



# Laboratorio 5

Darlene Larissa Méndez Bedoya 1142526

Repositorio: <https://github.com/Larissa-M008/Laboratorio-PenCom.git>

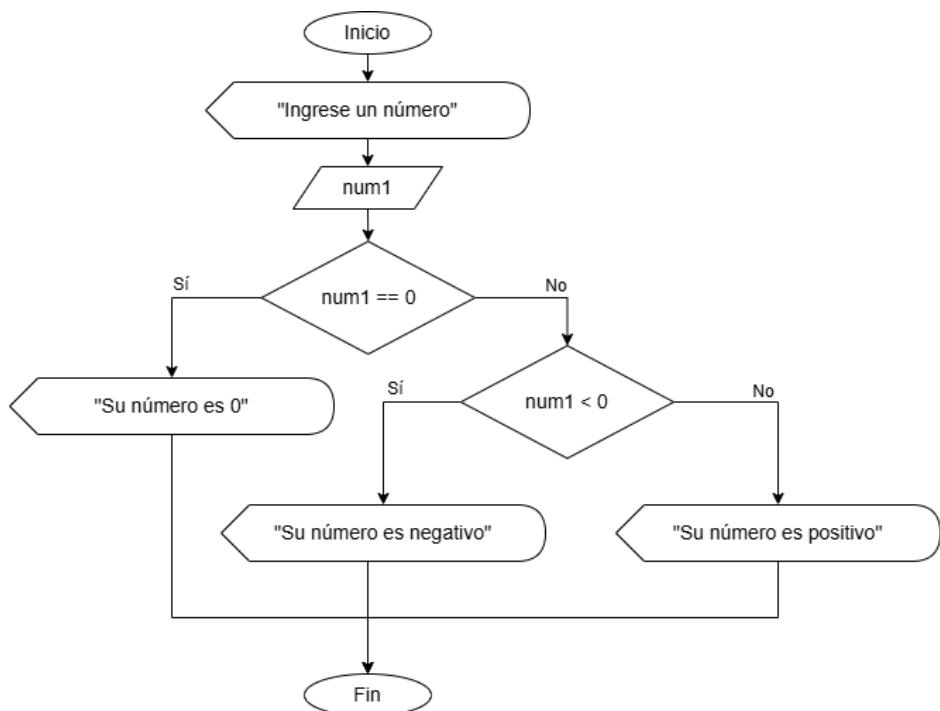
Desafío # 1: Estructuras Selectivas (Else if) (15pts).

Realizar el diagrama de flujo y el código en C# de un programa que solicite al usuario ingresar un número entero e indique en pantalla si se trata de un número positivo, negativo o cero. No olvide realizar el análisis de entradas, procesos y salidas.

Código:

```
Console.WriteLine("Larissa Méndez Bedoya_1142526");
Console.WriteLine("Laboratorio 5 PC");
Console.WriteLine("");
Console.WriteLine("Desafío 1");
Console.WriteLine("Ingrese un número");
string num1 = Console.ReadLine();
int num2 = int.Parse(num1);
if (num2 == 0) {
    Console.WriteLine("Su numero es cero.");
} else if (num2 < 0) {
    Console.WriteLine("Su número es negativo");
} else {
    Console.WriteLine("Su número es positivo");
}
```

Diagrama de flujo:



