

MANUAL TÉCNICO

AUGUS

TABLE OF CONTENTS

Índice

| | |
|---|---|
| Introducción_____ | 1 |
| Objetivos del sistema AUGUS IDE_____ | 2 |
| Contenido técnico_____ | 3 |
| Definiciones dirigidas por la sintaxis_____ | 8 |

Introducción

De la necesidad de un IDE que permitiera manejar un entorno de programación en el cual se pudiera manejar un lenguaje de mediano nivel nació AUGUS IDE, un ambiente de desarrollo capaz de interpretar el lenguaje AUGUS el cual esta basado en PHP y MIDES, el cual permite realizar acciones básicas, tales como operaciones aritméticas, lógicas y relacionales, manejo de arreglos, pilas, structs y etiquetas.

OBJETIVOS DEL SISTEMA AUGUS IDE

Objetivos del sistema AUGUS IDE

- Interpretar el lenguaje de mediano nivel AUGUS
- Manejar de manera centralizada el código que se desarrolle dentro de esta plataforma
- Proveer al usuario un ambiente de fácil usabilidad en cuanto a la interfaz grafica

CONTENIDO TÉCNICO

Contenido técnico

ESPECIFICACIONES DE DESARROLLO

El proyecto AUGUS IDE contiene un licenciamiento tipo GNU General Public License v3.0, el cual permite utilizar el código fuente de manera comercial, modificación, distribución, uso de patentes y de uso privado.

Para tener una descripción mas detallada de lo que comprende esta licencia, visitar:

<https://github.com/obatres/InterpretePython/blob/master/LICENSE>

el código fuente puede ser encontrado en el siguiente link:

<https://github.com/obatres/InterpretePython>

HERRAMIENTAS

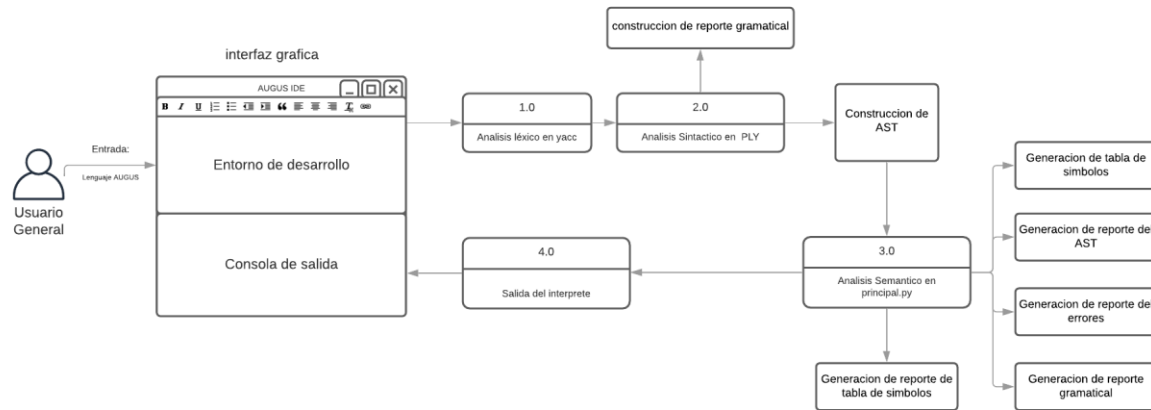
- PLX: Generador de analizadores léxicos y sintácticos.
- Python 3.7: Es un lenguaje de programación
- Windows 8.1: Sistema operativo
- Visual Studio Code: Editor
- PyQt5: como gestor de interfaz grafica

LIBRERIAS

| Librería | Versión |
|-------------------|------------|
| Astroid | 2.4.1 |
| Click | 7.1.2 |
| Colorama | 0.4.3 |
| Graphviz | 0.14 |
| Isort | 4.3.21 |
| Lazy-object-proxy | 1.4.3 |
| Mccabe | 0.6.1 |
| Pylint | 2.5.2 |
| PyQt5 | 5.13.0 |
| PyQt5-sip | 12.8.0 |
| Pyqt5-tools | 5.13.0.1.5 |
| Six | 1.14.0 |
| Toml | 0.10.1 |
| Typed-ast | 1.4.1 |
| Wrapt | 1.12.1 |

