

# Examen final: Verificación del tipo de número.

Alan Andrés Mérida Morales, 202100023<sup>1,\*</sup>

<sup>1</sup>Facultad de Ingeniería, Escuela de Ingeniería Mecánica Eléctrica,  
Universidad de San Carlos, Ciudad Universitaria, Zona 12, Guatemala.

## I. INTRODUCCIÓN

Este programa es una aplicación de consola en Python y Octave que permite a los usuarios ingresar números y verificar ciertas propiedades matemáticas de estos números. La aplicación está conectada a una base de datos PostgreSQL para gestionar la información de los usuarios y almacenar los resultados de las verificaciones. Los usuarios pueden registrarse e iniciar sesión con su nombre y carné. Una vez autenticados, pueden ingresar un número y verificar si es primo, perfecto o palíndromo. Los resultados de estas verificaciones se guardan tanto en la base de datos como en un archivo de texto con un formato que incluye el nombre del usuario, su carné, el número ingresado y las propiedades verificadas. Además, la aplicación permite ver el historial de las verificaciones realizadas y ofrece la opción de borrar los registros tanto en la base de datos como en el archivo de texto. El programa proporciona una interfaz de menú interactivo para facilitar la navegación y el uso de las diferentes funcionalidades.

## II. RESULTADOS

### A. Códigos realizados

#### 1. Python

```
1 import psycopg2
2 import os
3 import re
4
5 # Conexión a la base de datos
6 conn = psycopg2.connect(
7     dbname='user',
8     user='postgres',
9     password='@merida',
10    host='localhost',
11    port='5432'
12 )
13 cursor = conn.cursor()
14
15 # Nombre del archivo de texto
16 archivo_texto = 'resultados_verificaciones.txt'
17
18 # Verificar si el archivo de texto existe, si no, crearlo
19 if not os.path.exists(archivo_texto):
20     with open(archivo_texto, 'w') as archivo:
21         archivo.write('')
22     print('El archivo de texto ha sido creado.')
23 else:
24     print('El archivo de texto ya existe.')
25
26 # Función para validar que no haya caracteres especiales
27 def contiene_caracteres_especiales(cadena):
28     return bool(re.search(r'["a-zA-Z0-9 ]', cadena))
29
30 # Función para pedir una opción numérica
31 def pedir_opcion_numerica(mensaje):
32     while True:
33         opcion = input(mensaje).strip()
34         if opcion.isdigit():
35             return opcion
```

Figura 1: Código realizado en python

```
36 else:
37     print('Por favor ingrese solo números.')
38
39 # Menú inicial para ingresar usuario
40 while True:
41     print('--- Menú Inicial ---')
42     print('1. Ingresar Usuario')
43     print('2. Salir')
44     opcion_inicial = pedir_opcion_numerica('Ingrese su elección: ')
45
46     if opcion_inicial == '2':
47         print('Saliendo del programa...')
48         break
49     elif opcion_inicial == '1':
50         respuesta = input('¿Ya tienes un usuario registrado? (Y/N): ').strip().upper()
51
52         if respuesta == 'Y':
53             nombre = input('Ingrese su nombre: ').strip()
54             carne = input('Ingrese su carné (9 dígitos): ').strip()
55
56             if contiene_caracteres_especiales(nombre):
57                 print('El nombre no debe contener caracteres especiales.')
58                 continue
59
60             if not carne.isdigit() or len(carne) != 9:
61                 print('El carné debe contener exactamente 9 dígitos numéricos.')
62                 continue
63
64             cursor.execute('SELECT id FROM usuario WHERE nombre = %s AND carne = %s',
65                             (nombre, carne))
66             result = cursor.fetchone()
```

Figura 2: Código realizado en python

```
67 if not result:
68     print('Usuario no encontrado. Por favor, registre primero los datos del
69     usuario.')
70     continue
71 else:
72     print('¡Bienvenido de nuevo!')
73
74 while True:
75     print('--- Menú de Usuario ---')
76     print('1. Ingresar datos')
77     print('2. Ver historial de datos')
78     print('3. Borrar datos')
79     print('4. Salir')
80     opcion_usuario = pedir_opcion_numerica('Ingrese su elección: ')
81
82     if opcion_usuario == '1':
83         numero_str = input('Ingrese el número a verificar: ').strip()
84
85         if not numero_str.isdigit():
86             print('Por favor ingrese solo números.')
87         else:
88             numero = int(numero_str)
89
90             # Verificar si es primo
91             es_primo = numero > 1 and all(numero % i != 0 for i in range(2,
92                                     int(numero ** 0.5) + 1))
93
94             if es_primo:
95                 print('El número es primo.')
96             else:
97                 print('El número no es primo.')
98
99             # Verificar si es perfecto
100             suma_divisores = sum(i for i in range(1, numero) if numero % i
101                                 == 0)
102             es_perfecto = suma_divisores == numero
103             if es_perfecto:
104                 print('El número es perfecto.')
105             else:
```