**Entradas:** Donde se ingresa el número a evaluar.

**Procesos:** Las condiciones If, else if y else, donde se clasifica el número si es igual, menor o mayor a 0.

**Salidas:** Donde muestran el mensaje de si el número es cero, positivo o negativo.

### Desafío # 2: Estructuras Selectivas (Else if) (15pts).

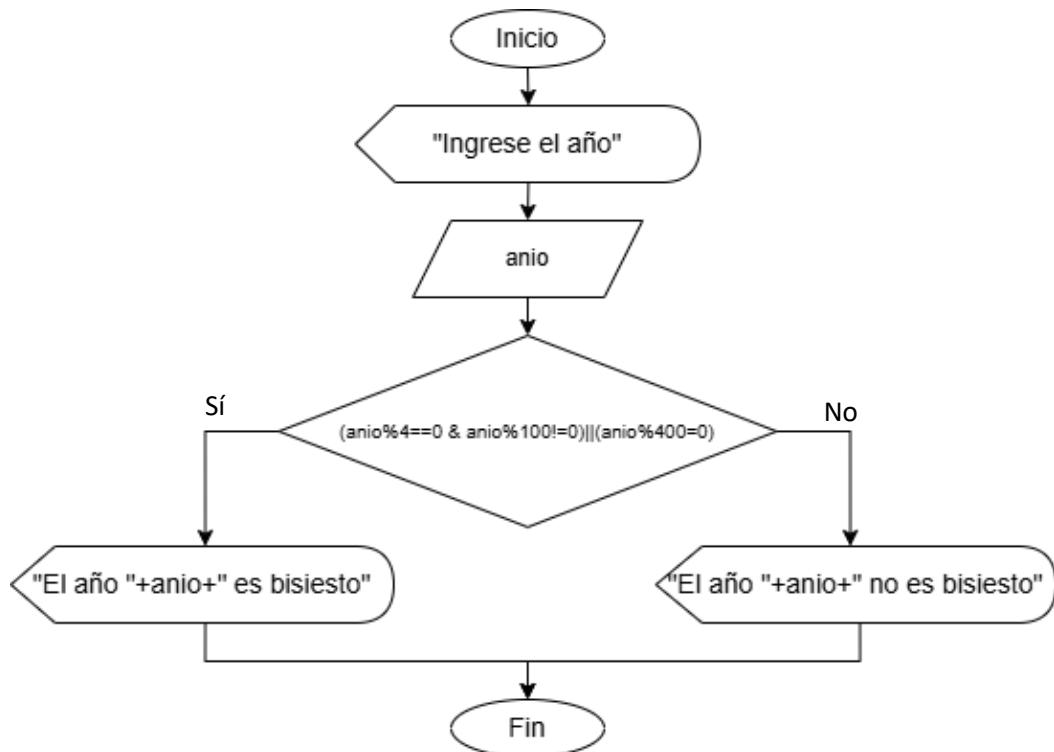
Un año bisiesto es aquel donde el mes de febrero contiene un día adicional con el objetivo de equilibrar el año cronológico con el solar. La norma indica que un año es bisiesto cuando puede dividirse exactamente entre 4 pero no es divisible entre 100. Con la excepción de los años que son divisibles entre 400.

Realizar el diagrama de flujo y código en C# de un programa que solicite al usuario ingresar el año (número entero) e indique en pantalla si se trata de un año bisiesto o no. No olvide realizar el análisis de entradas, procesos y salidas.

**Código:**

```
Console.WriteLine("");
Console.WriteLine("Desafío 2");
Console.WriteLine("Ingrese el número del año");
string anio = Console.ReadLine();
int anio1 = int.Parse(anio);
if ((anio1 % 4 == 0 & anio1 % 100 != 0) || (anio1 % 400 == 0))
{
    Console.WriteLine("El año " + anio + " es bisiesto");
}
else
{
    Console.WriteLine("El año " + anio + " no es bisiesto");
}
```

**Diagrama de flujo:**



**Entradas:** Donde se ingresa el año a evaluar.

**Procesos:** El if que clasifica si el año es bisiesto o no.

**Salidas:** Los textos que muestra si el año es bisiesto o no.

### Desafío #3: Estructuras Selectivas (Else if) (20pts).

El boleto de ornato es un tributo anual obligatorio por parte de los vecinos, destinado a financiar proyectos y actividades que contribuyen a mejorar el aspecto estético y funcional de la Ciudad de Guatemala. La contribución de cada vecino se determina en base a su ingreso mensual, puede utilizar la siguiente tabla como referencia:

Ingreso Mensual	Arbitrio	Con Multa
De Q 500.01 a 1,000.00	Q 10.00	Q 20.00
De Q 1,000.01 a 3,000.00	Q 15.00	Q 30.00
De Q 3,000.01 a 6,000.00	Q 50.00	Q 100.00
De Q 6,000.01 a 9,000.00	Q 75.00	Q 150.00
De Q 9,000.01 a 12,000.00	Q 100.00	Q 200.00
De Q 12,000.01 en adelante	Q 150.00	Q 300.00

Realizar el diagrama de flujo y código en C# de un programa que tenga como entrada el ingreso mensual y una variable Booleana llamada “multa” y muestre al usuario la cantidad que contribuirá al ornato de la ciudad. No olvide realizar el análisis de entradas, procesos y salidas.

