

Examen Final 0980*

Emerson Aldair Pérez Rivera, 201902852^{1, **}

¹Facultad de Ingeniería, Universidad de San Carlos,
Edificio T1, Ciudad Universitaria, Zona 12, Guatemala.

I. DIAGRAMA DE FLUJO

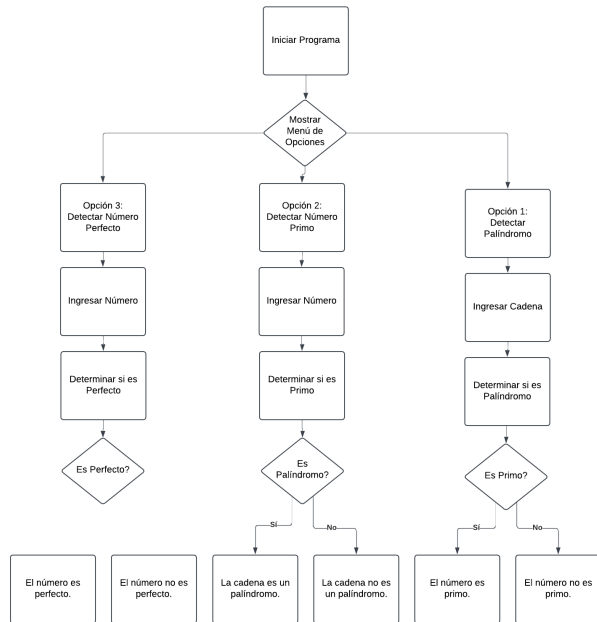


Figura 1

Fuente: Elaboracion Propia, 2024

II. ALGORITMO

A. Función es_palindromo

Entrada: Una cadena de caracteres `cadena`. **Proceso:**

- Comparar `cadena` con su inversa (`cadena[::-1]`).

Salida: Retornar `True` si `cadena` es igual a su inversa, de lo contrario `False`.

B. Función es_primo

Entrada: Un número entero `numero`. **Proceso:**

- Si `numero` es menor o igual a 1, retornar `False`.
- Para cada número `i` desde 2 hasta la raíz cuadrada de `numero`:

- Si `numero` es divisible por `i`, retornar `False`.
- Si no se encontró ningún divisor, retornar `True`.

Salida: Retornar `True` si `numero` es primo, de lo contrario `False`.

C. Función es_perfecto

Entrada: Un número entero `numero`. **Proceso:**

- Inicializar `suma_divisores` a 0.
- Para cada número `i` desde 1 hasta `numero-1`:
 - Si `numero` es divisible por `i`, sumar `i` a `suma_divisores`.
- Comparar `suma_divisores` con `numero`.

Salida: Retornar `True` si `suma_divisores` es igual a `numero`, de lo contrario `False`.

D. Función menu

Proceso:

- Mostrar el menú con opciones:
 - Detectar palíndromo.
 - Detectar número primo.
 - Detectar número perfecto.
 - Salir.
- Leer la opción ingresada por el usuario.
- Según la opción seleccionada:
 - Opción 1: Pedir una cadena, verificar si es un palíndromo usando `es_palindromo` y mostrar el resultado.
 - Opción 2: Pedir un número, verificar si es primo usando `es_primo` y mostrar el resultado.
 - Opción 3: Pedir un número, verificar si es perfecto usando `es_perfecto` y mostrar el resultado.
 - Opción 4: Mostrar mensaje de despedida y salir del bucle.
 - Cualquier otra opción: Mostrar mensaje de opción inválida.
- Repetir: El proceso se repite hasta que el usuario selecciona la opción de salir.

