Estructuras Alternativas

Introducción a la Programación
Facultad de Informática
Univ.Nac. del Comahue

Estructuras de Control

Las Estructuras de Control controlan la secuencia, orden o flujo de ejecución de las instrucciones de un programa

- Podemos dividirlas en tres categorías:
  - Secuencial
  - Alternativa o de Condición ⟵**HOY**
  - Iterativa o Repetitiva

Estructura Alternativa o Condicional

Estructuras que **dirigen la ejecución** de un programa hacia un grupo de instrucciones u otro dependiendo de una **condición** o **expresión booleana** .

¡Nos basamos en una condición para decidir qué conjunto de instrucciones se debe ejecutar!

¡Permite tomar decisiones!

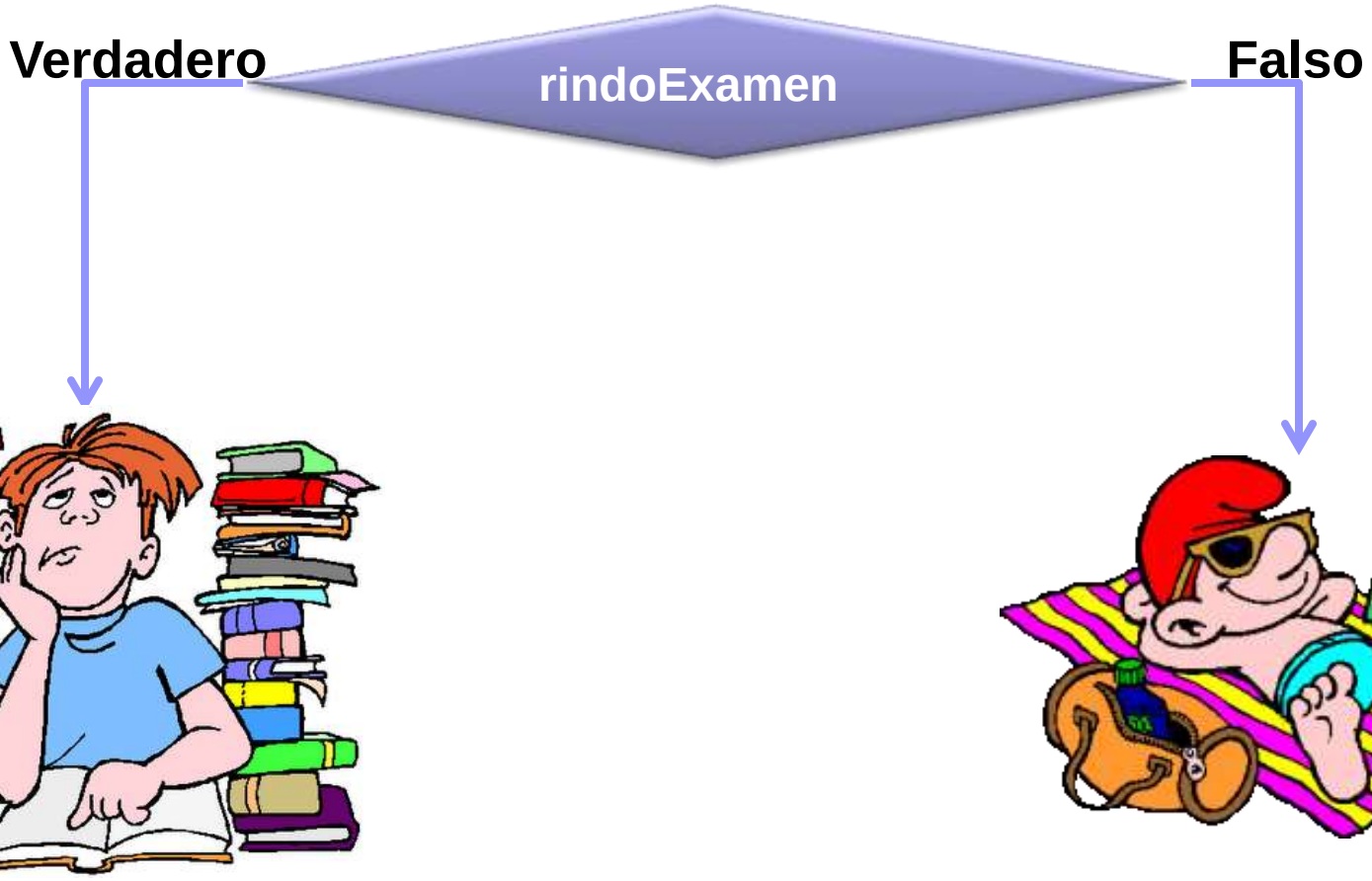
Estructura Alternativa
(en la vida cotidiana: Escuela Primaria)