#### Código:

```
Console.WriteLine("");
Console.WriteLine("Desafío 3");
Console.WriteLine("Ingrese su salario: ");
string salario = Console.ReadLine();
double salario1 = double.Parse(salario);
Console.WriteLine("Si tiene multa Si=0/No=1");
string multa1 = Console.ReadLine();
int multa2 = int.Parse(multa1);
RespuestaMulta = (multa2 == 0);
if ((salario1 >= 500.01 & salario1 <= 1000.00)) {
    ValorBoleto = 10;
    if (RespuestaMulta)
    {
        ValorBoleto = ValorBoleto * 2;
    }
    Console.WriteLine("Debe pagar:" + ValorBoleto + " Si multa es " + RespuestaMulta);
}
else if ((salario1 >= 1000.01 & salario1 <= 3000.00))
{ ValorBoleto = 15;
    if (RespuestaMulta)
    {
        ValorBoleto = ValorBoleto * 2;
    }
    Console.WriteLine("Debe pagar:" + ValorBoleto + " Si multa es " + RespuestaMulta);
}
else if (salario1 >= 3000.01 & salario1 <= 6000.00)
{
    ValorBoleto = 50;
    if (RespuestaMulta)
    {
        ValorBoleto = ValorBoleto * 2;
    }
    Console.WriteLine("Debe pagar:" + ValorBoleto + " Si multa es " + RespuestaMulta);
}
else if (salario1 >= 6000.01 & salario1 <= 9000.00)
{
    ValorBoleto = 75;
    if (RespuestaMulta)
    {
        ValorBoleto = ValorBoleto * 2;
    }
    Console.WriteLine("Debe pagar:" + ValorBoleto + " Si multa es " + RespuestaMulta);
}
else if (salario1 >= 9000.01 & salario1 <= 12000.00)
{
    ValorBoleto= 100;
    if (RespuestaMulta)
    {
        ValorBoleto = ValorBoleto * 2;
    }
    Console.WriteLine("Debe pagar:" + ValorBoleto + " Si multa es " + RespuestaMulta);
}
```

```
else
{
    ValorBoleto = 150;
    if (RespuestaMultas)
    {
        ValorBoleto = ValorBoleto * 2;
    }
    Console.WriteLine("Debe pagar:" + ValorBoleto + " Si multa es " + RespuestaMultas);
}
```

**Entradas:** Donde se ingresa el salario y si la multa es Sí=0 y No=1

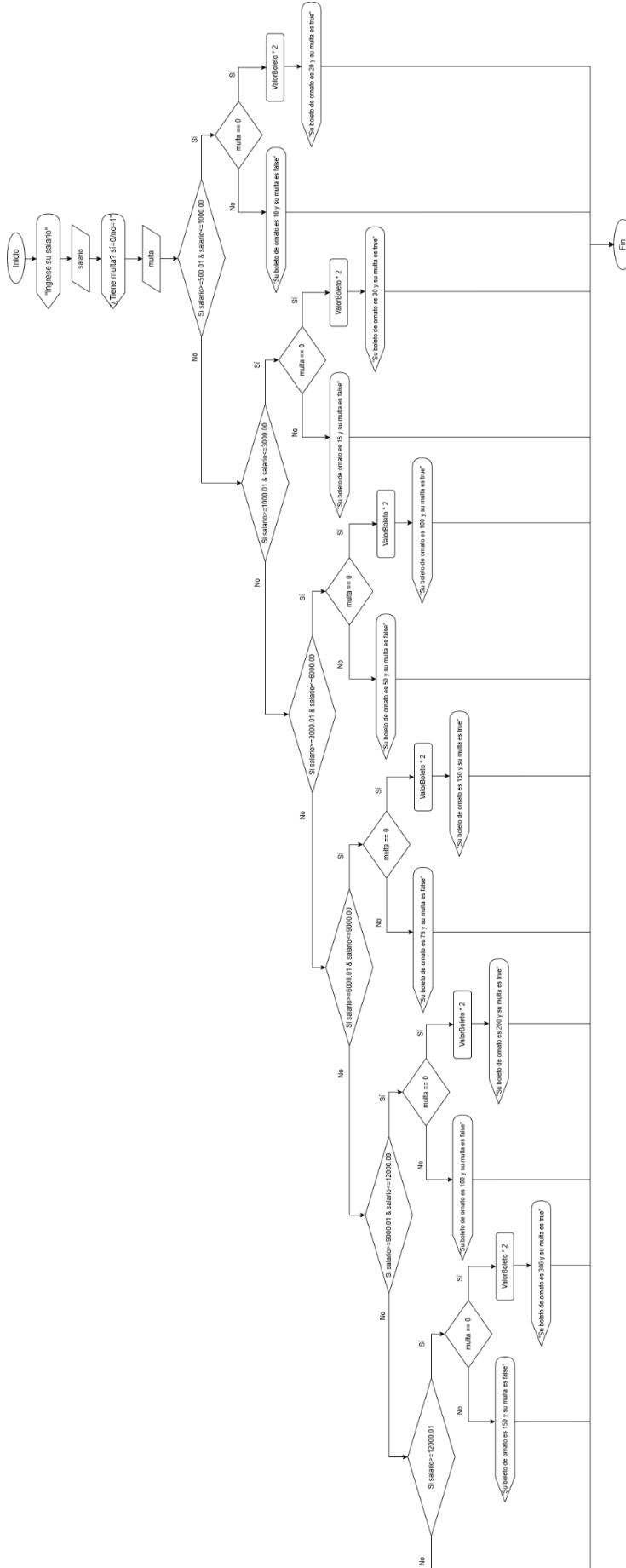
**Procesos:** Cada if que procesa si el salario es correspondiente a cada rango de salario y si tenía multa debía multiplicarlo por 2.