Figura 3: Código realizado en python

\* 3690273450101@ingenieria.usac.edu.gt

```

1802         print('El número no es perfecto.')
1803
1804     # Verificar si es palindromo
1805     es_palindromo = str(numero) == str(numero[::-1])
1806     if es_palindromo:
1807         print('El número es palindromo.')
1808     else:
1809         print('El número no es palindromo.')
1810
1811     # Insertar los resultados en la base de datos
1812     cursor.execute('INSERT INTO verificaciones (id_usuario,
1813 numero, es_primo, es_perfecto, es_palindromo)
1814     VALUES (%s, %s, %s, %s, %s);',
1815         (result[0], numero, es_primo, es_perfecto,
1816          es_palindromo))
1817
1818     conn.commit()
1819     print('Los resultados han sido guardados en la base de datos.')
1820
1821     # Guardar en el archivo de texto
1822     with open(archivo_texto, 'a') as archivo:
1823         archivo.write(f'Usuario: {nombre} (Carné: {carne})\n')
1824         archivo.write(f'Número: {numero}\n')
1825         archivo.write(f'Es primo: {1 if es_primo else 0}\n')
1826         archivo.write(f'Es perfecto: {1 if es_perfecto else 0}\n')
1827         archivo.write(f'Es palindromo: {1 if es_palindromo else
1828 0}\n')
1829
1830     print('Datos guardados correctamente.')
1831
1832     elif opcion_usuario == '2':
1833         # Mostrar historial de datos desde la tabla 'verificaciones'
1834         cursor.execute('SELECT id_usuario, numero, es_primo, es_perfecto,
1835 es_palindromo FROM verificaciones;')
1836         result = cursor.fetchall()
1837
1838         if not result:
1839             print('No hay datos registrados.')
1840         else:

```

Figura 4: Código realizado en python

```

1336         print('----- Historial de verificaciones -----')
1337         print('ID Usuario|Número|Primo|Perfecto|Palindromo')
1338
1339         # Iterar sobre los resultados y mostrarlos
1340         for row in result:
1341             id_usuario, numero, es_primo, es_perfecto, es_palindromo =
1342             row
1343             print(f'{id_usuario}|{numero}|{es_primo}|{es_perfecto}|{es_palindromo}')
1344
1345         elif opcion_usuario == '3':
1346             try:
1347                 # Borrar los registros de la tabla 'verificaciones'
1348                 cursor.execute('DELETE FROM verificaciones;')
1349                 conn.commit()
1350
1351             except Exception as e:
1352                 print(f'Error al borrar los datos: {e}')
1353
1354             print('Todos los datos de la tabla de verificaciones han sido
1355 borrados.')
1356
1357             # Borrar los datos del archivo de texto 'resultados_verifica-
1358             ones.txt'
1359             if os.path.exists(archivo_texto):
1360                 # Si el archivo existe, borrarlo
1361                 with open(archivo_texto, 'w') as archivo:
1362                     archivo.write('')
1363             print('Los datos del archivo de texto "resultados_verifica-
1364             ones.txt" han sido borrados.')
1365         else:
1366             print('El archivo "resultados_verificaciones.txt" no
1367             existe.')
1368
1369     except Exception as e:
1370         print(f'Error al borrar los datos: {e}')
1371

```

Figura 5: Código realizado en python

```

1602         elif opcion_usuario == '4':
1603             print('Saliedo al menú inicial...')
1604             break
1605         else:
1606             print('Opción no válida. Por favor, elija una opción entre 1 y 4.')
1607
1608     elif respuesta == 'N':
1609         # Registrar nuevo usuario
1610         nombre = input('Ingrese su nombre: ').strip()
1611         carne = input('Ingrese su carne (9 dígitos): ').strip()
1612
1613         if contiene_caracteres_especiales(nombre):
1614             print('El nombre no debe contener caracteres especiales.')
1615             continue
1616
1617         if not carne.isdigit() or len(carne) != 9:
1618             print('El carné debe contener exactamente 9 dígitos numéricos.')
1619             continue
1620
1621         cursor.execute('INSERT INTO usuario (nombre, carne) VALUES (%s, %s);', (nombre,
1622 carne))
1623
1624         conn.commit()
1625         print('Usuario registrado correctamente. Ahora puedes iniciar sesión.')
1626     else:
1627         print('Respuesta no válida. Debes ingresar "Y" o "N".')
1628
1629 # Cerrar la conexión a la base de datos
1630 cursor.close()
1631 conn.close()
1632

