

Curso de Programación

Orientado a videojuegos

Empezamos en minutos...

Maestro Jedi: Daniel Delgado
Duración: al menos 10 horas.
El resto depende de ti...

Tipos de Decisiones

Las estructuras de decisión pueden ser de tres tipos:

Decisión Simple: consiste en decidir si se ejecuta un conjunto de instrucciones o se omiten. En este caso en el programa queda expresado qué debe hacer la computadora si la condición es verdadera solamente. Si es falso simplemente se sigue con las instrucciones que están después del bloque o de estructura de decisión.

Un ejemplo de esta decisión en programación sería tomar la vida de nuestro personaje, si la variable vida == 0 entonces acabar el juego, (o ir al punto de respawn). Si todavía es mayor de 0, pues no se ejecutan las instrucciones de gameover o respawn y sigue el juego.

Decisión doble: se presenta cuando se tienen dos alternativas claras de ejecución y dependiendo del resultado de la condición se ejecuta, primero la que es verdadera, y si es falsa se ejecuta la otra.

Un ejemplo de esta decisión doble es por ejemplo en cualquier juego de plataformas, preguntamos ¿está el personaje en el piso? **Si está** en el piso, entonces habilitar las funciones de caminar, correr, brincar, saltar, disparar, y **si no está** en el piso, entonces habilitar animación de caída, y poder flotar con la colita... (it's a me, Mario...)

Decisión Multiple: es la estructura que programamos al evaluar una variable y dependiendo de su valor se ejecutan una serie de instrucciones. En este caso, el conjunto de valores debe ser mayor de dos, y solo se evalúa la condición de igualdad. Cada posible valor está asociado con una secuencia de instrucciones, y son mutuamente excluyentes.

Como ejemplo de esto, tenemos un menú en un juego. La opción 1 es para jugar el modo campaña, la opción 2 es para jugar el modo Multijugador, la 3 para Configuración del juego, la 4 para Ayuda, y la opción 5 para Salir del juego. Como se ve cada opción es excluyente de la otra, y sólo se evaluaría si el usuario escogió entre la opción 1 y 5... no puede seleccionar dos opciones al mismo tiempo.

La Sentencia If-Then

Esta instrucción evalúa un valor lógico, una expresión de relación o expresión lógica y retorna un valor lógico, solo un verdadero o falso, así toma la decisión de ejecutar el siguiente bloque de instrucciones o lo ignora.

Trabaja con decisiones simples o dobles. Esta es su estructura en Pseudocódigo.

```
IF <condición> THEN
    Instrucciones
EndIF
```

Decisiones Simples:

Ejemplo Cálculo del valor absoluto de un número. El valor absoluto de un número es la distancia que hay entre el 0 y ese número y cómo una distancia no debe ser negativa, así el número sea positivo o negativo, el valor absoluto del número será el mismo número sin el signo.

```
| 4 | = 4
| -3 | = 3
```

Los datos de entrada: número

Proceso: verificar que el número sea negativo, entonces multiplicar por -1 para cambiar el signo, caso contrario escribir el mismo número.

Los datos de salida: valor absoluto.

```
Begin AbsoluteValue
```

```
    Var Number: number;
```

```
    Input “Escriba un número para
    calcular su valor absoluto”, number;
```

```
    IF number < 0 THEN
```

```
        number = number * -1;
```

```
    EndIF
```

```
    Print “El valor absoluto es: ” &
    number;
```

```
End AbsoluteValue
```

Otro ejemplo...

Este ejemplo es uno de los más usados en clases de programación.

Detectemos si un número es par. Un número par es un número que es divisible exactamente entre 2.

Datos de entrada: el número a evaluar.

Datos de salida: Escribir en pantalla el mensaje “Es par” solo si el número es par.

Proceso: Número MOD 2 (Recuerda esto es el resto de la división, si la división es exacta el MOD va a ser igual a 0)

```
Begin EvenNumber
    Var Number: number;
    Input “Introduzca un número para
saber si es par”, number;
    IF number MOD 2 == 0 THEN
        Print “El número es par”;
    EndIF
End EvenNumber
```

Decisión doble con IF - THEN - ELSE

Se debe implementar una decisión doble cuando se cuenta con dos opciones para continuar la ejecución del programa, y éstas dependen de una condición; para decirlo más fácil se necesita ejecutar una serie de instrucciones si la condición es verdadera y otra serie de instrucciones si la condición es falsa.

Mientras las decisiones simples se usan para incluir saltos en la ejecución del programa, las decisiones dobles permiten hacer bifurcaciones. Así se presenta en pseudocódigo:

```
IF <condición> THEN
    Instrucciones
ELSE
    Instrucciones
EndIF
```

En el caso del algoritmo anterior, de los pares, nos faltó el caso de mostrar entonces que el número era impar.

Usaremos el mismo código y añadiremos el ELSE en la sentencia para mostrar el mensaje cuando corresponda:

```
Begin EvenNumber
    Var Number: number;
    Input "Introduzca un número para saber si es par",
    number;
    IF number MOD 2 == 0 THEN
        Print "El número es par";
    ELSE
        Print "El número es impar";
    EndIF
End EvenNumber
```

Otro ejemplo antes de la tormenta... digo antes de los ejercicios...

La división de dos números se puede hacer siempre y cuando el divisor sea distinto de 0.

Hacer el programa que calcule la división de dos números positivos, haciendo los dos cálculos posibles (Numero1 / Numero2 y Numero2 / Numero1) y alertar al usuario en caso de que el divisor sea cero.

Datos de Entrada: dos números positivos

Datos de salida: solución de las dos divisiones posibles, o mensaje de error en caso de que el divisor sea cero

Proceso: $B > 0$ entonces mostrar resultado de A / B
 $A > 0$ entonces mostrar resultado de B / A

```
Begin BidirectionalDivision
  Var Number: num1, num2;
  Input "Introduzca dos números", num1, num2;
  IF num2 > 0 THEN
    Print num1 & " entre " & num2 & " = " & num1 /
num2;
  ELSE
    Print "Divisor " & num2 & " es igual a 0 o es
negativo";
  EndIF
  IF num1 > 0 THEN
    Print num2 & " entre " & num1 & " = " & num2
/ num1;
  ELSE
    Print "Divisor " & num1 & " es igual a 0 o es
negativo";
  EndIF
End BidirectionalDivision
```