TemperaturaCorporal > 37

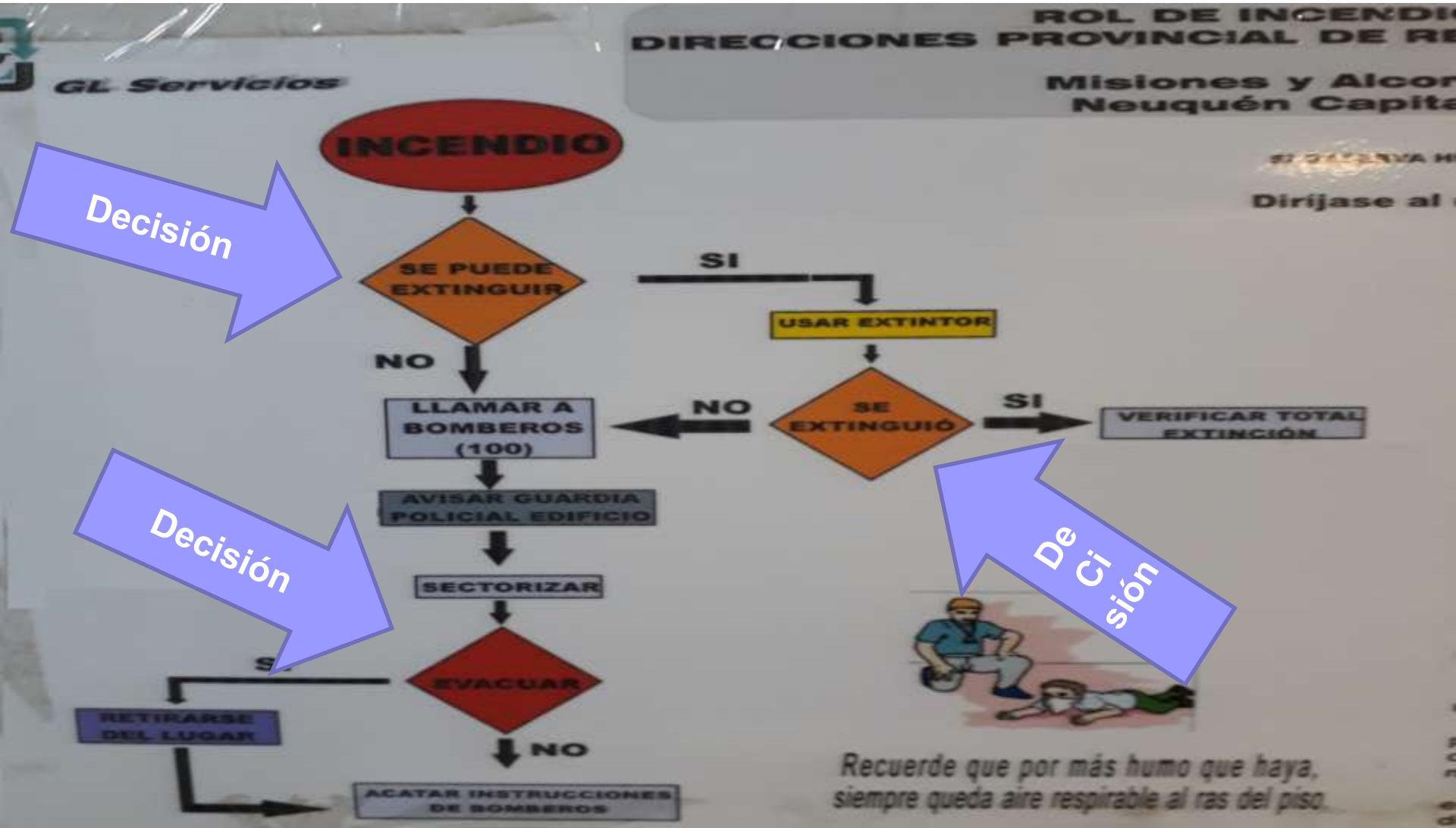
Verdadero

Falso

Estructur Alternativa
(en la vida cotidiana: En la Universidad)



Estructura Alternativa
¿Qué hacer ante un incendio?



Estructuras Alternativas

Para este tipo de estructuras veremos la instrucción SI (pseudocódigo) / IF (PHP) y sus variantes:

Pseudocódigo:	PHP:
a. SI	IF
b. SI .. SINO	IF .. ELSE
c. SI .. OTRO-SI	IF .. ELSEIF
d. SI .. OTRO-SI .. SINO	IF .. ELSEIF .. ELSE

Estructuras Alternativas

Pseudocódigo:		PHP:
a.	SI	IF
b.	SI .. SINO	IF .. ELSE
c.	SI .. OTRO-SI	IF .. ELSEIF
d.	SI .. OTRO-SI .. SINO	IF .. ELSEIF .. ELSE

Lo que tiene que quedar claro: Cuando escribimos esta instrucción, la palabra reservada **SI** (pseudocódigo)/ **IF**(PHP), indica el comienzo de la instrucción alternativa, e **indica que en el código fuente se tomará una decisión basados en una condición**.

El resto de las palabras, OTRO-SI/ELSEIF, SINO/ELSE, son optativas: escribirlas o no dependerá del análisis del problema que estamos resolviendo.

Todas las instrucciones que hemos visto hasta el momento, son herramientas para programar. Utilizarlas o no siempre depende del problema a resolver. Es fundamental un buen análisis del mismo!.

Instrucción Alternativa IF
Sintaxis

Diseño: Pseudocódigo

```
SI (condición) ENTONCES
  instruccion1
  instruccion2
  ...
  instruccionN
FIN SI
```

PHP

```
if(condicion){
    instruccion1;
    instruccion2;
     …
    instruccionN;
}
```

Conjunto de instrucciones que se ejecutan si la condición es verdadera.

En esta variante de la instrucción SI, si la condición es falsa no se ejecuta ninguna instrucción

Identar = desplazar el texto 4 espacios hacia la derecha

instruccionX = Asignación, Lectura, Escritura, invocación a fc sin retorno, Alternativa

Instrucción Alternativa IF

Especificación (código fuente)

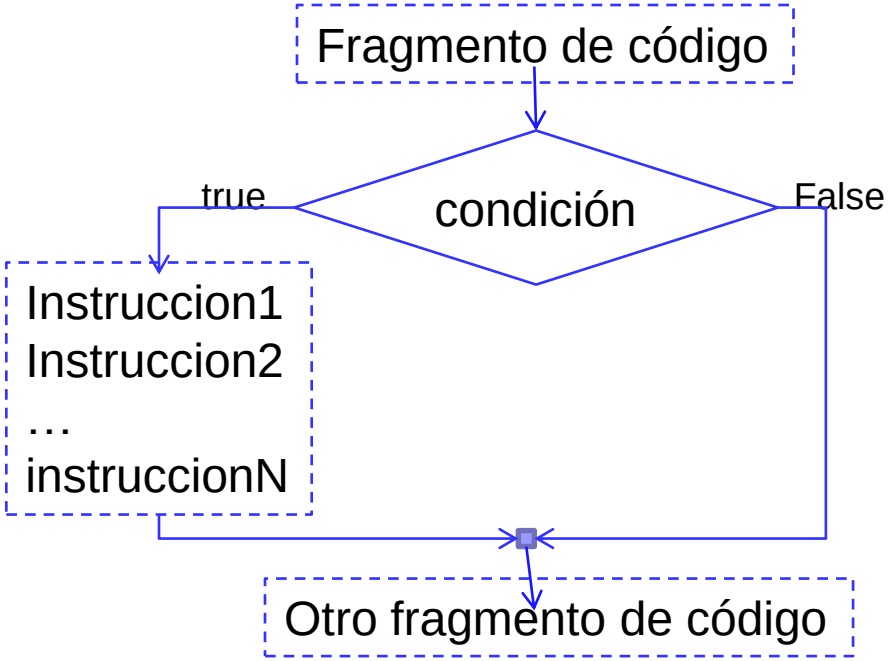
Diseño: Pseudocódigo

```
SI (condición) ENTONCES
        instruccion1
        instruccion2
        ...
        instruccionN
FIN SI
```

PHP

```
if (condicion){
    instruccion1;
    instruccion2;
     …
    instruccionN;
}
```

Ejecución



Utilizando este **Diagrama de flujo**, podemos interpretar cómo se ejecutará la instrucción alternativa. Las flechas indican el orden del flujo de ejecución:

En ejecución, si la condición es verdadera se ejecuta el bloque de instrucciones, si es falso no se ejecuta ningún bloque

Ejemplo

1- Debemos leer un número de teclado, si el número es mayor o igual a cero debemos escribir en pantalla el cartel “Numero positivo”

Entrada: (por teclado) unNumero entero

Salida: (por pantalla) un string (Texto) que indique si el número es positivo

Proceso: Debemos decidir si unNumero >= 0, si es así se debe escribir el cartel.

