**DOCUMENTO TALLER 9**

**MÓDULO DE FORMACIÓN: FUNDAMENTOS DE PROGRAMACIÓN**

**TEMAS APRENDIZAJE: Estructuras lógicas**

**ACTIVIDAD DE ENSEÑANZA – APRENDIZAJE – EVALUACIÓN:** Estructuras Lógicas

**TIEMPO DE LA ACTIVIDAD DE E-A-E: 5** Horas

**TIEMPO DEL TALLER DE APRENDIZAJE:**  5 Horas

**OBJETIVOS**

1. Tipos de estructuras

2. Definiciones de operaciones

**DESARROLLO DE LA CLASE**

ESTRUCTURAS LOGICAS

Son operaciones lógicas, y se utilizan generalmente para validar. Las Estructuras Lógicas se clasifican en dos:

* **SELECTIVAS o Bifurcaciones:** Selección de una o más opciones, es decir, solamente puedo seleccionar una. Las selectivas se dividen en dos:

SI

**NO**

* Si
* Seleccione según el caso.
* **REPETITIVAS:** Repiten varios procesos ciclos o bucles en tanto se controlen. Estas se dividen en tres:
* **MQ** mientras que
* **HQ** hasta que
* Desde/para.

Todas las estructuras lógicas tienen una condición, definiendo condición como una operación lógica una expresión que tiene un valor de verdad (SI) o falsedad (NO) como respuesta, es decir, una relación de condición entrega un valor determinado booleano y su antónimo, que se realiza o se obtiene a través de esa operación lógica entre dos valores sea numérico, alfanumérico, mediante un operador lógico.

Condición es la comparación entre dos valores que están representados datos o variables.

Tenga presente como detalle analógico que las Operaciones matemáticas cuyo propósito es entregar el resultado numérico, esta se produce por la realización que se hace entre sus operandos por medio de un operador aritmético.

Las operaciones lógicas se originan de las operaciones aritméticas a quienes se les llama álgebra de bool o booleana.

Operadores lógicos simples que buscan comparar dos valores con respuesta booleana estos son:

< Menor que

> Mayor que

= Igual a

Diferente de

Mayor o igual a

Menor o igual a

Una operación lógica compuesta es una composición de dos operaciones simples, está compuesta por:

Conjunción

Disyunción

Negación

ESTRUCTURA LÓGICA SI:

**CASO I:** Cuando un algoritmo viene ejecutando un proceso se encuentra con el condicional si, el cual se va a ejecutar si la condición es verdadera. Veamos la estructura del diagrama de flujo y el seudocódigo. Tenga presente que la S es sentencia cualquiera.

No

**Inicio**

S1

S2

S2CV

SNCV

**1**

Si

**FIN**

**1**

**C**

Inicio

S1

S2

.

.

Sn

Si C

S1CV

S2CV

.

.

Sncv

Fin si

Fin Inicio

***Pseudocodigo***

**EJEMPLO:**

**Se digitan 2 valores A y B si, A es mayor que B, imprima un mensaje que diga “a es mayor que B”, almacene en memoria la suma de A y B, e imprima el resultado almacenado en memoria.**

**Desarróllelo. Si no lo puede desarrollar consulte con compañeros. Al final lo socializaremos.**

Inicio

a>b

“a es mayor que b”

Fin

c=a+b

SI

“”c

a

b

NO

1

1

Inicio

Lea a

Lea b

Si (a es mayor que b)

Imprima “a es mayor que b”

Almacenar c=a+b

Imprima c

Fin si

Fin

**CASO II:** Cuando un algoritmo viene ejecutando un proceso se encuentra con el condicional si, el cual se va a ejecutar si la condición es verdadera. Antes de finalizar y después de la estructura se ejecutan varias sentencias.

Inicio

S1

S2

.

.

SN

Si C

S1CV

S2CV

.

.

Sncv

Fin si

SN1

SNN

Fin Inicio

S2CV

SNCV

**1**

SN1

SNN

**FIN**

**1**

**Inicio**

S1

SN

**C**

si

no

EJEMPLO:

**Se digitan 2 valores A y B si, A es mayor que B, imprima un mensaje que diga “A es mayor que B”, almacene en memoria la suma de A y B, e imprima el resultado almacenado en memoria. Antes de finalizar el programa que imprima un mensaje que diga “chaooo”, antes debe enviar otro mensaje que diga “que vuelvas”.**

**Desarróllelo. Si no lo puede desarrollar consulte con compañeros. Al final loi socializaremos.**

Inicio

a>b

“a es mayor que b”

Fin

c=a+b

SI

“”c

a

b

NO

1

1

“que vuelvas”

“chao”

Inicio

Lea a

Lea b

Si (a es mayor que b)

Imprima “a es mayor que b”

Almacenar c=a+b

Imprima c

Fin si

Imprima “que vuelvas”

Imprima “chao”

Fin

**CASO III**

Caso en el que tiene los dos valores de la condición, verdadera y falsa, y al tomar una ejecución finaliza el programa.

**Inicio**

S1

SN

S2CV

SNCV

**1**

**FIN**

**1**

S1CF

SNCF

Si

No

**C**

Inicio

S1

S2

.

.

SN

Si C

S1CV

S2CV

.

.

SnCV

Si Falso

S1CF

S2CF

.

.

SnCF

Fin si

Fin Inicio

EJEMPLO:

**Se digitan 2 valores A y B, si A es mayor que B, imprima un mensaje que diga “a es mayor que B”, almacene en memoria la suma de A y B, e imprima el resultado almacenado en memoria. En caso contrario, imprima un mensaje que diga “B es mayor que A”, almacene en memoria el producto entre A y B, e imprima el resultado almacenado en memoria.**

**Desarróllelo. Si no lo puede desarrollar consulte con compañeros. Al final loi socializaremos.**

Inicio

a>b

“a es mayor que b”

Fin

c=a+b

SI

“”c

a

b

NO

1

1

“b es mayor que a”

c=a+b

“”c

Inicio

Lea a

Lea b

Si (a es mayor que b)

Imprima “a es mayor que b”

Almacenar c=a+b

Imprima c

Fin si

Imprima “a es mayor que b”

Almacenar c=a+b

Imprima c

Fin

**CASO IV**

Caso completo.

Inicio

S1

S2

.

.

SN

Si C

S1CV

S2CV

.

.

SnCV

Si Falso

S1CF

S2CF

.

.

SnCF

Fin si

SN1

SNN

Fin Inicio

S2CV

SNCV

**1**

**Inicio**

S1

SN

SN1

SNN

**FIN**

**1**

S1CF

SNCF

**C**

Si

No

Deduzca el texto y el ejemplo a aplicar en este caso

Inicio

a>b

“a es mayor que b”

Fin

c=a+b

SI

“”c

a

b

NO

1

1

“b es mayor que a”

c=a+b

“”c

“que vuelvas”

“chao”

Inicio

Lea a

Lea b

Si (a es mayor que b)

Imprima “a es mayor que b”

Almacenar c=a+b

Imprima c

Fin si

Imprima “a es mayor que b”

Almacenar c=a+b

Imprima c

Imprima “que vuelvas”

Imprima “chao”

Fin

**CASO V**

Es el caso que dentro de una condición verdadera o falsa, va otra condición.

Esto es el caso de los si anidados