

## **INTRODUCCIÓN A LA PROGRAMACIÓN**

### **NOTAS DE PROFUNDIZACIÓN**

### **SENTENCIA MULTIPLE**

#### **¡BIENVENIDOS!**

A continuación, profundizaremos en lo relativo a las estructuras condicionales, las cuales comparan una variable o constante con otras que formen parte de la estructura en la que se esté planteando, para posteriormente, de acuerdo con el resultado de esa comparación, seguir las instrucciones específicas que estén dentro del algoritmo. Conviene señalar que existen diferentes tipos de condicionales, cada uno con características, aplicaciones y funcionalidades distintas, que se usan de acuerdo a la naturaleza del problema o situación que se presenta.

En el uso de las estructuras algorítmicas condicionales, se deben establecer los diagramas de flujos y los pseudocódigos que permitan evidenciar de forma organizada, cada uno de los pasos que conlleven a la solución lógica de la situación planteada.

#### **COMENCEMOS**

##### **Antecedentes relevantes**

La descripción de flujo de datos y procesos ayuda a documentar el análisis de un sistema de información a través de distintas herramientas como diagramas de flujo y pseudocódigos. Estos muestran de forma descriptiva y precisa la estructura de un programa antes de ser codificado.

Los diagramas de flujo de datos DFD indican de forma gráfica el funcionamiento de un programa. Por su parte, el pseudocódigo señala de manera narrativa el movimiento de los datos.

Como se observó en semanas anteriores, existen diferentes estructuras que muestran un orden lógico y organizado para la resolución de problemas. El contenido de esta semana está enfocado en la estructura que permite obtener un resultado mediante una comparación de datos. En otras palabras, las estructuras de datos, también se le conocen como selectivas, ya que permiten realizar comparaciones con una o varias variables y constantes.

En relación a lo anterior revisa la siguiente información:

Las Estructuras de control van a permitir poder establecer condiciones o control donde así sea solicitado por los requerimientos del sistema.
Por medio de las estructuras se podrán construir condiciones para la toma de decisiones.
Las estructuras de control permiten ordenar el sistema.

Tabla 1: Estructuras de Control.

Fuente: Elaboración Propia

## ESTRUCTURAS ALGORITMICAS CONDICIONALES

Las estructuras de control condicionales son muy importantes en programación ya que permiten tomar decisiones en tiempo de ejecución en base a ciertas condiciones. Permite al programa ejecutar un bloque de código si se cumple una determinada condición y otro bloque de código si no se cumple. Esto es esencial para crear programas que se comporten de manera adaptativa y puedan manejar diferentes situaciones.

Además, también permite crear programas más legibles y fáciles de entender, ya que el código está organizado de manera lógica y estructurada. También permite crear programas más seguros ya que se pueden validar las entradas del usuario y evitar posibles errores.

La estructura condicional es una herramienta esencial para la toma de decisiones en programación, lo que permite crear programas más flexibles, legibles y seguros. La sintaxis básica es la siguiente:

```
si (condición) {  
    // código a ejecutar si se cumple la condición  
    si (condición) {  
        // código a ejecutar si se cumple la condición  
    }  
    sino {  
        // código a ejecutar si no se cumple la condición  
    }  
}
```

```
Algoritmo Ejemplo_si_hacer_si_sino  
//Definición de las variables  
Definir num1, num2, num3 como real  
//solicitar datos al usuario  
escribir "Ingrese el primer número: "  
leer num1  
escribir "Ingrese el segundo número: "  
leer num2  
escribir "Ingrese el tercer número: "  
leer num3  
//comparar los números  
si (num1 > num2) entonces  
    si (num1 > num3) entonces  
        escribir "Mayor es:", num1  
    sino  
        escribir "Mayor es", num3  
    fin si  
sino  
    si (num2 > num3) entonces  
        escribir "Mayor es:", num2  
    sino  
        escribir "Mayor es: ", num3  
    fin si  
fin si  
FinAlgoritmo
```

Imagen 1: Sentencia Múltiple

Fuente: Elaboración propia

Para conocer más:

- La sintaxis Sentencia Múltiple: SI – HACER SI – SINO
- Ejemplo de un Pseudocódigo donde se desea saber cuál es el número mayor entre 3 números. En este caso, se solicita al usuario que ingrese tres números, luego utiliza la estructura "SI-HACER SI-SINO" para comparar los números y determinar cuál es el mayor, y finalmente lo imprime en pantalla.

## SENTENCIA MÚLTIPLE: SI – HACER SI – SINO

La estructura algorítmica "SI-HACER SI-SINO" es posiblemente la más utilizada, es muy útil cuando se necesita ejecutar una o más instrucciones en caso de que una condición se dé, y ejecutar una o más instrucciones diferentes en caso contrario. También es conocida como estructura condicional de selección doble, debido a que "ejecuta un bloque de instrucciones cuando la proposición (condición) es verdadera y un bloque diferente cuando ésta es falsa." López (2009, p. 59).