Diseño: Para concentrarnos en la instrucción alternativa, vamos a diseñar un programa principal.
Hay muchas formas de resolverlo: por ejemplo se podría diseñar un módulo cuyo parámetro formal sea un número y el retorno sea true o false si el número es o no positivo

Ejemplo

Entrada: (por teclado) unNumero entero

Proceso: Debemos decidir si unNumero >= 0, si es así se debe escribir el cartel.

Salida: (por pantalla) un string que indique si el número es positivo

```
PROGRAMA principal
        ENTERO unNumero
        ESCRIBIR ("Ingrese un número: ")
        LEER(unNumero)

        SI (unNumero>=0) ENTONCES
                ESCRIBIR (unNumero, " es positivo")
        FIN SI
        ESCRIBIR (".fin.")
FIN PROGRAMA
```

Diseño

```php
<?php
/***principal***/
/*int $unNumero*/
echo "Ingrese un número:";
$unNumero = trim(fgets(STDIN));

if ($unNumero>=0) {
    echo $unNumero." es positivo";
}

echo ".fin.";
```

PHP

Ejemplo

Cómo hay dos posibilidades en la ejecución (1° posibilidad: que la condición sea verdadera. 2° posibilidad: que la condición sea falsa) realizaremos 1 traza por cada posibilidad.

Traza 1: para el número 91

Traza 2: para el número -5

```
PROGRAMA principal
        Entero unNumero
        ESCRIBIR ("Ingrese un número: ")
        LEER(unNumero)
        SI (unNumero>=0) ENTONCES
                ESCRIBIR (unNumero, " es positivo")
        FIN SI
        ESCRIBIR (".fin.")
FIN PROGRAMA
```

Traza 1 Algoritmo principal

unNumero	Salida/Pantalla
91	Ingrese un número:
	91 es positivo
	.fin.

Traza 2 Algoritmo principal

unNumero	Salida/Pantalla
-5	Ingrese un número:
	.fin.

Instrucción IF .. ELSE
Sintaxis

Diseño

```
SI (condición) ENTONCES
    instruccion1
    instruccion2
    ...
    instruccionN
SINO
    instruccion_b1
    instruccion_b2
    ...
    instruccion_bN
FIN SI
```

Conjunto de instrucciones que se ejecutan si la condición es verdadera.

Conjunto de instrucciones que se ejecutan si la condición es falsa.

instruccionX = Asignación, Lectura, Escritura, invocación a fc sin retorno, Alternativa

PHP

```
if(condicion){
    instruccion1;
    instruccion2;
     …
    instruccionN;
} else {
    instruccion_b1;
    instruccion_b2;
    …
    instruccion_bN;
}
```

Instrucción IF .. ELSE

Especificación (código fuente)

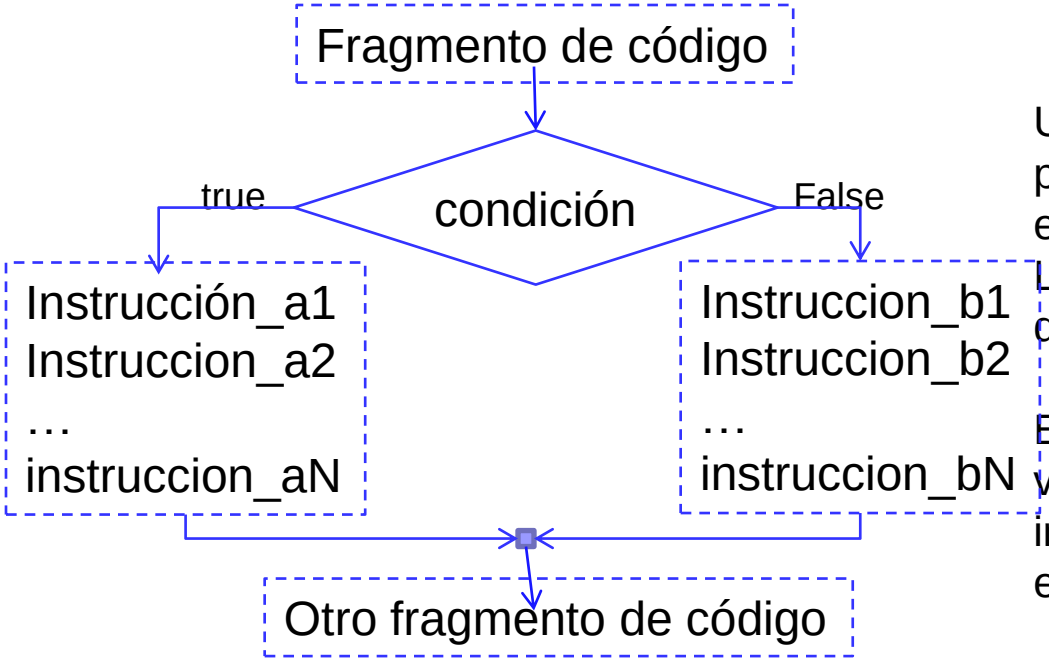
Diseño

```
SI (condición) ENTONCES
        instruccion_a1
        instruccion_a2
        ...
        instruccion_aN
SINO
        instruccion_b1
        instruccion_b2
        ...
        instruccion_bN
FIN SI
```

PHP

```
if (condicion){
    instruccion_a1;
    instruccion_a2;
     …
    instruccion_aN;
} else {
    instruccion_b1;
    instruccion_b2;
    …
    instruccion_bN;
}
```

Ejecución



Utilizando este **Diagrama de flujo**, podemos interpretar cómo se ejecutará la instrucción alternativa. Las flechas indican el orden del flujo de ejecución:

En ejecución, si la condición es verdadera se ejecuta el bloque de instrucciones A, si es falsa se ejecuta el bloque de instrucciones B

Ejemplo

2- Debemos leer un número de teclado, si el número es mayor o igual a cero debemos escribir en pantalla el cartel “Numero positivo”. Caso contrario debemos escribir en pantalla el cartel “Numero negativo”

Entrada: (por teclado) unNumero entero

Salida: (por pantalla) un string que indique si el número es positivo

Proceso: Debemos decidir SI unNumero >= 0, si es así se debe escribir el cartel nro positivo, SINO es así, escribimos nro negativo

Diseño: Para concentrarnos en la instrucción alternativa, vamos a diseñar un programa principal.

