



LOGALI

TEORÍA

Bifurcaciones condicionales

SAP ABAP Programación





Contenido

1. Resumen de la lección	3
2. Sentencia IF/ELSE/ENDIF	3
3. Sentencia CASE/ENDCASE	3
4. Sentencia DO/ENDO	4
5. Sentencia WHILE/ENDWHILE	5



1. Resumen de la lección

En ABAP existen varias formas de ejecutar distintas secuencias de sentencias, según determinadas condiciones. A continuación, se detallan todas las posibilidades que nos ofrece ABAP.

2. Sentencia IF/ELSE/ENDIF

En la **construcción IF** puede definir **cualquier expresión lógica** como condición de verificación. Si se cumple la condición, el sistema ejecuta el bloque de sentencias correspondiente. De lo contrario, se verifica la condición específica en la siguiente bifurcación **ELSEIF** (es posible incluir varias bifurcaciones). Si no cumple ninguna de las condiciones específicas, se ejecuta la bifurcación **ELSE**, si es que existe. Las bifurcaciones **ELSEIF** y **ELSE** son opcionales.

```
IF gv_var > 0.  
    * Sentencias  
ELSEIF gv_var = 0.  
    * Sentencias  
ELSE.  
    * Sentencias  
ENDIF.
```

3. Sentencia CASE/ENDCASE

La construcción **CASE** se utiliza para distinguir los casos de forma clara. El contenido del campo especificado en la parte CASE se verifica con respecto a los objetos de datos listados en la bifurcación **WHEN** para comprobar si coinciden. Si el contenido del campo coincide, se procesa el bloque de sentencias correspondiente. Si ninguna comparación



coincide, el sistema ejecuta la bifurcación **OTHERS** si está disponible. Excepto la primera bifurcación **WHEN**, todos los otros suplementos son opcionales.

```
□ CASE gv_var.  
  ○ WHEN 'A'.  
    * Sentencias  
  ○ WHEN 'B'.  
    * Sentencias  
  ○ WHEN OTHERS.  
    * Sentencias  
  └─ ENDCASE.
```

4. Sentencia DO/ENDO

El bloque de sentencia **DO** y **ENDDO** se ejecuta de forma continua hasta que se abandona la iteración mediante sentencias de terminación como **EXIT**. También tiene la opción de especificar el número máximo de ejecuciones; en caso contrario es posible que el resultado sea un **bucle sin fin** o **bucle infinito**.

```
* El bloque de código se va ejecutar 10 veces  
□ DO 10 TIMES.  
  * Sentencias  
  └─ ENDDO.  
  
▶ □ * El bloque de código se va ejecutar hasta  
▶   * que se cumple la condición <abort_condition>  
▶ □ DO.  
  □ IF <abort_condition>.  
    EXIT.  
  └─ ENDIF.  
  └─ ENDDO.
```



5. Sentencia WHILE/ENDWHILE

El bloque de sentencia **WHILE** y **ENDWHILE** se ejecuta de forma continua hasta que no se cumpla la condición especificada. Siempre se verifica la condición especificada. Siempre verifica la condición antes de ejecutar el bloque de sentencias.

```
WHILE <condicion>.  
    * Sentencias  
ENDWHILE.
```