Se mostrará un ejemplo de un diagrama de flujo de datos y un pseudocódigo donde se desea saber si una persona es mayor de edad o menor de edad. En este caso, la condición a evaluar consta principalmente de verificar si la variable edad es mayor o igual a 18 años (tomando 18 años como mayoría de edad). Dado esto, la condición a evaluar sería "edad >= 18 años", esta condición permitirá mostrar un mensaje indicando que la persona es mayor de edad si esa condición se cumple, de lo contrario mostrará un mensaje indicando lo contrario.

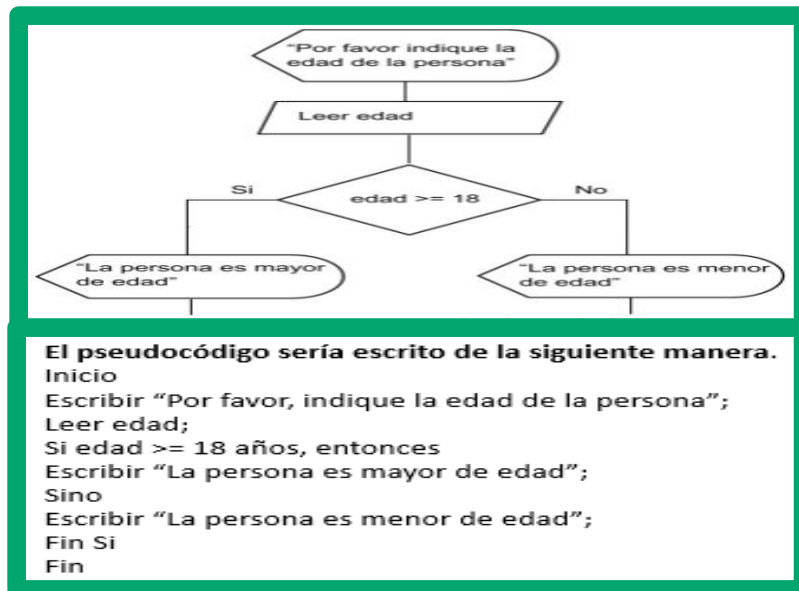


Imagen 2: Ejemplo De un Diagrama de Flujo y Pseudocódigo

Fuente: Elaboración propia

Para conocer más:

- Se le solicita al usuario que ingrese la Edad, quiere decir que en este punto es la Entrada de Datos.
- En el proceso existe una toma de decisión o también llamado condicional.
- Dependiendo de lo evaluado en el condicional o toma de decisión se obtendrá el mensaje o resultado.
- Se muestra el ejemplo, pero en Pseudocódigo donde se evalúa si la edad de la persona es mayor o menor de edad y se muestra con un mensaje.

## CASO MODELO

La estructura de control Caso, al igual que las estructuras anteriores, permite ejecutar una acción o determinada operación, pero presenta más de dos (2) alternativas, pudiendo ejecutar instrucciones si una variable toma algún valor en particular. Para comprender mejor esta definición veamos el siguiente ejemplo:

Se requiere mostrar un mensaje indicando el día de la semana de acuerdo con el número ingresado por el usuario. Si el número es el 1 debe mostrar el mensaje “Lunes”, si el numero ingresado es 2 debe mostrar el mensaje “Martes”, y así sucesivamente hasta el número 7 que debe mostrar el mensaje “Domingo”. Por lo tanto, de ingresar un valor fuera de ese rango se mostrará un mensaje indicando “El número ingresado no corresponde a ningún día”.

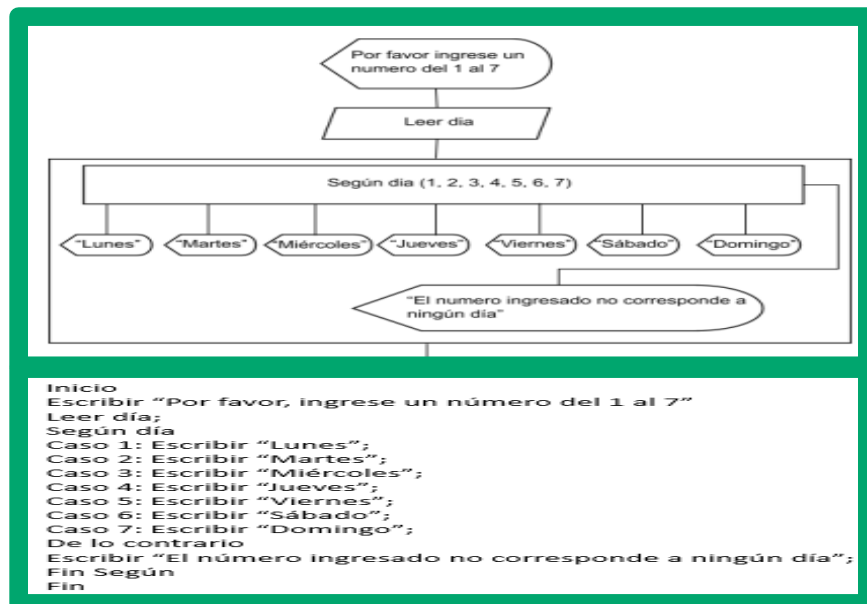


Imagen 3: Caso modelo

Fuente: Elaboración propia

Para conocer más:

- Se realiza la entrada de datos ingresando el día de la semana representado del 1 al 7 como días de la semana.
- En la estructura de control según sea el caso, se indicará la toma de decisión, también conocida como toma de decisión múltiple.
- Cuando se realiza este tipo de estructura hay dos tipos de salida, cuando la condición se cumple indicara el día de la semana, pero en caso que no se cumpla existe una salida de mensaje de error, esta salida de error es realizada por el programador este tipo de estructura se recomienda para realizar menú de opciones en la construcción de un programa.
- Como se muestra en la figura así quedaría el pseudocódigo de la estructura de control de caso.