Ejemplo

Entrada: (por teclado) unNumero entero

Proceso: Debemos decidir SI unNumero >= 0, si es así se debe escribir el cartel nro positivo, SINO es así, escribimos nro negativo

Salida: (por pantalla) un string que indique si el número es positivo

```
PROGRAMA principal
        Entero unNumero
        ESCRIBIR ("Ingrese un número: ")
        LEER(unNumero)

        SI (unNumero>=0) ENTONCES
                ESCRIBIR (unNumero, " es positivo")
        SINO
                ESCRIBIR (unNumero, " es negativo")
        FIN SI

        ESCRIBIR (".fin.")
FIN PROGRAMA
```

Diseño

```php
/***principal***/
/*Int $unNumero*/
echo "Ingrese un número:";
$unNumero = trim(fgets(STDIN));

if ($unNumero>=0) {
    echo $unNumero." es positivo";
}
else {
    echo $unNumero." es negativo";
}

echo ".fin.";
```

PHP

Ejemplo

Cómo hay dos posibilidades en la ejecución (1° posibilidad: que la condición sea verdadera. 2° posibilidad: que la condición sea falsa) realizaremos 1 traza por cada posibilidad.

Traza 1: para el número 91

Traza 2: para el número -5

```
PROGRAMA principal
        Entero unNumero
        ESCRIBIR ("Ingrese un número: ")
        LEER(unNumero)
        SI (unNumero>=0) ENTONCES
            ESCRIBIR (unNumero, " es positivo")
        SINO
            ESCRIBIR (unNumero, " es negativo")
        FIN SI
        ESCRIBIR (".fin.")
FIN PROGRAMA
```

Traza 1 Algoritmo principal

unNumero	Salida/Pantalla
91	Ingrese un número:
	91 es positivo
	.fin.

Traza 2 Algoritmo principal

unNumero	Salida/Pantalla
-5	Ingrese un número:
	-5 es negativo
	.fin.

Instrucción IF..ELSEIF

Múltiples decisiones consecutivas

SINTAXIS

Diseño

```
SI (condición_1) ENTONCES
    instruccion_a1
    ...
OTRO-SI (condición_2) ENTONCES
    instruccion_b1
    ...
OTRO-SI (condición_N) E
    instruccion_n1
    ...
SINO
    instruccion_1
    ...
FIN SI
```

PHP

```
if (condicion_1){
    instruccion_a1;
    …
} elseif(condicion_2){
    instruccion_b1;
    …
} elseif(condicion_N){
    instruccion_n1;
    …
} else {
}
```

instrucciones que se ejecutan si condición_1 es true.

instrucciones que se ejecutan si condición_1 es false y condición_2 es true

instrucciones que se ejecutan si condición_1 es false, condición_2 es false condición_3 es true

instrucciones que se ejecutan si todas las condiciones anteriores son false,

Instrucción IF..ELSEIF

Múltiples decisiones consecutivas

SINTAXIS

Diseño

```
SI  (condición_1) ENTONCES
      instruccion_a1
      ...
OTRO-SI  (condición_2) ENTONCES
      instruccion_b1
      ...
OTRO-SI (condición_N) ENTONCES
      instruccion_n1
      ...
SINO
      instruccion_1
      ...
FIN SI
```

PHP

```
if (condicion_1){
    instruccion_a1;
    …
} elseif(condicion_2){
    instruccion_b1;
    …
} elseif(condicion_N){
    instruccion_n1;
    …
}else {
    instruccion_1;
    …
}
```