```

Figura 6: Código realizado en python

## 2. Octave

```

1  if (exist('OCTAVE_VERSION', 'builtin') == 0)
2      % Estamos en Octave
3      pkg load database; % Cargar el paquete para interactuar con bases de datos
4  end
5
6  % Inicialización de conexión con la base de datos
7  conn = pg_connect(sprintf('dbname=%s;user=%s;host=%s;localhost',
8  'port', '5432', 'user', 'postgres', 'password', 'Almerida'));
9
10 % Nombre del archivo de texto para guardar los resultados
11 archivo_texto = 'verificaciones.txt';
12
13 % Verificar si el archivo de texto existe, si no, crearlo
14 if exist(archivo_texto, 'file') == 0
15     fid = fopen(archivo_texto, 'w');
16     fclose(fid);
17 else
18     disp('El archivo de texto ya existe.');
```

Figura 7: Código realizado en octave

```

42     if isempty(result.data)
43         disp('Usuario no encontrado. Por favor, registre primero los datos del usuario.');
```

Figura 8: Código realizado en octave

```

81     case 1
82         % Verificar que ya hay un usuario registrado
83         nombre = nombre;
84         carne = carne;
85
86         % Buscar al usuario en la base de datos
87         result = pg_exec_params(conn, 'SELECT id FROM usuario WHERE nombre = $1 AND carne = $2;', (nombre, carne));
88
89         if isempty(result.data)
90             disp('Usuario no encontrado. Por favor, registre los datos del usuario primero.');
```

Figura 9: Código realizado en octave

```

121         if es_primo
122             disp('El número es primo.');
```

```

123         else
124             disp('El número no es un número primo.');
```

```

125         end
126     end
127
128     % Verificar si el número es perfecto
129     suma_divisores = 0;
130     for i = 1:(numero - 1)
131         if mod(numero, i) == 0
132             suma_divisores = suma_divisores + i;
133         end
134     end
135
136     if suma_divisores == numero
137         es_perfecto = true;
138         disp('El número es un número perfecto.');
```

```

139     else
140         es_perfecto = false;
141         disp('El número no es un número perfecto.');
```

```

142     end
143
144     % Verificar si el número es palindromo
145     num_str = num2str(numero); % Convertir el número a cadena de texto
146     if strcmp(num_str, flip(num_str))
147         es_palindromo = true;
148         disp('El número es un número palindromo.');
```

```

149     else
150         es_palindromo = false;
151         disp('El número no es un número palindromo.');
```

```

152     end
153
154     % Insertar los resultados en la base de datos
155     pq_exec_params(conn, ['INSERT INTO verificaciones (id_usuario, numero,
156     es_primo, es_perfecto, es_palindromo) ...
157     VALUES ($1, $2, $3, $4, $5)'], ...
158     {id_usuario, numero, es_primo, es_perfecto,
159     es_palindromo});
158     disp('Los resultados han sido guardados en la base de datos.');
```

