

**Universidad Tecnológica Centroamericana**

Facultad de Ingeniería

**Proyecto: Definición**

**Docente:**

Ing. Carlos Vallejo

**Asignatura**:

Compiladores I

**Presentado por:**

Carlos Romero 11711209

Richardson Cárcamo 11711075

**Tegucigalpa, M.D.C,**

**25 de agosto de 2020**

# LENGUAJE RILL-20

## Declaración de Variables

Sin tipo en la declaración, ya que los tipos son char (‘’), bool (false/true) e int, son fáciles de distinguir entre ellos.

La declaración será de la forma:

**ID:** valor.,

### Arreglos

La declaración de loa arreglos será de la siguiente manera. Los arreglos podrán ser declarados con elementos dentro de ellos o no, pero deberá ingresarse [] o [[]] para definir que son arreglos.

**ID:** [].,

**ID:** [[]].,

**ID:** [1,2,’c’,false]

**ID:** [[1,2,3],[‘a’,’b’,’c’],[false,true,false]].,

NOTA: Los arreglos serán de tamaño variable, por lo que no requiere una definición de tamaño. Los arreglos admitirán diferentes tipos de datos dentro de un mismo arreglo; además, los arreglos de dos dimensiones admitirán arreglos de múltiples tamaños dentro del arreglo del mismo arreglo.

## Operadores relacionales

Estos serán:

* Igual: =
* Menor: <
* Mayor: >
* Mayor igual: <=
* Menor igual: >=

## Fin de línea

Los fines de línea serán representados por la combinación de caracteres “.,”. Y estos deberán ir en cada sentencia tanto fuera como dentro de los bloques **if**, **wh**, **fr** etc.

## Bloques de Contexto

Para la definición de ámbitos dentro del código fuente se utilizará lo siguiente:

* Abrir: (x:
* Cerrar: :)

Donde x representa el tipo de bloque de contexto que es (**if** para IF y SWITCH, **wh** para WHILE, **fr** para FOR, **fun** para las funciones y procedimientos).

## Bloques de decisión IF-SWITCH

Los bloque de decisión será una combinación de IF con SWITCH, pero dependiendo de la definición éste tendrá un comportamiento distinto:

### Comportamiento IF

(condición)(**if**:

----Sentencia verdadera----

**br**.,

(condición):

----Sentencia-----

**br.,**

(**df**):

----Sentencia falsa-----

**br**.,

:)

De esta manera al declararse el IF con una condición (de la forma **a op\_rel b**), el compilador entenderá que es un bloque con comportamiento IF y las sentencias justo después de la declaración serán las que se ejecuten en caso de ser la condición verdadera.

Además, los bloques dentro del IF con la estructura:

(condición):

----Sentencia----

**br.,**

Funcionarán como ELSEIF del lenguaje. Del mismo modo, El bloque IF puede o no tener bloque ELSE, y éste será representado el bloque con la estructura:

(**df**):

----Sentencia falsa-----

**br**.,

### Comportamiento SWITCH

(variable)(**if**:

(4):

----Sentencia-----

**br.,**

(<4):

----Sentencia-----

**br.,**

(**df**):

----Sentencia falsa-----

**br**.,

:)

Cuando la declaración del bloque if: sea con una sola variable, éste se comportará como un SWITCH. La variable en la definición deberá ser una variable ya declarada en el contexto exterior del bloque SWITCH.

Los bloque que representan los CASE será con la estructura siguiente:

(<4):

----Sentencia-----

**br.,**

Y la declaración del case (<4), puede tener dos estructura:

* (valor) – que evaluará si la variable declarada en la definición de SWITCH es igual al valor en la definición del CASE.
* (op\_rel valor) – que evaluará si la variable declarada en la definición de SWITCH cumple dicha condición.

## Bloques de decisión WHILE-FOR

### WHILE

El bloque de comportamiento WHILE tendrá la siguiente estructura:

(condición)(**wh**:

----Sentencias----

:)

### FOR

El bloque de comportamiento FOR tendrá la siguiente estructura:

(var)(condición, incremento)(**fr**:

:)

Si la variable en la definición no había sido declarada antes, el ámbito de la variable será sólo dentro del bloque FOR, la condición tendrá la estructura (op\_rel valor) y el incremento puede ser (+num) o (-num).

## Funciones y Procedimientos

(parámetros separados por coma) nombre (**fun**:

:)

El compilador analizará automáticamente si ésta retorna o no, basándose en si ésta utiliza el token de retorno “ret“. El llamado a funciones se hará de la siguiente manera.

nombre ().,

## Conversión automática de tipos

c = "hola" -> c = ['h','o','l','q']

a:true

b:5

a+b= 6

## Print

(pp: "texto a imprimir"- variableImprimir :)

## Comentarios

Estos tendrán la forma:

|Éste es un comentario de una línea|

| Comentario que contiene más

de una línea |

## Código de Ejemplo

() **main** (**fun**:

|ESTE ES UN COMENTARIO|

|Ejemplo de if que se comporta como IF|

a: false.,

(a=true)(**if**:

|imprima 0|

(pp: "0":).,

**br**.,

(**df**):

|imprima -1|

(**pp**: "-1" :).,

**br**.,

:)

|Ejemplo de if que se comporta como SWITCH|

b: 'b'.,

(b)(**if**:

('a'):

(**pp**: "a":).,

**br.,**

('b'):

(**pp**: "b":).,

**br.,**

(**df**):

(**pp**: "0":).,

**br**.,

:)

|Ejemplo de wh que se comporta como WHILE|

c: 0.,

(c<10)(**wh**:

c: c+1.,

:)

|Ejemplo de FOR|

d : ['a',0].,

(i)(<2, +1)(**fr**:

(**pp**: "i) " - d[i]:).,

:)

|Ejemplo de función|

e: [[1,0],[0,1]].,

Matriz (e).,

:)

|Ejemplo de función|

(m) Matriz (**fun**:

(i)(<2, +1)(**fr**:

(j)(<2, +1)(**fr**:

(**pp**: m[i][j] :).,

:)

(**pp**: "\n" :).,

:)

:)