENCIAS;
}
```

Donde condición puede ser cualquier condición que involucre operadores aritméticos,
relacionales y lógicos, y cualquier tipo de variable. Si el valor resultante de la condición es
mayor que 0, la condición es VERDADERA, si el resultado de la condición es menor o igual
a 0 la condición es FALSA. Ejemplo:

```
(VarAr[1] + 1 > var2 * 2 && var3 != var4 )
```

Sentencia switch

La instrucción Switch, con ncase y con o sin Default.

```
switch (variable)
{
case valor1:
SENTENCIAS;
break;
case valor2:
SENTENCIAS;
break;
.... ....
case valorn:
SENTENCIAS;
break;
default:
SENTENCIAS;
break;
}
```

Ciclo for

for (variable = valor_inicial; condición; incremento|disminucion/accion posterior)
{
SENTENCIAS;
}
Valor_inicial será un número entero o una variable tipo entero, y accionposterior es la
accion que se ejecuta al finalizar una iteracion, condición puede ser cualquier condición,
refiérase a la parte condición de la instrucción if.
La variable de asignación puede ser declarada antes o inicializada directamente en la
misma asignación.

Ciclo while

while (condición)
{
SENTENCIAS;
}
Condición puede ser cualquier condición, refiérase a la parte condición de la instrucción if

Ciclo do while

do
{
SENTENCIAS;
}
while (condición);
Condición puede ser cualquier condición, refiérase a la parte condición de la instrucción if.

Instrucción continue y break

Los ciclos pueden incluir la palabra reservada continue cuya función es obviar las
instrucciones siguientes e iniciar la siguiente iteración.
La palabra reservada break es opcional e indica que se finaliza cualquier ciclo o sentencia
de control sin ejecutar el código que se encuentre por debajo de esta palabra.
Las instrucciones continue y break dependen del contexto por lo que están sujetas a
errores semánticos.

Llamadas a funciones y procedimientos

Si en dado caso la funcion o el procedimiento llevaran parametros, estos pueden ser:

Por valor: tipo_var1 nombre_var1, tipo_var2 nombre_var2, ... , tipo_varn nombre_varn
Por referencia: tipo_var1 &nombre_var1, tipo_var2 &nombre_var2, ... , tipo_varn
&nombre_varn
Invocación (activación):
Variable = PY.nombre_funcion(parámetro);
PY.nombre_procedimiento(parámetros);
Si la firma de una funcion/procedimiento invocado coincide con varias
funciones/procedimientos, entonces se activa la que se encuentra mas abajo en la seccion
de declaraciones de librerias.

scanf

Esta instrucción permitirá asignar valores a las variables.
scanf(" mascara", &variable);
Mascarase refiere a texto y al indicador de que tipo de dato leerá, ejemplo:CENTRO UNIVERSITARIO DE OCCIDENTE
DIVISIÓN DE CIENCIAS DE LA INGENIERÍA
ORGANIZACIÓN DE LENGUAJES Y COMPILADORES 2
PROYECTO 2
scanf("Valor %d", &var); //Leerá del teclado un valor para asignar a variable tipo int
scanf("Valor %c", &var); //Leerá del teclado un valor para asignar a variable tipo char
scanf("Valor %f", &var); //Leerá del teclado un valor para asignar a variable tipo float

printf

Esta instrucción permitirá desplegar mensajes y los valores de las variables
printf(" texto "); ó printf("El valor es mascara", var);
En la primera instrucción solo se mostrará texto, en la segunda se mostrará el texto y el
valor que contenga la variable var, puede haber más de una variable por mostrar.
Mascarase refiere a texto y al indicador de que tipo de dato desplegará, ejemplo:
printf("Valor %d", var);//Desplegará el valor de una variable tipo int
printf ("Valor %c", var); //Desplegará el valor de una variable tipo char
printf ("Valor %f", var);//Desplegará el valor de una variable tipo float

clrscr

limpiar la pantalla, ejemplo:
clrscr();

getch
Esta instrucción leerá una tecla del teclado para poder seguir la ejecución del programa. El
valor que regresa puede o no ser asignado a una variable tipo char o int, si es asignado a
una variable tipo char la variable tendrá el carácter que se presionó; si es asignado a una
variable tipo int la variable tendrá el valor ASCII del carácter que se presionó.
Clases
Todas las clases serán públicas, por lo que deben incluir el modificador de visibilidad
'public', sin embargo, los miembros pueden incluir alguno de los modificadores 'public' y
'private'.
Las clases deben soportar herencia simple.
El uso de las clases se maneja de la siguiente manera
Declaración
JAVA.Nombre_Clase Nombre_Var1, Nombre_Var2;
JAVA.Nombre_Clase Nombre_Var1(parámetros); // llamando a constructor
Llamada a métodos
Variable = JAVA.nombre_objeto.método(parámetros);CENTRO

JAVA.nombre_objeto.método(parámetros);
Si el nombre de una clase coincide con varias clases, entonces se usa la que se encuentra
mas abajo en la sección de declaraciones de librería