Figura 10: Código realizado en octave

## B. Diagrama de flujo

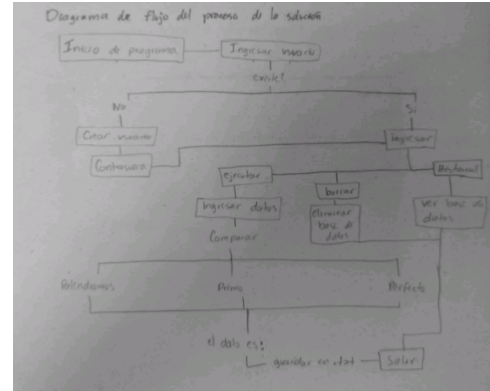


Figura 13: Diagrama de flujo del programa realizado

## C. Algoritmo

```

159     % Guardar los resultados en el archivo de texto
160     fid = fopen(archivo_texto, 'a'); % Abrir el archivo en modo "append"
161     fprintf(fid, 'Usuario: %s (Carné: %s)\n', nombre, carne);
162     fprintf(fid, 'Número: %d\n', numero);
163     fprintf(fid, 'Es primo: %d\n', es_primo);
164     fprintf(fid, 'Es perfecto: %d\n', es_perfecto);
165     fprintf(fid, 'Es palindromo: %d\n', es_palindromo);
166     fclose(fid); % Cerrar el archivo
167     disp('Los resultados también han sido guardados en el archivo de texto.');
```

```

168     end
169
170     case 2
171         % Mostrar el historial de datos directamente desde la tabla 'verificaciones'
172         result = pq_exec_params(conn, 'SELECT id_usuario, numero, es_primo, es_perfecto,
173         es_palindromo FROM verificaciones');
```

```

174         if isempty(result.data)
175             disp('No hay datos registrados.');
```

```

176         else
177             disp('----- Historial de verificaciones -----');
```

```

178             disp('ID Usuario\tNúmero\tPrimo\tPerfecto\tPalindromo');
```

```

179             % Iterar sobre los resultados y mostrarlos de forma legible
180             for i = 1:size(result.data, 1)
181                 id_usuario = result.data(i, 1);
182                 numero = result.data(i, 2);
183                 es_primo = result.data(i, 3);
184                 es_perfecto = result.data(i, 4);
185                 es_palindromo = result.data(i, 5);
186             end
187
188             % Mostrar la información de manera formateada
189             fprintf('%d\t%d\t%d\t%d\t%d\n', id_usuario, numero, es_primo, es_perfecto,
190             es_palindromo);
191         end
192     end
193
194     case 3
195         % Borrar todos los datos de la tabla 'verificaciones'
196         try
197             pq_exec_params(conn, 'DELETE FROM verificaciones');
```

```

198             disp('Todos los datos de la tabla de verificaciones han sido borrados.');
```

Figura 11: Código realizado en octave

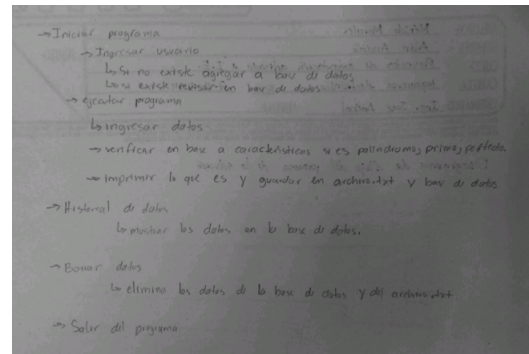


Figura 14: Algoritmo del programa realizado

## D. Funcionamiento del programa

### 1. Python

```

200     % Borrar los datos del archivo de texto 'verificacion.txt'
201     if exist('verificacion.txt', 'file') == 2
202         % Si el archivo existe, vaciarlo
203         fclose(fopen('verificacion.txt', 'w'));
204         disp('Los datos del archivo de texto "verificacion.txt" han sido borrados.');
```

```

205     else
206         disp('El archivo "verificacion.txt" no existe.');
```

```

207     end
208
209     catch
210         disp('Error al borrar los datos.');
```

```

211     end
212
213     case 4
214         disp('Saliedo del menú principal...');
```

```

215     break;
216
217     otherwise
218         disp('Opción no válida. Intente de nuevo.');
```

```

219     end
220
221     else
222         disp('Opción no válida. Intente de nuevo.');
```

```

223     end
224 end
```

Figura 12: Código realizado en octave



Figura 15: Funcionamiento del programa en Python

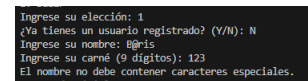


Figura 16: Funcionamiento del programa en Python

```
--- Menú Inicial ---
1. Ingresar Usuario
2. Salir
Ingrese su elección: 1
¿Ya tienes un usuario registrado? (Y/N): Y
Ingrese su nombre: Boris
Ingrese su carné (9 dígitos): 199900023
¡Bienvenido de nuevo!
--- Menú de Usuario ---
1. Ingresar datos
2. Ver historial de datos
3. Borrar datos
4. Salir
Ingrese su elección: |
```

Figura 17: Funcionamiento del programa en Python

```
Ingrese su elección: 1
Tipos de números
1. Un número primo es aquel que solo tiene dos divisores: 1 y el mismo número.
2. Un número perfecto es aquel que es igual a la suma de sus divisores propios (exceptuando el mismo número).
3. Un número palíndromo es aquel que se lee igual de izquierda a derecha que de derecha a izquierda.
Ingrese el dato que desea verificar (solo números): 33
El número no es un número primo.
El número no es un número perfecto.
El número es un número palíndromo.
Los resultados han sido guardados en la base de datos.
Los resultados también han sido guardados en el archivo de texto.
--- Menú Principal ---
1. Ingresar datos
2. Ver historial de datos
3. Borrar datos
4. Salir
Ingrese su elección: 2
----- Historial de verificaciones -----
ID Usuario\Número\Primo\Perfecto\Palíndromo
1 33 0 0 1
--- Menú Principal ---
1. Ingresar datos
2. Ver historial de datos
3. Borrar datos
4. Salir
Ingrese su elección: |
```