Instrucción IF..ELSEIF

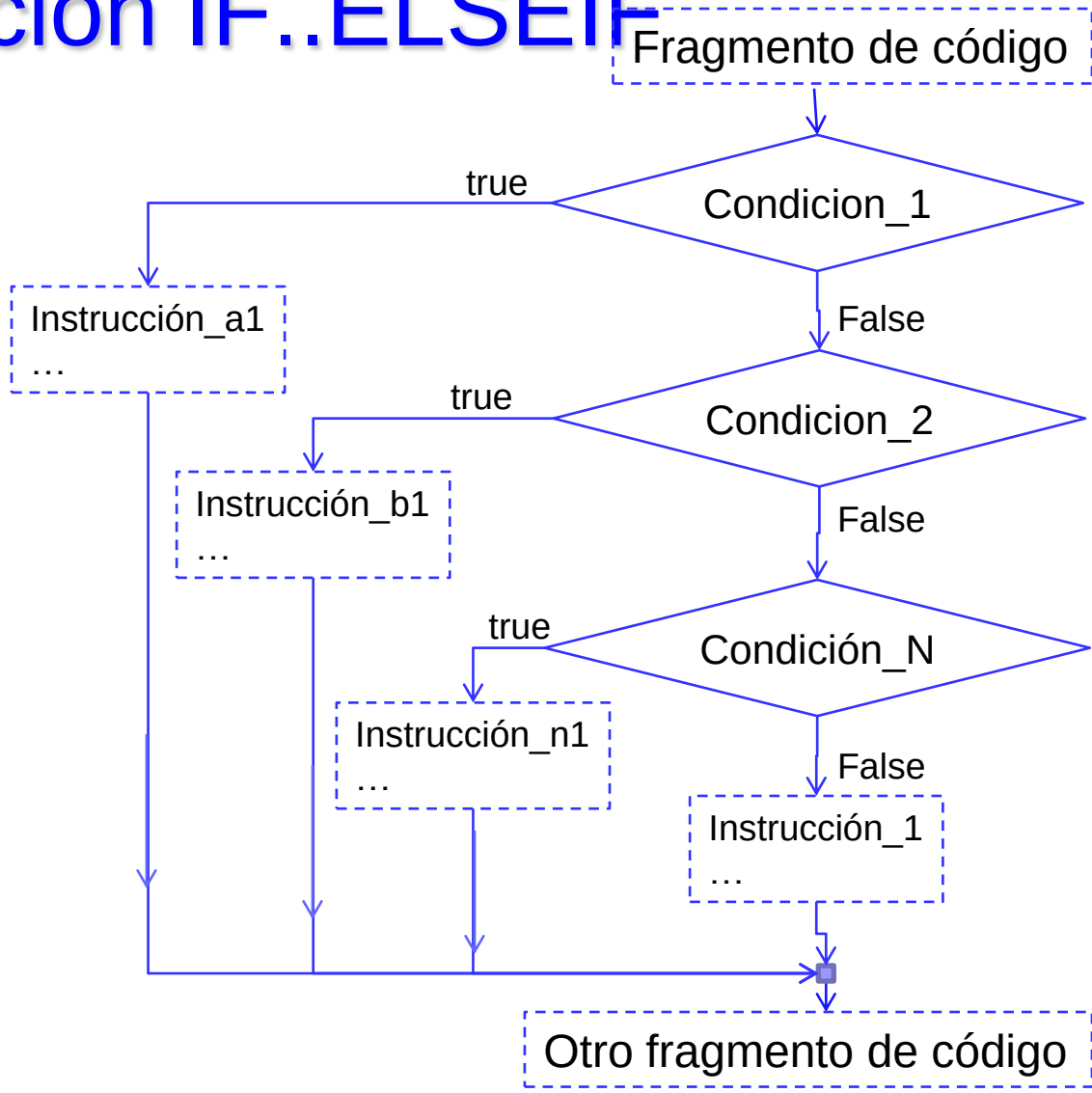
Diseño

SI (*condición_1*) **ENTONCES**
instruccion_a1
...
OTRO-SI (*condición_2*) **ENTONCES**
instruccion_b1
...
OTRO-SI (*condición_N*) **ENTONCES**
instruccion_n1
...
SINO
instruccion_1
...
FIN SI

PHP

```
if (condicion_1){
    instruccion_a1;
    …
} elseif (condicion_2){
    instruccion_b1;
    …
} elseif (condicion_N) {
    instruccion_n1;
    …
} else {
    instruccion_1;
    …
}
```

Especificación (código fuente)



Ejecución

Instrucción IF..ELSEIF

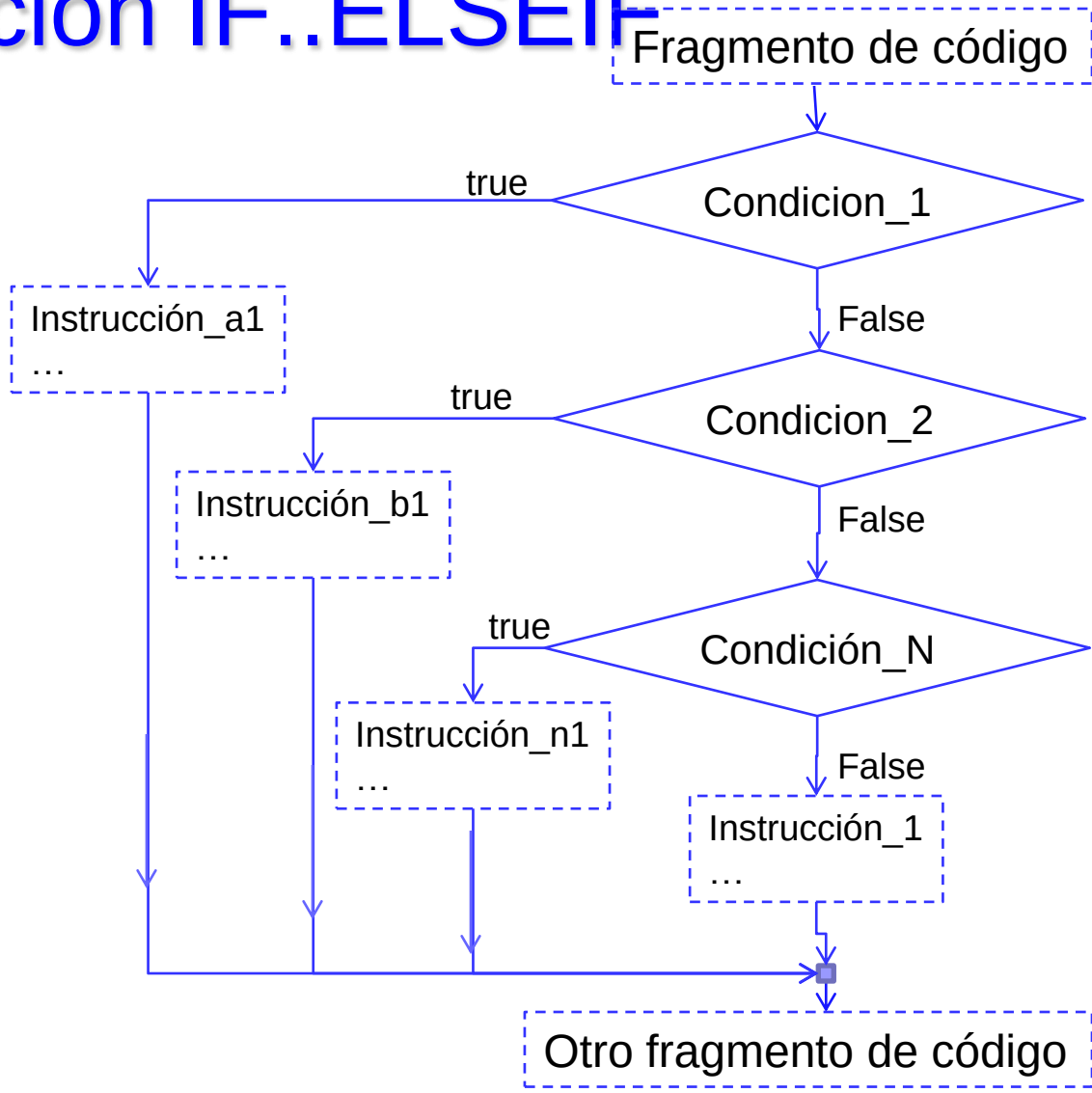
Diseño

SI (*condición_1*) **ENTONCES**
instruccion_a1
...
OTRO-SI (*condición_2*) **ENTONCES**
instruccion_b1
...
OTRO-SI (*condición_N*) **ENTONCES**
instruccion_n1
...
SINO
instruccion_1
...
FIN SI

PHP

```
if (condicion_1){
    instruccion_a1;
    …
} elseif (condicion_2){
    instruccion_b1;
    …
} elseif (condicion_N) {
    instruccion_n1;
    …
} else {
    instruccion_1;
    …
}
```

Especificación (código fuente)



Ejecución

Ejemplo

3- Debemos leer un número, si es positivo escribimos “Es positivo”, si es cero escribimos “Es cero”, sino “Es negativo”.