**Salidas:** Cada vez que mostraba el valor del boleto de ornato en pantalla, si tenía multa o no.

**Resultado de como se ve en pantalla:**

```
Desafío 3
Ingrese su salario:
3400.05
Si tiene multa Sí=0/No=1
0
Debe pagar:100 Si multa es True
```

## Diagrama de flujo:



Para poder visualizar mejor el diagrama de flujo:

<https://drive.google.com/file/d/1Ok1Spn51L1WU2ncBE4oklhs-AV5yQkl5/view?usp=sharing>

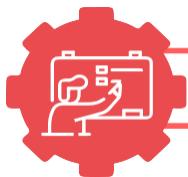
#### Desafío #4: Estructuras selectivas & Operadores aritméticos (50pts)

Desarrollar un programa en C# que simule una máquina de pago de parqueo en un centro comercial. La máquina inicia solicitando al usuario la cantidad de horas (número entero) que estuvo estacionado y calculará el monto a pagar asumiendo que el costo por hora es de Q10.

Luego el usuario ingresará el monto con el que realizará el pago. Debido a que esta máquina solamente acepta billetes, puede asumir que el pago será un número entero.

- Si el monto ingresado es menor al total a pagar, el programa deberá mostrar un mensaje de error y finalizar.
- si el monto ingresado es exactamente igual a la cantidad que debe pagar se mostrará un mensaje indicando que no se requiere cambio y finalizará el programa.
- Por último, si el monto ingresado es mayor al cobro del parqueo el programa calculará el vuelto que se debe entregar y mostrará la cantidad de cada billete entregado en el vuelto.

Con la finalidad de utilizar la menor cantidad posible de billetes se priorizará la entrega de billetes de mayor denominación. La máquina cuenta con disponibilidad de billetes de Q1, Q5, Q10, Q20, Q50 y Q100.



#### Pista

La relación entre las denominaciones de billetes es similar a la conversión entre horas, minutos y segundos. Puedes apoyarte de los operadores **DIV** y **MOD** para realizar los cálculos.

Una posible estrategia es comenzar calculando cuántos billetes de Q100 caben en el monto ingresado. Luego, con el residuo obtenido, puedes repetir el proceso usando la siguiente denominación más pequeña. Continúa sucesivamente hasta completar la descomposición del monto.

#### Código

```
Console.WriteLine("");
Console.WriteLine("Desafío 4");
Console.WriteLine("Ingrese cuántas horas estuvo estacionado: ");
string horas1 = Console.ReadLine();
int horas = int.Parse(horas1);
int horasTotales = horas * 10;
Console.WriteLine("Su total es de: "+horasTotales);
Console.WriteLine("Ingrese el total a pagar: ");
string pago1 = Console.ReadLine();
int pago = int.Parse(pago1);
int vuelto = pago - horasTotales;
if (pago < horasTotales)
{
    Console.WriteLine("Error, los fondos no son suficientes");
}
else if (pago == horasTotales)
{
    Console.WriteLine("No se requiere cambio ¡Feliz día!");
}
```

```

else
{
    Console.WriteLine("Su vuelto es de: "+vuelto);
    int billetes = vuelto / 100;
    Console.WriteLine("Billetes de Q100: " + billetes);
    vuelto = vuelto - (billetes*100);
    billetes = vuelto/ 50;
    Console.WriteLine("Billetes de Q50: " + billetes);
    vuelto = vuelto - (billetes*50);
    billetes = vuelto / 20;
    Console.WriteLine("Billetes de Q20: " + billetes);
    vuelto = vuelto - (billetes*20);
    billetes = vuelto / 10;
    Console.WriteLine("Billetes de Q10: " + billetes);
    vuelto = vuelto - (billetes*10);
    billetes = vuelto / 5;
    Console.WriteLine("Billetes de Q5: " + billetes);
    vuelto = vuelto - (billetes*5);
    billetes = vuelto / 1;
    Console.WriteLine("Billetes de Q1: " + billetes);
}