## CASO MODELO (EJEMPLO PRACTICO)

Desarrollar un algoritmo que permita llevar el control de venta de una Mini Tienda, donde se requiere llevar el control de la venta y que se le entregue una factura al cliente indicando sus datos como Rut y el Nombre del Cliente, la cantidad de productos y el importe o total a pagar por parte del cliente.

El programa debe tener un menú con cuatro opciones:

1. Ingresar los datos del cliente.
2. Ingresar los datos de la compra.
3. Generar factura de la compra.
4. Salir del programa.

El usuario debe seleccionar una opción ingresando el número correspondiente. Si el usuario selecciona la opción 1, se le preguntará si el cliente es nuevo y, en caso afirmativo, se le pedirá que ingrese los datos del cliente (nombre y RUT). Y en caso contrario continua con el cliente anterior. Si el usuario selecciona la opción 2, se le pedirá que ingrese el nombre, precio y cantidad de los productos que está comprando.

El programa calculará el costo parcial del producto y lo sumará al costo total de la compra. También llevara un sumador para saber la cantidad de productos que el cliente compra. Si el usuario selecciona la opción 3, se generará una factura con los datos del cliente y el costo total de la compra y la cantidad de artículos comprados. Si el usuario selecciona la opción 4, el programa finalizará. Si el usuario ingresa una opción que no se encuentre entre la opción 1 hasta la 4, el programa mostrará un mensaje de error. El programa seguirá ejecutándose hasta que el usuario seleccione la opción de finalizar (opción 4).

Para conocer más:

- A continuación, se muestra un algoritmo, donde se puede apreciar la estructura repetir hasta, que va a permitir que el programa se ejecute hasta que el usuario lo indique por medio de la opción 4 para este caso, luego se muestra la estructura de caso (Según opc hacer) esta estructura permite en el programa realizar un menú de opciones como se muestra en el algoritmo.
- A continuación se muestra la segunda parte del algoritmo donde están el resto de las opciones.

<pre>Algoritmo Menu_opciones Repetir Escribir "Menú de Opciones" Escribir "1. Ingresar los datos del cliente" Escribir "2. Ingresar los datos de la compra" Escribir "3. Generar factura de la compra" Escribir "4. Finalizar" Escribir "Seleccione la opción:" Leer opc Según opc Hacer 1: Escribir " Es Cliente Nuevo S/N" Leer clie si clie ="s" o clie="S" Entonces Escribir "Ingresar los Datos del Cliente" Escribir "Ingrese el Nombre Cliente" Leer Nomb_Cliente Escribir "Ingrese el Rut del Cliente" Leer Rut_Cliente FinSi</pre>	<pre>Algoritmo Menu_opciones Repetir Escribir "Menú de Opciones" Escribir "1. Ingresar los datos del cliente" Escribir "2. Ingresar los datos de la compra" Escribir "3. Generar factura de la compra" Escribir "4. Finalizar" Escribir "Seleccione la opción:" Leer opc Según opc Hacer 1: Escribir " Es Cliente Nuevo S/N" Leer clie si clie ="s" o clie="S" Entonces Escribir "Ingresar los Datos del Cliente" Escribir "Ingrese el Nombre Cliente" Leer Nomb_Cliente Escribir "Ingrese el Rut del Cliente" Leer Rut_Cliente FinSi</pre>
---	---

Imagen 4: Ejemplo práctico, Estructura de Control de Caso.

Fuente: Elaboración propia

## AVERIGUA MÁS

### TE RECOMENDAMOS

Para profundizar más en el tema planteado se le recomienda realizar consultas a las siguientes referencias bibliográficas:

Duque, D., Saint-Priest, Y., Segovia, P., Loaiza, D. (2017). *Tipos de Datos y Expresiones- Algoritmos y programación en pseudocódigo*. Editorial Universidad. Santiago de Cali.

López J. (2009). *Algoritmos y Programación*. Guía para docentes.

Peralta, R. & Sánchez, G. *Algoritmos*. f <https://docplayer.es/28311683-Algoritmos-intro-prof-raquel-torres-peralta-gerardo-sanchez-s-unison.html>

Rancel, M. (2017). *La Algoritmización: requisito necesario para la solución de problemas con el empleo de un lenguaje de programación.* f  
<https://www.redalyc.org/journal/5891/589167671004/html/>