

# Select-Case y Buenas prácticas

[Descripción]

## Resumen Ejecutivo

[Descripción de resumen Ejecutivo]

## Objetivo

- Introducir la estructura Select Case para simplificar la toma de decisiones en base a un valor numérico (la nota).
- Mostrar cómo usar Select Case para asociar valores numéricos a resultados específicos.
- Demostrar el uso de Select Case para responder a diferentes niveles de desempeño
- Practicar Select Case con rangos de valores para categorizar edades.
- Demostrar buenas prácticas en sentencias condicionales, como verificar condiciones de borde y limitar la cantidad de anidación.
- Demostrar el uso de Select Case para responder a diferentes niveles de desempeño
- Demostrar cómo estructurar condicionales para validar entradas y realizar asignaciones de forma clara y ordenada.

## Introducción

Al ingresar al mundo de la programación, uno de los elementos clave es la toma de decisiones mediante estructuras condicionales. Estas permiten que un programa ejecute distintas acciones basadas en condiciones o valores específicos. Entre las herramientas que facilitan esta lógica en Visual Basic se encuentra la estructura Select Case.

¿Qué es Select Case? Select Case es una estructura condicional utilizada para evaluar el valor de una variable y ejecutar distintos bloques de código según el valor de dicha variable. Esta estructura es ideal cuando se tienen múltiples condiciones relacionadas con un solo valor (por ejemplo, clasificar días de la semana, categorías de edad, o rangos de calificación). Select Case mejora la claridad del código y permite organizar las decisiones de forma ordenada y comprensible, evitando el uso excesivo de If...Elseif.

**Buenas Prácticas en Sentencias Condicionales** El uso de If...Elseif y Select Case requiere seguir ciertas buenas prácticas para que el código sea más claro, eficiente y fácil de mantener.

## Contenido

1. Select-Case .....	3
Pongamos en Práctica la Teoría.....	3
Caso de estudio 1 .....	3
Caso de estudio 2 Determinar el Día de la Semana.....	4
Caso de estudio 3 Calificación de Desempeño en el Trabajo.....	5
Caso de Estudio 4: Categoría de Edad.....	6
2. <b>Buenas Prácticas en sentencias condicionales</b> .....	7
Pongamos en Práctica la Teoría.....	7
Caso de Estudio 1: Validación de Notas de Estudiantes .....	7
Caso de Estudio 2: Determinación de Etapas de Vida .....	8
Caso de Estudio 3: Cálculo de Descuento en Compra.....	9
Caso de Estudio 4: Determinación de Horarios según el Turno .....	10

# 1. Select-Case

## Pongamos en Práctica la Teoría

### Caso de estudio 1

Caso de estudio: Clasificación de Notas de Estudiantes

**Objetivo:**

Introducir la estructura Select Case para simplificar la toma de decisiones en base a un valor numérico (la nota)

Descripción: Un profesor quiere clasificar las notas de un estudiante en función de su valor para dar retroalimentación personalizada.

Desarrollo:

- Crear un TextBox para que el usuario ingrese la nota (0 a 100).
- Usar Select Case para evaluar el valor de la nota y clasificarla en rangos.
- Mostrar la clasificación en un Label.

Solución:

```
Dim nota As Integer = Val(txtNota.Text)
Dim clasificacion As String
Select Case nota
    Case 90 To 100
        clasificacion = "Excelente"
    Case 80 To 89
        clasificacion = "Muy Bueno"
    Case 70 To 79
        clasificacion = "Bueno"
    Case 60 To 69
        clasificacion = "Suficiente"
    Case Else
        clasificacion = "Reprobado"
End Select
lblResultado.Text = "Clasificación: " & clasificacion
```

## Caso de estudio 2 Determinar el Día de la Semana

Caso de estudio: Determinar el Día de la Semana

### Objetivo:

Mostrar cómo usar Select Case para asociar valores numéricos a resultados específicos.

### Descripción:

Una aplicación solicita un número del 1 al 7 y muestra el día de la semana correspondiente.

Desarrollo:

- Crear un TextBox para ingresar el número.
- Utilizar Select Case para evaluar el número y mostrar el día correspondiente.
- Mostrar el resultado en un Label.