Entrada: (por teclado) unNumero entero

Salida: (por pantalla) un string que indique si el número es positivo

Proceso: Debemos decidir SI unNumero > 0, si es así se debe escribir el cartel nro positivo, si unNumero = 0 se debe escribir es cero, SINO, escribimos nro negativo

Diseño: Para concentrarnos en la instrucción alternativa, vamos a diseñar un programa principal.

Ejemplo

Entrada: (por teclado) unNumero entero

Proceso: Debemos decidir SI unNumero > 0, si es así se debe escribir el cartel nro positivo, si unNumero = 0 se debe escribir es cero, SINO, escribimos nro negativo

Salida: (por pantalla) un string que indique si el número es positivo

```
PROGRAMA principal
        Entero unNumero
        ESCRIBIR ("Ingrese un número: ")
        LEER(unNumero)

        SI (unNumero>0) ENTONCES
                ESCRIBIR (unNumero, " es positivo")
        OTRO-SI (unNumero=0) ENTONCES
                ESCRIBIR (unNumero, " es cero")
        SINO
                ESCRIBIR (unNumero, " es negativo")
        FIN SI

        ESCRIBIR (".fin.")
FIN PROGRAMA
```

Diseño

```php
/***principal***/
/*Int $unNumero*/
echo "Ingrese un número:";
$unNumero = trim(fgets(STDIN));
if ($unNumero>0) {
    echo $unNumero." es positivo";
}
elseif ($unNumero==0) {
        echo $unNumero." es cero";
    }
else {
        echo $unNumero." es negativo";
    }
echo ".fin.";
```

PHP

Ejemplo

Cómo hay tres posibilidades en la ejecución (1° posibilidad: que la 1° condición sea verdadera. 2° posibilidad: la 1° condición sea falsa y la 2°condición verdadera. La 3° posibilidad: que la 1° condición sea falsa y la 2° condición sea falsa). realizaremos 1 traza por cada posibilidad.

Traza 1: para el número 91
Traza 2: para el número -5
Traza 3: para el número 0

```
PROGRAMA principal
        Entero unNumero
        ESCRIBIR ("Ingrese un número: ")
        LEER(unNumero)
        SI (unNumero>0) ENTONCES
                ESCRIBIR (unNumero, " es positivo")
        OTRO-SI (unNumero=0) ENTONCES
                ESCRIBIR (unNumero, " es cero")
        SINO
                ESCRIBIR (unNumero, " es negativo")
        FIN SI
        ESCRIBIR (".fin.")
FIN PROGRAMA
```

Traza 1 Algoritmo principal

unNumero	Salida/Pantalla
91	Ingrese un número:
	91 es positivo
	.fin.

Traza 2 Algoritmo principal

unNumero	Salida/Pantalla
-5	Ingrese un número:
	-5 es negativo
	.fin.

Ejemplo

Cómo hay tres posibilidades en la ejecución (1° posibilidad: que la 1° condición sea verdadera. 2° posibilidad: la 1° condición sea falsa y la 2°condición verdadera. La 3° posibilidad: que la 1° condición sea falsa y la 2° condición sea falsa). realizaremos 1 traza por cada posibilidad.

Traza 1: para el número 91
Traza 2: para el número -5
Traza 3: para el número 0

```
PROGRAMA principal
        Entero unNumero
        ESCRIBIR ("Ingrese un número: ")
        LEER(unNumero)
                    false
        SI (unNumero>0) ENTONCES
                ESCRIBIR (unNumero, " es positivo")
                          true
        OTRO-SI (unNumero=0) ENTONCES
                ESCRIBIR ( unNumero, " es cero")
        SINO
                ESCRIBIR (unNumero, " es negativo")
        FIN SI
        ESCRIBIR (".fin.")
FIN PROGRAMA
```

Traza 3 Algoritmo principal

unNumero	Salida/Pantalla
0	Ingrese un número:
	0 es cero
	.fin.

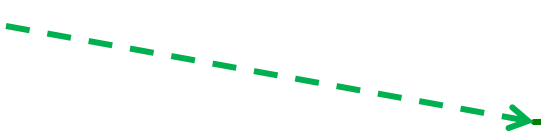
Las instrucciones alternativas las podemos combinar de muchas formas para resolver un problema. Las podemos:

- **Anidar**

- **Escribir en forma secuencial**

Anidar Instrucciones

SI (*condición_1*) **ENTONCES**
instruccion_a1
instrucción_a2
instrucción_aN
SINO
instruccion_b1
instrucción_b2
instrucción_bM
FIN SI



SI (*condición_1*) **ENTONCES**
instruccion_a1
SI (*condición_a1*) **ENTONCES**
instruccion_aa1
instrucción_aa2
instrucción_aaN
FIN SI
instrucción_aN
SINO
instruccion_b1
instrucción_b2
instrucción_bM
FIN SI

instruccionX = Asignación, Lectura, Escritura, invocación a fc sin retorno, Alternativa

Anidar Instrucciones

SI (*condición_1*) **ENTONCES**
instruccion_a1
instrucción_a2
instrucción_aN
SINO
instruccion_b1
instrucción_b2
instrucción_bM
FIN SI

SI (*condición_1*) **ENTONCES**
instruccion_a1
SI (*condición_a1*) **ENTONCES**
instruccion_aa1
instrucción_aa2
instrucción_aaN
FIN SI
instrucción_aN
SINO
SI (*condición_b1*) **ENTONCES**
instruccion_ba1
instrucción_ba2
instrucción_baN
SINO
instruccion_ba1
instrucción_ba2
FIN SI
instrucción_b2
instrucción_bM
FIN SI

Es muy importante identar las instrucciones!

