DIAGRAMAS DE FLUJO DE

ESTRUCTURAS ALGORITMICAS CONDICIONALES

**FRASES CONDICIONALES SELECTIVAS**

Estructura (frase) Condicional Selectiva *IF-THEN* ----------------

*condiciónLogi*

V *then*

F

*IF*

# Bfi

------

------

*Bfi* es un bloque o serie de frases o instrucciones algorítmicas.

Sintaxis en lenguaje Java y Ensamblador x86.

|  |  |
| --- | --- |
| Lenguaje Java | Lenguaje Ensamblador x86 |
| **if** ( *condiciónLogi* )  {  *Bfi*  } | *instruccionesX86*<*condiciónLogi*>  *JnoCond* **finSi**  *Bfi*  **finSi:** |

Estructura Condicional Selectiva *IF-THEN-ELSE* ----------------

*condiciónLogi*

*then* V

F *else*

*IF*

# Bfi1 --------

# Bfi2 --------

*Bfi1* y *Bfi2* representando un grupo o bloque de instrucciones

Sintaxis en lenguaje Java y Ensamblador x86.

|  |  |
| --- | --- |
| Lenguaje Java | Lenguaje Ensamblador x86 |
| **if** ( *condiciónLogi* )  {  *Bfi1*  }  **else**  {  *Bfi2*  } | *instruccionesX86*<*condiciónLogi*>  *JnoCond* **else**  *Bfi1*  JMP **finSi**  **else:**  *Bfi2*  **finSI:** |

Estructura Condicional Selectiva *SWITCH o CASE* ----------------

*SWITCH expresión*

(aritmética o carácter)

# Bfi1 -------

*val1*

# Bfi2 -------

*val2*

# Bfi3 -------

*val3*

valor distinto a todos los *val#*

# BfiD -------

# ........

*Bfi*, *Bfi2*, *Bfi3*, … y *BfiD*, representan lo mismo que *Bfi* en la estructura *IF-THEN*.

Sintaxis en el lenguaje Java

**switch (** *expresión* **)**

**{**

**case** *val 1***:**

*Bfi1*

**break;**

**case** *val 2***:**

*Bfi2*

**break;**

………………..

**default** **:**

*BfiD*

**break;**

**}**

Las constantes *val 1*, *val 2*, *val 3*, etc. pueden ser numéricas enteras (p.e. **case** 15**:**) ó caracteres (p.e. **case** ‘C’**:**).

**FRASES CONDICIONALES REPETITIVAS**

Estructura Condicional Repetitiva *WHILE* ----------------

# Bfi

*WHILE*

*condiciónLogi*

V

F

Sintaxis en lenguaje Java y Ensamblador x86.

|  |  |
| --- | --- |
| Lenguaje Java | Lenguaje Ensamblador x86 |
| **while** ( *condiciónLogi* )  {  *Bfi*  } | **iniWh:**  *instruccionesX86*<*condiciónLogi*>  *JnoCond* **finWh**  *Bfi*  JMP **iniWh**  **finWh:** |

Estructura Condicional Repetitiva *DO-WHILE*. ----------------

*DO*

*while*

# Bfi

*condiciónLogi*

V

F

Sintaxis en lenguaje Java y Ensamblador x86.

|  |  |
| --- | --- |
| Lenguaje Java | Lenguaje Ensamblador x86 |
| **do**  {  *Bfi*  }  **while** ( *condiciónLogi* ) | **iniDo:**  *Bfi*  *instruccionesX86*<*condiciónLogi*>  *Jcond* **iniDo** |

Estructura Condicional Repetitiva *FOR* , en sus dos formas *ascendente* y *descendente*. ----------------

ascendente descendente

# Bfi

*FOR*

*vc <= valfin*

V

F

# vc <- vc + inc

# vc <- valini

# Bfi

*FOR*

*vc >= valini*

V

F

# vc <- vc - dec

# vc <- valfin

Sintaxis en el lenguaje Java

**for (** *vc=valini; vc<=valfin; vc=vc+ini* **)** **for (** *vc=valfin; vc>=valini; vc=vc-dec* **)**

**{** **{**

*Bfi*  *Bfi*

**}** **}**

donde la *expre-lógica* y asignaciones que involucran la variable de control *vc* , quedan embebidas en la estructura del FOR.

A continuación se muestra el equivalente WHILE de las estructuras FOR.

*vc = valini* ***;*** *vc = valfin* ***;***

**while (** *vc <= valfin* **)** **while (** *vc >= valini* **)**

**{**  **{**

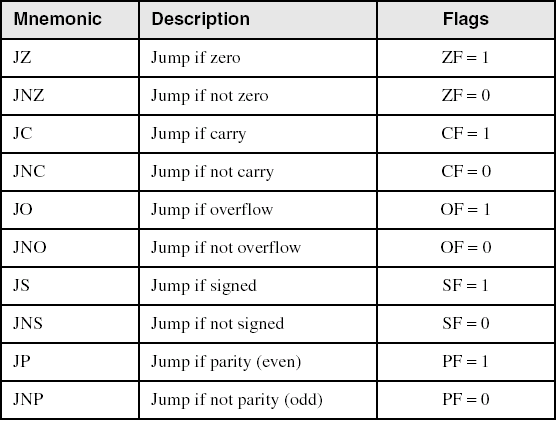
*Bfi*  *Bfi*

*vc = vc + inc* ***;*** *vc = vc - dec* ***;***

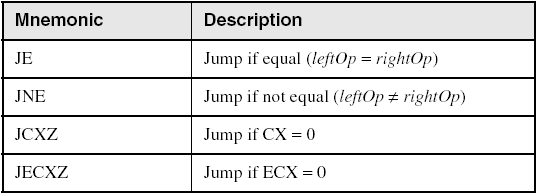
**}** **}**

En todo lo anterior *Bfi* sigue la misma regla que en el caso de contener una sóla instrucción.

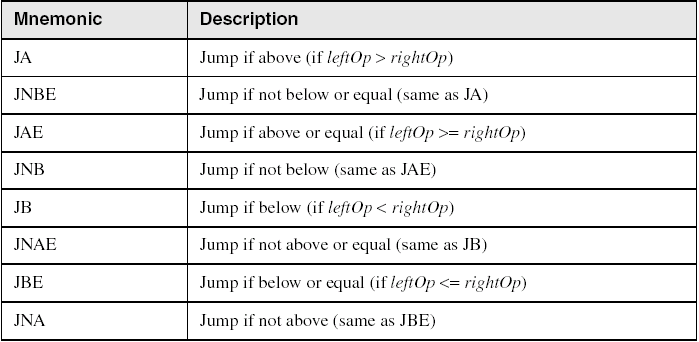
**J*conds* basados en Banderas específicas.**



**J*conds* basados en Igualdad.**



**J*conds* basados en operandos Sin Signo.**



**J*conds* basados en operandos Con Signo.**