CONTENIDO TÉCNICO

FLUJO DE TRABAJO



LENGUAJE

Registros

| | |
|-------------------|---|
| \$t0..\$tn | Temporales |
| \$a0..\$an | Parámetros |
| \$v0..\$vn | Valores devueltos por funciones |
| \$ra | Simulador de dirección de retorno por nivel |
| \$s0..\$sn | Pilas |
| \$sp | Puntero de la pila |

Instrucciones

| | |
|--|--|
| main: | Inicio del programa. |
| label: | Definición del inicio de una etiqueta. |
| goto label; | Salto incondicional hacia una etiqueta. |
| \$t1 = 10; | Asignación numérica. |
| \$t1 = 'hola'; \$t1 = "hola"; | Asignación de una cadena de caracteres. |
| \$t1 = \$t2; | Copia simple. |
| \$t1 = - \$t2; | Negativo. |
| \$t1 = &\$t2; | \$t1 es un puntero a la dirección de \$t2. |
| unset(\$t1); | Destruye la variable \$t1. |
| print(\$t1); | Imprime en pantalla el contenido de \$t1. |
| \$t1 = read(); | Lee la entrada del teclado queda en \$t1. |
| #comment | Comentario de una sola línea. |

CONTENIDO TÉCNICO

| | |
|--------------|------------------------|
| exit; | Finaliza la ejecución. |
|--------------|------------------------|

Conversiones

| | |
|-----------------------------|---|
| \$t1 = (int) \$t2; | Si \$t2 tiene decimales son eliminados. Si \$t2 es un carácter se toma su código ASCII. Si \$t2 es una cadena se toma el ASCII del primer carácter. Si \$t2 es un arreglo se aplica al primer elemento las reglas anteriores. |
| \$t1 = (float) \$t2; | Si \$t2 es entero se agrega ".0". Si \$t2 es un carácter se toma su código ASCII con decimal. Si \$t2 es una cadena se toma el ASCII del primer carácter más el decimal. Si \$t2 es un arreglo se aplica al primer elemento las reglas anteriores. |
| \$t1 = (char) \$t2; | Si \$t2 es un número de 0 a 255 se convierte en el carácter basado en ASCII. Si \$t2 es un número mayor a 255 entonces se aplica el módulo 256 para extraer el ASCII. Si \$t2 es un decimal, se quitan los decimales y se aplican las reglas anteriores. Si \$t2 es una cadena, se almacena solo el primer carácter. Si \$t2 es un arreglo, se almacena solo el primer valor aplicando las reglas anteriores. |

Operaciones

| | |
|-------------------------------------|--|
| \$t1 = \$t2 + \$t3; | Suma |
| \$t1 = \$t2 - \$t3; | Resta. |
| \$t1 = \$t2 * \$t3; | Multiplicación. |
| \$t1 = \$t2 / \$t3; | División. |
| \$t1 = \$t2 % \$t3 | Residuo. |
| \$t1 = abs(\$t2); | Valor absoluto. |
| \$t1 = !\$t2; | Not, si \$t2 es 0 \$t1 es 1, si \$t2 es 1 \$t1 es 0. |
| \$t1 = \$t2 && \$t3; | And, 1 para verdadero, 0 para falso. |
| \$t1 = \$t2 \$t3; | Or, 1 para verdadero, 0 para falso. |
| \$t1 = \$t2 xor \$t3; | Xor, 1 para verdadero, 0 para falso. |
| \$t1 = ~\$t2; | Not. |
| \$t1 = \$t2 & \$t3; | And. |
| \$t1 = \$t2 \$t3; | Or. |
| \$t1 = \$t2 ^ \$t3; | Xor. |