* Proyectos

** g-mail: 3741958620101@ingenieria.usac.edu.gt

III. SEUDOCODIGO

```

1 funci n es_palindromo(cadena):
2     retornar cadena == invertir(cadena)
3
4 funci n es_primo(numero):
5     si numero <= 1:
6         retornar False
7     para i desde 2 hasta ra z_cuadrada(numero):
8         si numero % i == 0:
9             retornar False
10    retornar True
11
12 funci n es_perfecto(numero):
13    suma_divisores = 0
14    para i desde 1 hasta numero-1:
15        si numero % i == 0:
16            suma_divisores += i
17    retornar suma_divisores == numero
18
19 funci n menu():
20    mientras Verdadero:
21        mostrar "Men : 1. Detectar pal ndromo
22                2. Detectar n mero primo 3. Detectar
23                n mero perfecto 4. Salir"
24        opcion = leer entrada del usuario como
25        entero
26        si opcion == 1:
27            cadena = leer cadena del usuario
28            si es_palindromo(cadena):
29                mostrar "La cadena es un
30                pal ndromo."
31            sino:
32                mostrar "La cadena no es un
33                pal ndromo."
34        sino si opcion == 2:
35            numero = leer n mero del usuario
36            si es_primo(numero):
37                mostrar "El n mero es primo."
38            sino:
39                mostrar "El n mero no es primo."
40        "
41        sino si opcion == 3:
42            numero = leer n mero del usuario
43            si es_perfecto(numero):
44                mostrar "El n mero es perfecto."
45        "
46        sino:
47            mostrar "El n mero no es
48            perfecto."
49        sino si opcion == 4:
50            mostrar " Hasta luego!"
51            romper bucle
52        sino:
53            mostrar "Opci n inv lida. Intente
54            nuevamente."
55
56 si __name__ == "__main__":
57     menu()

```

IV. CODIGO PYTHON

```

1
2 def es_palindromo(cadena):
3     """Determina si una cadena es un pal ndromo
4     ."""

```

```

4     return cadena == cadena[::-1]
5
6 def es_primo(numero):
7     """Determina si un n mero es primo."""
8     if numero <= 1:
9         return False
10    for i in range(2, int(numero**0.5) + 1):
11        if numero % i == 0:
12            return False
13    return True
14
15 def es_perfecto(numero):
16     """Determina si un n mero es perfecto."""
17    suma_divisores = 0
18    for i in range(1, numero):
19        if numero % i == 0:
20            suma_divisores += i
21    return suma_divisores == numero
22
23 def menu():
24     """Muestra el men de opciones y realiza
25     las operaciones correspondientes."""
26    while True:
27        print("\nMen :")
28        print("1. Detectar pal ndromo")
29        print("2. Detectar n mero primo")
30        print("3. Detectar n mero perfecto")
31        print("4. Salir")
32        opcion = int(input("Ingrese una opci n:
33        "))
34
35        if opcion == 1:
36            cadena = input("Ingrese una cadena:
37            ")
38            if es_palindromo(cadena):
39                print("La cadena es un
40                pal ndromo.")
41            else:
42                print("La cadena no es un
43                pal ndromo.")
44            elif opcion == 2:
45                numero = int(input("Ingrese un
46                n mero: "))
47                if es_primo(numero):
48                    print("El n mero es primo.")
49                else:
50                    print("El n mero no es primo.")
51            elif opcion == 3:
52                numero = int(input("Ingrese un
53                n mero: "))
54                if es_perfecto(numero):
55                    print("El n mero es perfecto.")
56                else:
57                    print("El n mero no es perfecto
58                    .")
59            elif opcion == 4:
60                print(" Hasta luego!")
61                break
62            else:
63                print("Opci n inv lida. Intente
64                nuevamente.")
65
66 if __name__ == "__main__":
67     menu()
68
69

```

V. METIGACION DE ERRORES

- Error de tipo al ingresar una opción que no es un número entero.
- Ingreso de un número negativo para las opciones 2 y 3 (números primos y perfectos).
- Ingreso de una cadena vacía o solo espacios en blanco para la detección de palíndromos.
- Selección de una opción no válida en el menú.
- Ingreso de un número muy grande que podría causar un tiempo de espera prolongado en la detección de números primos o perfectos.
- Error de división por cero si se intenta verificar el número cero como primo o perfecto.
- Fallo al usar caracteres especiales en la entrada de palíndromo (dependiendo de la lógica esperada).
- Entrada de texto en lugar de números para las opciones 2 y 3.
- Posibles problemas de codificación si se ingresan caracteres no ASCII en algunas configuraciones.
- Error al intentar convertir entradas que no son números enteros en las opciones que lo requieren.

VI. DIAGRAMA DE GANTT

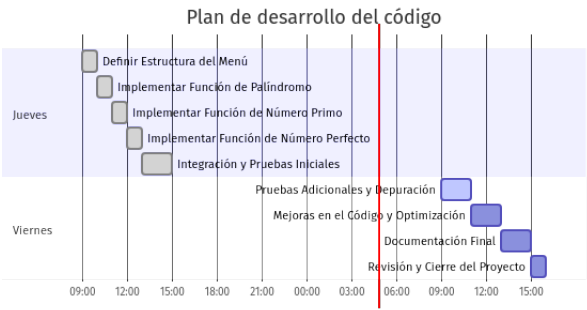


Figura 2

Fuente: Elaboracion Propia, 2024

VII. REPOSITORIO GITHUB

