

# Programacion estructurada.

Daniel Reyes Barrera

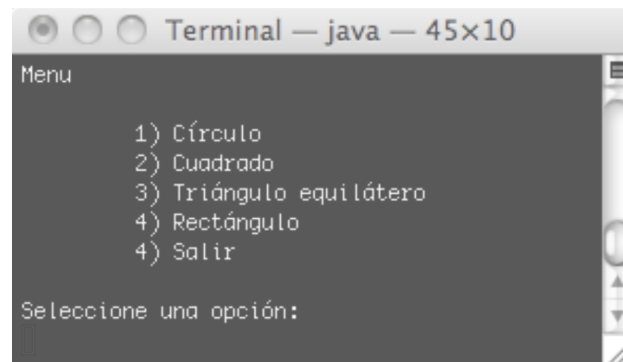
23 de noviembre de 2020

## Resumen

En este documento se han resuelto algunos problemas computacionales utilizando las herramientas aprendidas en la clase 5 – Programación estructurada del curso de programación C++, como son Modularización, métodos y funciones las cuales nos proporcionan un mejor uso para un bloque recurrente de instrucciones.

## 1. Ejercicio 1.

Escriba un programa que imprima un menú para seleccionar un tipo de figura geométrica de la siguiente forma:



El usuario debe seleccionar una opción y el programa debe calcular el área y perímetro de la opción seleccionada. Programe cada opción en un método independiente. El programa debe regresar al menu principal hasta que el usuario seleccione la opción salir.

### 1.1. Problema computacional.

**Objetivo:** Dado un número entero como año determinar si es un año bisiesto o no.

**Entrada:** Un número entero mayor que 0 representando un año.

**Salida:** La respuesta de si el número dado es un año bisiesto o no.

### 1.2. Algoritmo.

Para solucionar el problema computacional, partimos de la definición de un año bisiesto. Un año se denomina bisiesto si es múltiplo de 4, a excepción de que sí es múltiplo de 100 también y no de 400 entonces no es bisiesto, por tanto necesitaremos utilizar el condicional **if-else** para validar que se cumplan las condiciones para que el año sea bisiesto o no.

El código fuente está disponible en mi repositorio de git hub. [1]

### 1.3. Instancia del problema.

Como prueba de escritorio, se seleccionaron las siguientes instancias del problema. Entrada: 44, 200, 800 y 122423. La salida del programa se observa en la Figura 4.

## 2. Ejercicio 2.

El máximo común divisor de dos enteros es el entero más grande que puede dividir a cada uno de los dos números. Programe el algoritmo para calcular el máximo común divisor en un método independiente.

### 2.1. Problema computacional.

**Objetivo:** Dado un número entero como año determinar si es un año bisiesto o no.

**Entrada:** Un número entero mayor que 0 representando un año.

**Salida:** La respuesta de si el número dado es un año bisiesto o no.

```
daniel@daniel-Lenovo-G470: ~/Escritorio/programacion/directorio/build
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
daniel@daniel-Lenovo-G470:~/Escritorio/programacion/directorio/build$ ./apps/calcular_area
Menu
1) Círculo
2) Cuadrado
3) Triángulo equilátero
4) Rectángulo
5) Salir
Seleccione una opción: 1
Radio: 4
Area = 50.2655
Menu
1) Círculo
2) Cuadrado
3) Triángulo equilátero
4) Rectángulo
5) Salir
Seleccione una opción: 4
Base: 2
Altura: 5
Area = 10
Menu
1) Círculo
2) Cuadrado
3) Triángulo equilátero
4) Rectángulo
5) Salir
Seleccione una opción: 5
Saliendo...
daniel@daniel-Lenovo-G470:~/Escritorio/programacion/directorio/build$
```

Figura 1: Ejecución de algunas instancias del problema.

## 2.2. Algoritmo.

Para solucionar el problema computacional, partimos de la definición de un año bisiesto. Un año se denomina bisiesto si es múltiplo de 4, a excepción de que sí es múltiplo de 100 también y no de 400 entonces no es bisiesto, por tanto necesitaremos utilizar el condicional `if-else` para validar que se cumplan las condiciones para que el año sea bisiesto o no.