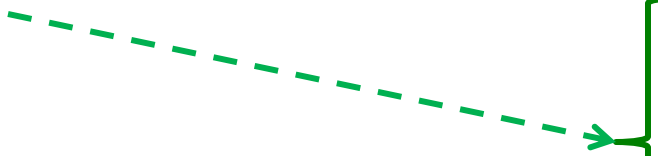
Anidar Instrucciones

SI (*condición_1*) **ENTONCES**
instruccion_a1
SI (*condición_a1*) **ENTONCES**
instruccion_aa1
instrucción_aa2
instrucción_aaN
FIN SI
instrucción_aN
SINO
SI (*condición_b1*) **ENTONCES**
instruccion_ba1
instrucción_ba2
instrucción_baN
SINO
instruccion_ba1
instrucción_ba2
FIN SI
instrucción_b2
instrucción_bM
FIN SI

```
if (condicion_1){
    instruccion_a1;
    if (condicion_a1){
        instruccion_aa1;
        instruccion_aa2;
        instruccion_aaN;
    }
    instruccion_aN;
} else {
    if (condicion_a1){
        instruccion_ba1;
        instruccion_ba2;
        instruccion_baN;
    } else {
        instruccion_ba1;
        instruccion_ba2;
    }
    instruccion_b2;
    instruccion_bN;
}
```

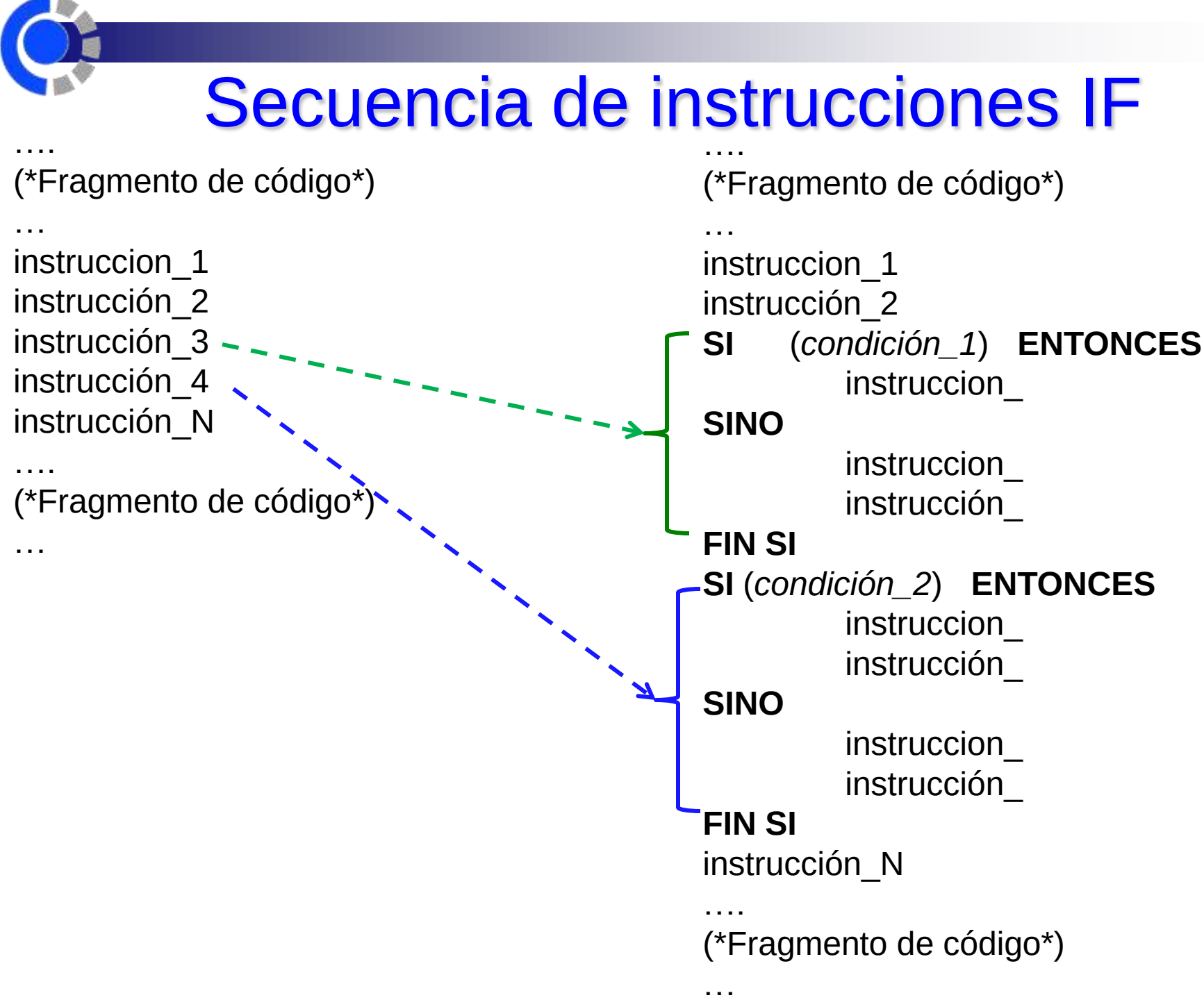
Secuencia de instrucciones IF

….
(*Fragmento de código*)
…
instruccion_1
instrucción_2
instrucción_3
instrucción_4
instrucción_N
….
(*Fragmento de código*)
…



….
(*Fragmento de código*)
…
instruccion_1
instrucción_2
SI (*condición_1*) **ENTONCES**
instruccion_
SINO
instruccion_
instrucción_
FIN SI
instrucción_4
instrucción_N
….
(*Fragmento de código*)
…

Secuencia de instrucciones IF



Las instrucciones alternativas las podemos combinar de muchas formas para resolver un problema. Las podemos:

- **Anidar**
- **Escribir en forma secuencial**

(Hacer el ejercicio de la siguiente filmina)

Ejercicio:

Una empresa asigna los sueldos a sus empleados teniendo en cuenta el cargo, su categoría y su antigüedad.
* Los cargos son “DIRECTOR”, “JEFE” o “EMPLEADO”
* Existen dos posibles categorías dentro de la empresa: 1 y 2
* La antigüedad son los años que el empleado trabaja en la empresa.