CONTENIDO TÉCNICO

| | |
|--|--|
| <code>\$t1 = \$t2 << \$t3;</code> | Shift de \$t2, \$t3 pasos a la izquierda. |
| <code>\$t1 = \$t2 >> \$t3;</code> | Shift de \$t2, \$t3 pasos a la derecha. |
| <code>\$t1 = \$t2 == \$t3;</code> | \$t1 = 1 si \$t2 es igual a \$t3, sino 0. |
| <code>\$t1 = \$t2 != \$t3;</code> | \$t1 = 1 si \$t2 no es igual a \$t3, sino 0. |
| <code>\$t1 = \$t2 >= \$t3;</code> | \$t1 = 1 si \$t2 es mayor o igual a \$t3, sino 0. |
| <code>\$t1 = \$t2 <= \$t3;</code> | \$t1 = 1 si \$t2 es menor o igual a \$t3, sino 0. |
| <code>\$t1 = \$t2 > \$t3;</code> | \$t1 = 1 si \$t2 es mayor a \$t3, sino 0. |
| <code>\$t1 = \$t2 < \$t3;</code> | \$t1 = 1 si \$t2 es menor a \$t3, sino 0. |
| <code>\$t1 = array();</code> | Define \$t1 como un arreglo o un struct, para diferenciarlos se utiliza ya sea el valor numérico o el nombre asociativo. |
| <code>\$t1[4] = 1;</code> | Asignación de un valor numérico (1) a un índice del arreglo (4). |
| <code>\$t1['nombre'] = 'carlos';</code> <code>\$t1["nombre"] = "carlos";</code> | Asignación de un valor cadena (carlos) a un componente del struct (nombre). |
| <code>\$t1 = \$t1[4];</code> | Acceso a un índice del arreglo. |
| <code>\$t1 = \$t2['nombre'];</code> | Acceso a un componente del struct. |
| <code>\$t1 = 'hola';</code> <code>print(\$t1[0]); #imprime h</code> | Acceder a un carácter de una cadena. |

Ejemplo de un archivo de entrada

CONTENIDO TÉCNICO

```
1  main:
2      $s0 = array(); #stack
3      $sp = -1; #null pointer
4      $a0 = 3; #m
5      $a1 = 3; #n
6      $sp = $sp + 1;
7      $s0[$sp] = $a0; #push
8      goto ack;
9  ret0:
10     print($v0);
11     exit;
12  ack:
13     if ($sp<0) goto ret3; #empty stack
14     $a0 = $s0[$sp]; #pop
15     $sp = $sp - 1;
16     if ($a0!=0) goto ret1;
17     $t4 = $a0 + 1;
18     $a1 = $a1 + $t4;
19     goto ack;
20  ret1:
21     if ($a1!=0) goto ret2;
22     $a1 = $a1 + 1;
23     $a0 = $a0 - 1; # --m
24     $sp = $sp + 1;
25     $s0[$sp] = $a0; #push
26     goto ack;
27  ret2:
28     $a0 = $a0 - 1; # --m
29     $sp = $sp + 1;
30     $s0[$sp] = $a0;
31     $a0 = $a0 + 1; # ++m
32     $sp = $sp + 1;
33     $s0[$sp] = $a0; #push
34     $a1 = $a1 - 1;
35     goto ack;
36  ret3:
37     $v0 = $a1;
38     goto ret0;
```

DEFINICIONES DIRIGIDAS POR LA SINTAXIS

Definiciones dirigidas por la sintaxis

GRAMATICA ASCENDENTE

| PRODUCCIONES | |
|-------------------------|---|
| Init | Instrucciones |
| Instrucciones | Instrucciones – instrucción |
| | Instrucción |
| Instrucción | Imprimir_instr |
| | definicion_instr |
| | asignacion_instr |
| | mientras_instr |
| | if_instr |
| | INICIO |
| | UNSETF |
| | EXITF |
| | ASIGNAARREGLO |
| | INICIAPILA |
| | ASIGNAPUNTERO |
| | ASIGNAPILA |
| | ASIGNACIONEXTRA |
| | DEFINEL |
| | DEFINEGOTO |
| DEFINEL | ID DOSP |
| DEFINEGOTO | GOTO ID PTCOMA |
| ASIGNACIONEXTRA | VALORESPARAM IGUAL expresion_log_relacional PTCOMA |
| VALORESPARAM | PARAMETRO |
| | VALORDEVUELTO |
| | DIRRETORNO |
| ASIGNAPILA | PILAPOS CORIZQ PILAPUNTERO CORDER IGUAL expresion_log_relacional PTCOMA |
| ASIGNAPUNTERO | PILAPUNTERO IGUAL expresion_log_relacional PTCOMA |
| INICIAPILA | PILAPOS IGUAL ARRAY PARIZQ PARDER PTCOMA |
| ASIGNAARREGLO | TEMPORAL ACCESO IGUAL expresion_log_relacional PTCOMA |
| UNSETF | UNSET PARIZQ expresion_numerica PARDER PTCOMA |
| INICIO | MAIN DOSP |
| EXITF | EXIT PTCOMA |
| imprimir_instr | PRINT PARIZQ expresion_log_relacional PARDER PTCOMA |
| definicion_instr | NUMERO TEMPORAL PTCOMA |
| asignacion_instr | TEMPORAL IGUAL expresion_log_relacional PTCOMA |