El código fuente está disponible en mi repositorio de git hub. [1]

## 2.3. Instancia del problema.

Como prueba de escritorio, se seleccionaron las siguientes instancias del problema. Entrada: 44, 200, 800 y 122423. La salida del programa se observa en la Figura 4.

```
daniel@daniel-Lenovo-G470: ~/Escritorio/programacion/directorio/build
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
daniel@daniel-Lenovo-G470:~/Escritorio/programacion/directorio/build$ ./apps/calcular_area
Menu
1) Círculo
2) Cuadrado
3) Triángulo equilátero
4) Rectángulo
5) Salir
Seleccione una opción: 1
Radio: 4
Area = 50.2655
Menu
1) Círculo
2) Cuadrado
3) Triángulo equilátero
4) Rectángulo
5) Salir
Seleccione una opción: 4
Base: 2
Altura: 5
Area = 10
Menu
1) Círculo
2) Cuadrado
3) Triángulo equilátero
4) Rectángulo
5) Salir
Seleccione una opción: 5
Saliendo...
daniel@daniel-Lenovo-G470:~/Escritorio/programacion/directorio/build$
```

Figura 2: Ejecución de algunas instancias del problema.

### 3. Ejercicio 3.

Programe el algoritmo para determinar si el número es palíndromo y el algoritmo para validar la entrada en métodos independientes. Sugerencia: Haga uso de los operadores módulo y división para separar el número tecleado en unidades, decenas, centenas, etc.

#### 3.1. Problema computacional.

**Objetivo:** Dado un número entero como año determinar si es un año bisiesto o no.

**Entrada:** Un número entero mayor que 0 representando un año.

**Salida:** La respuesta de si el número dado es un año bisiesto o no.

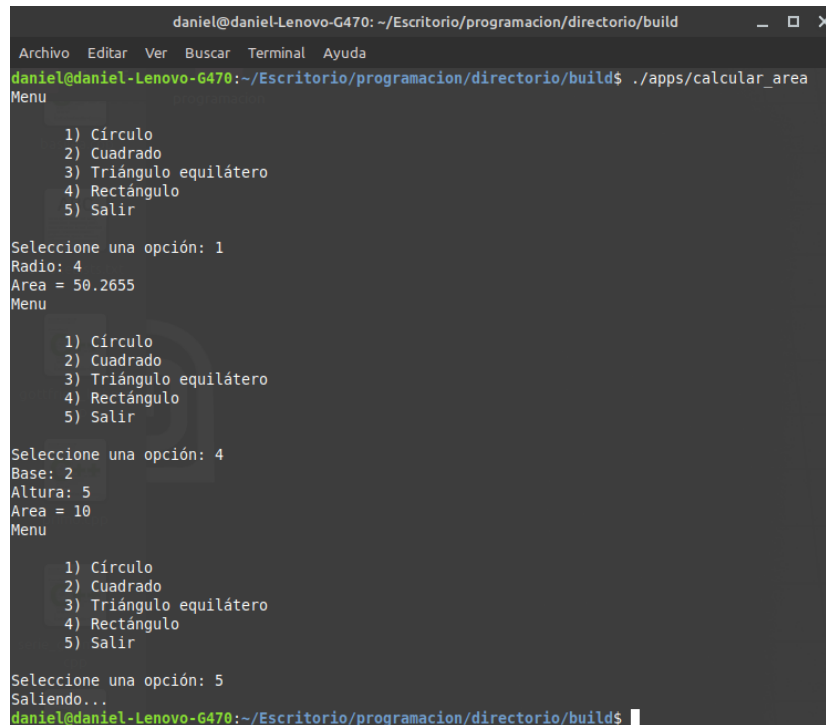
### 3.2. Algoritmo.

Para solucionar el problema computacional, partimos de la definición de un año bisiesto. Un año se denomina bisiesto si es múltiplo de 4, a excepción de que sí es múltiplo de 100 también y no de 400 entonces no es bisiesto, por tanto necesitaremos utilizar el condicional `if-else` para validar que se cumplan las condiciones para que el año sea bisiesto o no.