Solución:

```
Dim diaNumero As Integer = Val(txtNumeroDia.Text)
Dim diaSemana As String

Select Case diaNumero
    Case 1
        diaSemana = "Lunes"
    Case 2
        diaSemana = "Martes"
    Case 3
        diaSemana = "Miércoles"
    Case 4
        diaSemana = "Jueves"
    Case 5
        diaSemana = "Viernes"
    Case 6
        diaSemana = "Sábado"
    Case 7
        diaSemana = "Domingo"
    Case Else
        diaSemana = "Número inválido"
End Select

lblResultado.Text = "Día de la semana: " & diaSemana
```

## Caso de estudio 3 Calificación de Desempeño en el Trabajo

Caso de estudio: Calificación de Desempeño en el Trabajo

### Objetivo:

Demostrar el uso de Select Case para responder a diferentes niveles de desempeño

### Descripción:

Un sistema evalúa el desempeño laboral de un empleado según una calificación del 1 al 5.

Desarrollo:

- Crear un TextBox para ingresar la calificación de desempeño (1-5).
- Usar Select Case para mostrar una evaluación según la calificación.
- Mostrar la evaluación en un Label.

Solución:

```
Dim calificacion As Integer = Val(txtCalificacion.Text)
Dim evaluacion As String
```

```
Select Case calificacion
```

```
    Case 5
```

```
        evaluacion = "Desempeño excelente"
```

```
    Case 4
```

```
        evaluacion = "Desempeño bueno"
```

```
    Case 3
```

```
        evaluacion = "Desempeño aceptable"
```

```
    Case 2
```

```
        evaluacion = "Desempeño bajo"
```

```
    Case 1
```

```
        evaluacion = "Desempeño muy bajo"
```

```
    Case Else
```

```
        evaluacion = "Calificación inválida"
```

```
End Select
```

```
lblResultado.Text = "Evaluación: " & evaluacion
```

## Caso de Estudio 4: Categoría de Edad

### Caso de Estudio 4: Categoría de Edad

**Objetivo:**

Practicar Select Case con rangos de valores para categorizar edades.

**Descripción:**

Un programa categoriza a una persona según su edad.

**Desarrollo:**

- Crear un TextBox para ingresar la edad.
- Usar Select Case para clasificar la edad en diferentes etapas de la vida.
- Mostrar la categoría de edad en un Label.

**Solución:**

```
Dim edad As Integer = Val(txtEdad.Text)
Dim categoria As String

Select Case edad
    Case 0 To 12
        categoria = "Niño"
    Case 13 To 17
        categoria = "Adolescente"
    Case 18 To 64
        categoria = "Adulto"
    Case Is >= 65
        categoria = "Adulto Mayor"
    Case Else
        categoria = "Edad inválida"
End Select

lblResultado.Text = "Categoría de edad: " & categoria
```

## 2. Buenas Prácticas en sentencias condicionales

Tener buenas prácticas es esencial a la hora de hacer uso de If conficionales y anidados, tener un orden correcto puede ayudar a la hora de realizar los ejercicios y soluciones.

### Pongamos en Práctica la Teoría

#### Caso de Estudio 1: Validación de Notas de Estudiantes

##### Caso de Estudio 1: Validación de Notas de Estudiantes

**Objetivo:**

Demostrar buenas prácticas en sentencias condicionales, como verificar condiciones de borde y limitar la cantidad de anidación.

**Descripción:**

Un sistema de calificaciones necesita verificar si una nota ingresada es válida antes de clasificarla.

**Desarrollo:**

- Crear un TextBox para ingresar la nota (0 a 100).
- Verificar si la nota está dentro del rango permitido (0-100) antes de clasificarla.
- Usar If...Elseif...Else para una estructura clara y directa en la clasificación.
- Mostrar la clasificación o un mensaje de error en un Label.