DEFINICIONES DIRIGIDAS POR LA SINTAXIS

| | |
|---------------------------------|---|
| mientras_instr | MIENTRAS PARIZQ expresion_log_relacional PARDER LLAVIZQ instrucciones LLAVDER |
| if_instr | IF expresion_numerica DEFINEGOTO |
| expresion_numerica | expresion_numerica MAS expresion_numerica |
| | expresion_numerica MENOS expresion_numerica |
| | expresion_numerica POR expresion_numerica |
| | expresion_numerica DIVIDIDO expresion_numerica |
| | expresion_numerica RES expresion_numerica |
| | expresion_numerica ANDBIT expresion_numerica |
| | expresion_numerica ORBIT expresion_numerica |
| | expresion_numerica XORBIT expresion_numerica |
| | expresion_numerica IZQBIT expresion_numerica |
| | expresion_numerica DERBIT expresion_numerica |
| | NOTBIT expresion_numerica |
| | MENOS expresion_numerica %prec UMENOS |
| | PARIZQ expresion_log_relacional PARDER |
| | ENTERO |
| | DECIMAL |
| | ID |
| | PILAPOS |
| | PILAPUNTERO |
| | PILAPOS CORIZQ PILAPUNTERO CORDER |
| | PARAMETRO |
| | VALORDEVUELTO |
| | DIRRETORNO |
| | TEMPORAL |
| | PTEMPORAL |
| | CADENA |
| | CADE |
| | READ PARIZQ PARDER |
| | ARRAY PARIZQ PARDER |
| | ABS PARIZQ expresion_numerica PARDER |
| | TIPOCONVERSION expresion_numerica |
| | TEMPORAL ACCESO |
| ACCESO | ACCESO CORIZQ expresion_numerica CORDER |
| | CORIZQ expresion_numerica CORDER |
| TIPOCONVERSION | PARIZQ INT PARDER |
| | PARIZQ FLOAT PARDER |
| | PARIZQ CHAR PARDER |
| expresion_log_relacional | expresion_numerica MAYQUE expresion_numerica |

DEFINICIONES DIRIGIDAS POR LA SINTAXIS

| | |
|---------------------------------|--|
| | expresion_numerica MENQUE expresion_numerica |
| | expresion_numerica IGUALQUE expresion_numerica |
| | expresion_numerica NIGUALQUE expresion_numerica |
| | expresion_numerica MAYORIG expresion_numerica |
| | expresion_numerica MENORIG expresion_numerica |
| | expresion_log_relacional ANDLOG expresion_log_relacional |
| | expresion_log_relacional ORLOG expresion_log_relacional |
| | expresion_log_relacional XORLOG expresion_log_relacional |
| | NOTLOG expresion_log_relacional |
| expresion_log_relacional | expresion_numerica |

