



**Tecnológico
de Monterrey**

EaStat

Manual de usuario

Antonio Torres Carvajal – A01561769

6 de junio del 2022

Requerimientos del sistema	3
Ejecución	3
Sintaxis del lenguaje	5
Programa Base	5
Declaración de variables GLOBALES	5
Definición de funciones y variables locales	6
Condicional	6
Ciclo mientras	7
Arreglos y matrices	7

Requerimientos del sistema

1. Sistema operativo que pueda ejecutar python.
2. Contar con python instalado (versión 3.10.4. en adelante).

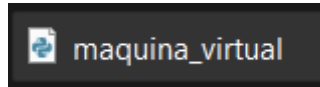
Ejecución

Para instalar el lenguaje solo es necesario descomprimir la carpeta descargada y seguir el siguiente directorio:

- "Actual\EaStat_Compiler\EaStat_Compiler"

Seguir los siguientes pasos:

1. Ejecutar el archivo "maquina_virtual.py"



2. El programa mostrará al usuario una serie de archivos de prueba ejecutables en el lenguaje EaStat.

```
Selecciona el test a ejecutar:
1. testA_Tokens.est
2. testB_SemEst1.est
3. testC_SemEst2.est
4. testD_DeclaracionDoble.est
5. testE_Indeterminado.est
6. testF.est
7. testG_VarNoDeclarada.est
8. testH_CuadsLineales.est
9. testI_CuadsCondicionales.est
10. testJ_CuadsCiclicos.est
11. testK_ExcesoENT.est
12. testL_ExcesoFLOT.est
13. testM_Constantes.est
14. testN_Funciones1.est
15. testO_Funciones2.est
16. testP_EjecExpsAritméticas.est
17. testQ_EjecEntradaSalida.est
18. testR_EjecCiclosCondiciones.est
19. testS_EjecFuncionVacio.est
20. testT_EjecFuncionENT.est
21. testU_EjecFuncionRecursiva.est
22. testV_EjecFactorialCiclo.est
23. testW_EjecFactorialRecursivo.est
24. testX_EjecFiboCiclo.est
25. testY_EjecFiboRecursivo.est

Test número:
```

3. El usuario deberá de seleccionar un archivo de prueba escribiendo el número del archivo que aparece a la izquierda de cada nombre.

```
20. testT_EjecFuncionENT.est
21. testU_EjecFuncionRecursiva.est
22. testV_EjecFactorialCiclo.est
23. testW_EjecFactorialRecursivo.est
24. testX_EjecFiboCiclo.est
25. testY_EjecFiboRecursivo.est

Test número: 25_
```

4. Al presionar “ENTER”, el archivo se compilará y se ejecutará.

```
Test número: 25
Corriendo "testY_EjecFiboRecursivo.est"

Compilando programa...

¡Programa compilado con éxito!

Importando código objeto...

¡Cuádruplos importados!

importando estructuras de ejecución...

¡Directorio importado!

Importando constantes identificadas

¡Constantes importadas!

Generando memoria...

¡Memoria generada!

Iniciando ejecución...
```

5. Al terminar, el usuario solo deberá presionar enter y la ventana se cerrará.

```
Iniciando ejecución...

|-----|

Ingresa el número de términos:
5
El resultado es:
3

|-----|

¡Programa ejecutado con éxito!
```

6. Si el usuario desea ejecutar su propio programa, debe de generar un archivo de texto y guardarlo con extensión .est en el siguiente directorio
- “\Actual\EaStat_Compiler\EaStat_Compiler\testing”
- Al volver a ejecutar, el usuario podrá ver su archivo en la consola.

Sintaxis del lenguaje

Programa Base

Para poder ejecutar el código sin problema, debe de seguir esta sintaxis inicial en la que partimos de la palabra reservada “inicio” seguido de la palabra “principal” y un cuerpo “{ },”

```
inicio

principal{

};
```

Declaración de variables GLOBALES

Cuando buscamos declarar variables globales, lo único que tendremos que realizar es agregar el siguiente formato entre la palabra reservada “inicio” y “principal”

```
<tipo> nombreDeVar;
<tipo> nombreDeVar2;
<tipo> nombreDeVar3;
```

Se cuentan con 3 tipos de variables:

- ent: variables enteras
- flot: variables flotantes
- car: caracteres

Por ende, las declaraciones quedarían de la siguiente manera

```
ent nombreDeVar;
flot nombreDeVar2;
car nombreDeVar3;
```

Estas variables se pueden usar en cualquier cuerpo ({ ... }); en el programa para expresiones aritméticas, argumentos, condicionales, escritura, etc.

```
inicio
ent nombreDeVar1;
ent nombreDeVar2;

principal{
    escrib(nombreDeVar1 + nombreDeVar2);
};
```

Definición de funciones y variables locales

Las funciones son declaradas entre la declaración de variables globales y la palabra reservada “principal”, aunque si no hay variables globales, es posible declarar funciones entre las palabras reservadas “inicio” y “principal”. No es posible iterar entre declaración de variable y definición de función.

```
inicio
ent eVar1;
ent eVar2;

vacio llamadaVerdadera(ent eVarA, ent eVarB){
    ent eLocal;
    flot fLocal;
    escrib(eVarA);
    escrib(eVarB);
};

principal{
    eVar1 = 4;
    eVar2 = 5;
    llamadaVerdadera(eVar1, eVar2);
};
```

Como se puede observar, también es posible declarar variables de forma local en las funciones, sin embargo sólo es posible que se declaren primero antes de cualquier otro estatuto y ese es el único otro lugar en el que se permite la declaración de variables locales..

Condicional

Si se desea realizar una condición se puede hacer de las siguientes maneras:
Condición de un camino:

```
si(5 > 0 || 7 == 7){
    eVar = 5;
};
```

Condición de dos caminos:

```
si(eVar1 > 0 && 8 < 20){

    eVar1 = eVar1 - 1;

} otro {

    eVar1 = eVar1 + 5;
```

```
};
```

Ciclo mientras

Si se desea realizar una condición se puede hacer de la siguiente forma:

```
mientras(i < 20){  
    i = i + 1;  
};
```

Arreglos y matrices

Para utilizar arreglos y matrices, solo es necesario declarar sus variables con hasta dos dimensiones de la siguiente manera:

```
ent Arr[9];  
ent Arr[9][8];
```

Para tener acceso a cada espacio, solo es necesario llamar a la variable con los mismos corchetes utilizados en la declaración.

```
Arr[9] = 50;  
Arr[9][8] = 68;
```