El código fuente está disponible en mi repositorio de git hub. [1]

### 3.3. Instancia del problema.

Como prueba de escritorio, se seleccionaron las siguientes instancias del problema. Entrada: 44, 200, 800 y 122423. La salida del programa se observa en la Figura 4.



```
daniel@daniel-Lenovo-G470: ~/Escritorio/programacion/directorio/build
Archivo Editar Ver Buscar Terminal Ayuda
daniel@daniel-Lenovo-G470:~/Escritorio/programacion/directorio/build$ ./apps/calcular_area
Menu
1) Círculo
2) Cuadrado
3) Triángulo equilátero
4) Rectángulo
5) Salir
Seleccione una opción: 1
Radio: 4
Area = 50.2655
Menu
1) Círculo
2) Cuadrado
3) Triángulo equilátero
4) Rectángulo
5) Salir
Seleccione una opción: 4
Base: 2
Altura: 5
Area = 10
Menu
1) Círculo
2) Cuadrado
3) Triángulo equilátero
4) Rectángulo
5) Salir
Seleccione una opción: 5
Saliendo...
daniel@daniel-Lenovo-G470:~/Escritorio/programacion/directorio/build$
```

Figura 3: Ejecución de algunas instancias del problema.

## 4. Ejercicio 4.

Programa el algoritmo de Schrage para generar números pseudo-aleatorios entre 0 y  $m$ . El  $j$ -ésimo elemento de la sucesión de números pseudo-aleatorios, denotado por  $I_j$ , se calcula con la siguiente ecuación:

$$I_j = a(I_{j-1} \bmod m)$$
$$I_j = \begin{cases} a(I_{j-1} \bmod q) - r[I_{j-1}/q] & \text{si } a(I_{j-1} \bmod q) - r[I_{j-1}/q] > 0 \\ a(I_{j-1} \bmod q) - r[I_{j-1}/q] + m & \text{en otro caso} \end{cases}$$

### 4.1. Problema computacional.

**Objetivo:** Dado un número entero como año determinar si es un año bisiesto o no.

**Entrada:** Un número entero mayor que 0 representando un año.

**Salida:** La respuesta de si el número dado es un año bisiesto o no.

### 4.2. Algoritmo.

Para solucionar el problema computacional, partimos de la definición de un año bisiesto. Un año se denomina bisiesto si es múltiplo de 4, a excepción de que sí es múltiplo de 100 también y no de 400 entonces no es bisiesto, por tanto necesitaremos utilizar el condicional **if-else** para validar que se cumplan las condiciones para que el año sea bisiesto o no.

El código fuente está disponible en mi repositorio de git hub. [1]

### 4.3. Instancia del problema.

Como prueba de escritorio, se seleccionaron las siguientes instancias del problema. Entrada: 44, 200, 800 y 122423. La salida del programa se observa en la Figura 4.

## 5. Conclusiones.

Las Estructura de control condicionada (**if-else** o **switch-case**), estructuras de control por repetición (**for**, **while**, **do-while**) pueden resolver muchos problemas en el cual se requieran ciclos o codiciones, pero si incluimos los array los problemas abarcables son innumerables.

```
daniel@daniel-Lenovo-G470: ~/Escritorio/programacion/directorio/build
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
daniel@daniel-Lenovo-G470:~/Escritorio/programacion/directorio/build$ ./apps/calcular_area
Menu
1) Círculo
2) Cuadrado
3) Triángulo equilátero
4) Rectángulo
5) Salir
Seleccione una opción: 1
Radio: 4
Area = 50.2655
Menu
1) Círculo
2) Cuadrado
3) Triángulo equilátero
4) Rectángulo
5) Salir
Seleccione una opción: 4
Base: 2
Altura: 5
Area = 10
Menu
1) Círculo
2) Cuadrado
3) Triángulo equilátero
4) Rectángulo
5) Salir
Seleccione una opción: 5
Saliendo...
daniel@daniel-Lenovo-G470:~/Escritorio/programacion/directorio/build$
```

Figura 4: Ejecución de algunas instancias del problema.

## Referencias

- [1] Daniel Reyes Barrera. *Problema 9*. 2020. URL: [https://github.com/danield877/cpp2020/blob/master/Clase\\_4\\_Tareas/basilea.cpp](https://github.com/danield877/cpp2020/blob/master/Clase_4_Tareas/basilea.cpp) (visitado 16-11-2020).