```

Como se ve en pantalla:

```

Desafío 4
Ingrese cuántas horas estuvo estacionado:
10
Su total es de: 100
Ingrese el total a pagar:
500
Su vuelto es de: 400
Billetes de Q100: 4
Billetes de Q50: 0
Billetes de Q20: 0
Billetes de Q10: 0
Billetes de Q5: 0
Billetes de Q1: 0

```

**Entradas:** Donde se ingresa las horas que se estuvo estacionado y el monto a cancelar

**Procesos:** La primera es donde se hace la conversión de horas\*costo. El if, else if y else, donde se filtra si el monto a pagar necesita vuelto, si es menor a lo que se debe pagar o cuanto es lo que se debe dar de vuelto. Además, el bloque en else calcula cuántos billetes de cada uno se debe de dar de vuelto.

**Salidas:** La primera es donde se muestra la conversión de horas\*costo, luego del vuelto y por último la tabla de cuántos billetes de cada uno se debe de dar como vuelto.



## Ejemplo

Datos de Entrada	Mostrar en pantalla
<b>Horas estacionado: 6</b> <b>Total a pagar: Q60</b> <b>Monto ingresado: Q25</b>	"Error, los fondos no son suficientes".
<b>Horas estacionado: 4</b> <b>Total a pagar: Q40</b> <b>Monto ingresado: Q40</b>	"No se requiere cambio, ¡Feliz día!"
<b>Horas estacionado: 8</b> <b>Total a pagar: Q80</b> <b>Monto ingresado: Q100</b>	Cambio → Q20 Billetes de Q100: 0 Billetes de Q50: 0 Billetes de Q20: 1 Billetes de Q10: 0 Billetes de Q5: 0 Billetes de Q1: 0
<b>Horas estacionado: 12</b> <b>Total a pagar: Q120</b> <b>Monto ingresado: Q200</b>	Cambio → Q80 Billetes de Q100: 0 Billetes de Q50: 1 Billetes de Q20: 1 Billetes de Q10: 1 Billetes de Q5: 0 Billetes de Q1: 0
<b>Horas estacionado: 9</b> <b>Total a pagar: Q90</b> <b>Monto ingresado: Q194</b>	Cambio → Q104 Billetes de Q100: 1 Billetes de Q50: 0 Billetes de Q20: 0 Billetes de Q10: 0 Billetes de Q5: 0 Billetes de Q1: 4

	<b>Críterio</b>	<b>Puntaje</b>
Desafío #1 15pts	Identifica correctamente Entrada, Proceso y Salida.	2 pts
	Diagrama de flujo claro y correcto	4 pts
	Identificadores nombrados correctamente en el código	2 pts
	Uso correcto de estructuras selectivas (else-if)	2 pts
	Sintaxis correcta, permitiendo compilar la solución	2 pts
	El programa en C# resuelve correctamente el problema	3 pts
Desafío #2 15pts	Identifica correctamente Entrada, Proceso y Salida.	2 pts
	Diagrama de flujo claro y correcto	4 pts
	Identificadores nombrados correctamente en el código	2 pts
	Uso correcto de estructuras selectivas (else-if)	2 pts
	Sintaxis correcta, permitiendo compilar la solución	2 pts
	El programa en C# resuelve correctamente el problema	3 pts
Desafío #3 20pts	Identifica correctamente Entrada, Proceso y Salida.	2 pts
	Diagrama de flujo claro y correcto	2 pts
	Identificadores nombrados correctamente en el código	3 pts
	Uso correcto de estructuras selectivas (else-if)	3 pts
	Sintaxis correcta, permitiendo compilar la solución	3 pts
	El programa en C# resuelve correctamente el problema	5 pts
Desafío #4 50 pts	Identificadores nombrados correctamente en el código	10 pts
	Uso correcto de estructuras selectivas (else-if)	10 pts
	Sintaxis correcta, permitiendo compilar la solución	10 pts
	El programa en C# resuelve correctamente el problema	20 pts
<b>Total</b>		<b>100 pts</b>