**Solución**

```
Dim nota As Integer = Val(txtNota.Text)
If nota < 0 Or nota > 100 Then
    lblResultado.Text = "Nota inválida. Debe estar entre 0 y 100."
Elseif nota >= 90 Then
    lblResultado.Text = "Clasificación: Excelente"
Elseif nota >= 80 Then
    lblResultado.Text = "Clasificación: Muy Bueno"
Elseif nota >= 70 Then
    lblResultado.Text = "Clasificación: Bueno"
Elseif nota >= 60 Then
    lblResultado.Text = "Clasificación: Suficiente"
Else
    lblResultado.Text = "Clasificación: Reprobado"
End If
```

## Caso de Estudio 2: Determinación de Etapas de Vida

### Caso de Estudio 1: Validación de Notas de Estudiantes

**Objetivo:**

Promover la verificación temprana de condiciones para mejorar la claridad y legibilidad.

**Descripción:**

Una aplicación clasifica a una persona en una etapa de vida según su edad. El programa debe validar primero que la edad ingresada sea válida.

**Desarrollo:**

- Crear un TextBox para ingresar la edad.
- Validar que la edad sea mayor o igual a 0 para proceder con la clasificación.
- Usar condicionales claros y evitar anidaciones innecesarias.

**Solución**

```
Dim edad As Integer = Val(txtEdad.Text)
If edad < 0 Then
    lblResultado.Text = "Edad inválida. Debe ser un número positivo."
Elseif edad <= 12 Then
    lblResultado.Text = "Categoría: Niño"
Elseif edad <= 17 Then
    lblResultado.Text = "Categoría: Adolescente"
Elseif edad <= 64 Then
    lblResultado.Text = "Categoría: Adulto"
Else
    lblResultado.Text = "Categoría: Adulto Mayor"
End If
```



## Caso de Estudio 3: Cálculo de Descuento en Compra

### Caso de Estudio 1: Validación de Notas de Estudiantes

**Objetivo:**

Promover la verificación temprana de condiciones para mejorar la claridad y legibilidad.

**Descripción:**

Una aplicación clasifica a una persona en una etapa de vida según su edad. El programa debe validar primero que la edad ingresada sea válida.

**Desarrollo:**

- Crear un TextBox para ingresar la edad.
- Validar que la edad sea mayor o igual a 0 para proceder con la clasificación.
- Usar condicionales claros y evitar anidaciones innecesarias.

**Solución**

```
Dim edad As Integer = Val(txtEdad.Text)
If edad < 0 Then
    lblResultado.Text = "Edad inválida. Debe ser un número positivo."
Elseif edad <= 12 Then
    lblResultado.Text = "Categoría: Niño"
Elseif edad <= 17 Then
    lblResultado.Text = "Categoría: Adolescente"
Elseif edad <= 64 Then
    lblResultado.Text = "Categoría: Adulto"
Else
    lblResultado.Text = "Categoría: Adulto Mayor"
End If
```

## Caso de Estudio 4: Determinación de Horarios según el Turno

### Caso de Estudio 4: Determinación de Horarios según el Turno

**Objetivo:**

Demostrar cómo estructurar condicionales para validar entradas y realizar asignaciones de forma clara y ordenada.

**Descripción:**

Una empresa clasifica sus horarios en tres turnos: mañana, tarde y noche. El programa muestra un mensaje de error si el turno es inválido.

**Desarrollo:**

- Crear un TextBox para ingresar el turno ("mañana", "tarde", "noche").
- Validar que el turno ingresado sea uno de los valores permitidos.
- Usar condicionales If...Elseif...Else para mostrar el horario correspondiente.

**Solución**

```
Dim turno As String = txtTurno.Text.ToLower()  
If turno = "mañana" Then  
    lblResultado.Text = "Horario: 8:00 a.m. - 12:00 p.m."  
Elseif turno = "tarde" Then  
    lblResultado.Text = "Horario: 1:00 p.m. - 5:00 p.m."  
Elseif turno = "noche" Then  
    lblResultado.Text = "Horario: 6:00 p.m. - 10:00 p.m."  
Else  
    lblResultado.Text = "Turno inválido. Ingrese 'mañana', 'tarde' o 'noche'.  
End If
```

**Conclusión**

Estos casos muestran cómo utilizar buenas prácticas en sentencias condicionales para mejorar la claridad, verificar condiciones de entrada de datos, y mantener el flujo lógico en el código.

Al implementar estas prácticas en el uso de Select Case y otras estructuras condicionales, el código se vuelve más robusto, confiable y fácil de entender, lo cual es fundamental en la formación de futuros ingenieros en ciencias y sistemas.