GRAMATICA DESCENDENTE

| PRODUCCIONES | |
|------------------------|-------------------------------------|
| Init | Instrucciones |
| Instrucciones | Instrucción instruccionesP |
| InstruccionesP | Instrucción instruccionesP |
| | EPSILON |
| Instrucción | Imprimir_instr |
| | definicion_instr |
| | asignacion_instr |
| | mientras_instr |
| | if_instr |
| | INICIO |
| | UNSETF |
| | EXITF |
| | ASIGNAARREGLO |
| | INICIAPILA |
| | ASIGNAPUNTERO |
| | ASIGNAPILA |
| | ASIGNACIONEXTRA |
| | DEFINEL |
| | DEFINEGOTO |
| DEFINEL | ID DOSP |
| DEFINEGOTO | GOTO ID PTCOMA |
| ASIGNACIONEXTRA | VALORESPARAM IGUAL expresion PTCOMA |
| VALORESPARAM | PARAMETRO |
| | VALORDEVUELTO |

DEFINICIONES DIRIGIDAS POR LA SINTAXIS

| | |
|---------------------------|--|
| | DIRRETORNO |
| ASIGNAPILA | PILAPOS CORIZQ PILAPUNTERO CORDER IGUAL expresion PTCOMA |
| ASIGNAPUNTERO | PILAPUNTERO IGUAL expresion PTCOMA |
| INICIAPILA | PILAPOS IGUAL ARRAY PARIZQ PARDER PTCOMA |
| ASIGNAARREGLO | TEMPORAL ACCESO IGUAL expresion PTCOMA |
| UNSETF | UNSET PARIZQ expresion PARDER PTCOMA |
| INICIO | MAIN DOSP |
| EXITF | EXIT PTCOMA |
| imprimir_instr | PRINT PARIZQ expresion PARDER PTCOMA |
| definicion_instr | NUMERO TEMPORAL PTCOMA |
| asignacion_instr | TEMPORAL IGUAL expresion PTCOMA |
| mientras_instr | MIENTRAS PARIZQ expresion PARDER LLAVIZQ instrucciones LLAVDER |
| if_instr | IF expresion DEFINEGOTO |
| expresion | expresion_numerica MAS expresion_numerica |
| | expresion_numerica MENOS expresion_numerica |
| | expresion_numerica POR expresion_numerica |
| | expresion_numerica DIVIDIDO expresion_numerica |
| | expresion_numerica RES expresion_numerica |
| | expresion_numerica ANDBIT expresion_numerica |
| | expresion_numerica ORBIT expresion_numerica |
| | expresion_numerica XORBIT expresion_numerica |
| | expresion_numerica IZQBIT expresion_numerica |
| | expresion_numerica DERBIT expresion_numerica |
| expresion_numerica | NOTBIT expresion |
| | |
| | MENOS expresion %prec UMENOS |
| | PARIZQ expresion PARDER |
| | ENTERO |
| | DECIMAL |
| | ID |
| | PILAPOS |
| | PILAPUNTERO |
| | PILAPOS CORIZQ PILAPUNTERO CORDER |
| | PARAMETRO |
| | VALORDEVUELTO |
| | DIRRETORNO |
| | TEMPORAL |
| | PTEMPORAL |
| | CADENA |
| | CADE |

DEFINICIONES DIRIGIDAS POR LA SINTAXIS

| | |
|-----------------------|--|
| | READ PARIZQ PARDER |
| | ARRAY PARIZQ PARDER |
| | ABS PARIZQ expresion PARDER |
| | TIPOCONVERSION expresion |
| expresion | TEMPORAL ACCESO |
| ACCESO | CORIZQ expresion_numerica CORDER ACCESOP |
| ACCESOP | CORIZQ expresion_numerica CORDER ACCESOP |
| | EPSILON |
| TIPOCONVERSION | PARIZQ INT PARDER |
| | PARIZQ FLOAT PARDER |
| | PARIZQ CHAR PARDER |
| expresion | expresion_numerica MAYQUE expresion_numerica |
| | expresion_numerica MENQUE expresion_numerica |
| | expresion_numerica IGUALQUE expresion_numerica |
| | expresion_numerica NIGUALQUE expresion_numerica |
| | expresion_numerica MAYORIG expresion_numerica |
| | expresion_numerica MENORIG expresion_numerica |
| | expresion_log_relacional ANDLOG expresion_log_relacional |
| | expresion_log_relacional ORLOG expresion_log_relacional |
| | expresion_log_relacional XORLOG expresion_log_relacional |
| | NOTLOG expresion |
| expresion | expresion_numerica |
| EPSILON | [vacio] |