El sueldo básico se calcula de la siguiente manera: Si se trata de un Director su sueldo asciende a $65952, si se trata de un Jefe $48000. Para cualquier otro caso, se establece que si es categoría 1 cobra $35000, y si es categoría 2 cobra $30000.

Por otra parte, el sueldo básico se incrementa en un 10% cuando la antigüedad supera los 15 años.

La tarea para el programador es: Especificar un módulo en pseudocódigo que calcule el sueldo con los parámetros formales necesarios para resolver el problema y retorne el sueldo completo.

Es necesario que realice trazas para testear el módulo. Al menos para los siguientes ejemplos:
a) DIRECTOR, categoria 1, 25 años de antigüedad
b) EMPLEADO, categoria 2, 3 años de antigüedad.

Traducir a PHP

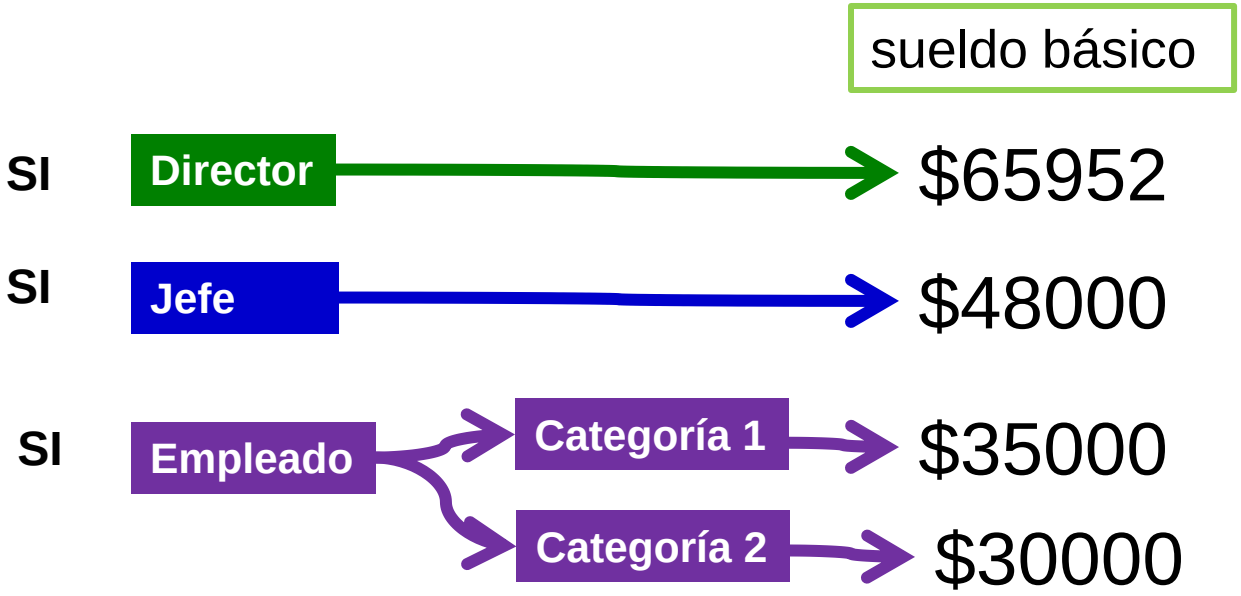
Ejercicio:

Una empresa asigna los sueldos a sus empleados teniendo en cuenta el cargo, su categoría y su antigüedad.

* Los cargos son “DIRECTOR”, “JEFE” o “EMPLEADO”
* Existen dos posibles categorías dentro de la empresa: 1 y 2
•La antigüedad son los años que el empleado trabaja en la empresa.

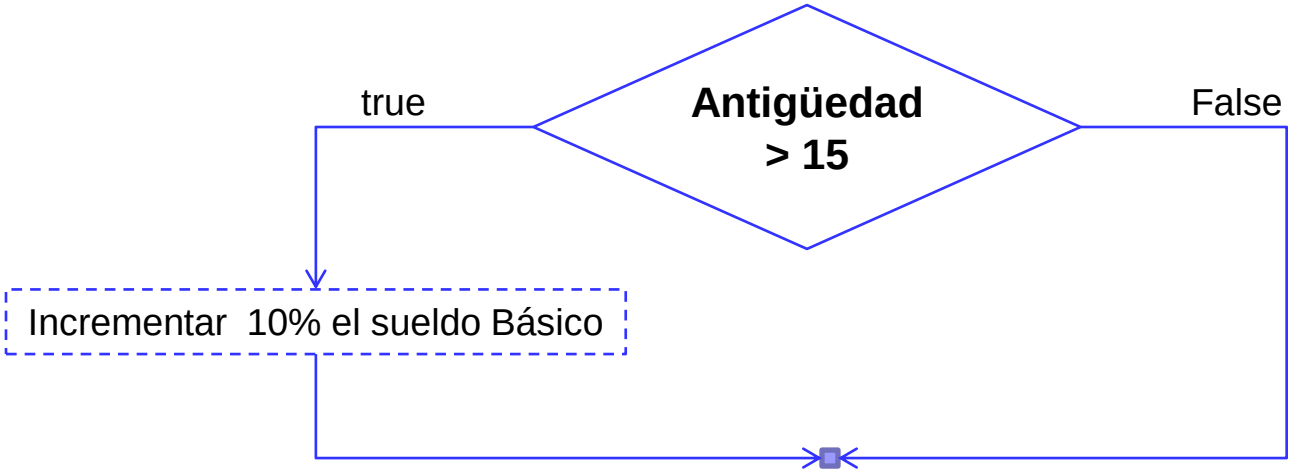
Ejercicio:

El sueldo básico se calcula de la siguiente manera: Si se trata de un Director su sueldo asciende a $65952, si se trata de un Jefe $48000. Para cualquier otro caso, se establece que si es categoría 1 cobra $35000, y si es categoría 2 cobra $30000.



Ejercicio:

Por otra parte, el sueldo básico se incrementa en un 10% cuando la antigüedad supera los 15 años.



Ejercicio:

La tarea para el programador es: **Especificar un (1) módulo** en **pseudocódigo** que calcule el sueldo con los parámetros formales necesarios para resolver el problema y retorne el sueldo completo.

Es necesario que realice **trazas** para testear el módulo. Al menos para los siguientes ejemplos:

a) DIRECTOR, categoria 1, 25 años de antigüedad

b) EMPLEADO, categoria 2, 3 años de antigüedad.

Traducir a PHP