Figura 22: Funcionamiento del programa en Octave

```
Ingrese su elección: 1
Ingrese el número a verificar: 513
El número no es primo.
El número no es perfecto.
El número no es palíndromo.
Los resultados han sido guardados en la base de datos.
Datos guardados correctamente.
--- Menú de Usuario ---
1. Ingresar datos
2. Ver historial de datos
3. Borrar datos
4. Salir
Ingrese su elección: |
```

Figura 18: Funcionamiento del programa en Python

```
Ingrese su elección: 3
Todos los datos de la tabla de verificaciones han sido borrados.
Los datos del archivo de texto "verificacion.txt" han sido borrados.
--- Menú Principal ---
1. Ingresar datos
2. Ver historial de datos
3. Borrar datos
4. Salir
Ingrese su elección: 4
Saliendo del menú principal...
--- Menú Inicial ---
1. Ingresar Usuario
2. Salir
Ingrese su elección: |
```

Figura 23: Funcionamiento del programa en Octave

```
Ingrese su elección: 2
----- Historial de verificaciones -----
ID Usuario Número Primo Perfecto Palíndromo
7 513 False False False
--- Menú de Usuario ---
```

Figura 19: Funcionamiento del programa en Python

3. Base de datos

```
--- Menú de Usuario ---
1. Ingresar datos
2. Ver historial de datos
3. Borrar datos
4. Salir
Ingrese su elección: 3
Todos los datos de la tabla de verificaciones han sido borrados.
Los datos del archivo de texto "resultados_verificaciones.txt" han sido borrados.
--- Menú de Usuario ---
```

Figura 20: Funcionamiento del programa en Python

2. Octave

```
>> tarea
El archivo de texto ya existe.
--- Menú Inicial ---
1. Ingresar Usuario
2. Salir
Ingrese su elección: 1
¿Ya tienes un usuario registrado? (Y/N): N
Ingrese su nombre: B@ris
Ingrese su carné: 123
El nombre solo debe contener letras.
--- Menú Inicial ---
1. Ingresar Usuario
2. Salir
Ingrese su elección: 1
¿Ya tienes un usuario registrado? (Y/N): Y
Ingrese su nombre: Alan
Ingrese su carné: 202100023
¡Bienvenido de nuevo!
--- Menú Principal ---
1. Ingresar datos
2. Ver historial de datos
3. Borrar datos
4. Salir
Ingrese su elección: |
```

Figura 21: Funcionamiento del programa en Octave

id	nombre	carne
1	Alan	202100023
2	Alan	202100023
3	Mafer	123456789
4	Sara	202012345
5	Luisa	098765432
6	Marcos	234567890
7	Boris	199900023

Figura 24: Base de datos con usuarios registrados

id	id_usuario	numero	es_primo	es_perfecto	es_palindromo
1	25	7	1441	false	false
2	26	7	6	false	true
3	27	7	737	false	false

Figura 25: Base de datos con datos verificados

#### 4. Archivos de texto

```

Usuario: Boris (Carné: 199900023)
Número: 31
Es primo: 1
Es perfecto: 0
Es palíndromo: 0

Usuario: Boris (Carné: 199900023)
Número: 6
Es primo: 0
Es perfecto: 1
Es palíndromo: 1

Usuario: Boris (Carné: 199900023)
Número: 2920
Es primo: 0
Es perfecto: 0
Es palíndromo: 0

Usuario: Boris (Carné: 199900023)
Número: 141
Es primo: 0
Es perfecto: 0
Es palíndromo: 1

```

Figura 26: Datos registrados en archivo de texto octave

```

Usuario: Alan (Carné: 202100023)
Número: 3
Es primo: 1
Es perfecto: 0
Es palíndromo: 1

Usuario: Alan (Carné: 202100023)
Número: 6
Es primo: 0
Es perfecto: 1
Es palíndromo: 1

Usuario: Alan (Carné: 202100023)
Número: 141
Es primo: 0
Es perfecto: 0
Es palíndromo: 1

```

Figura 27: Datos registrados en archivo de texto python

#### E. Repositorio privado

Repositorio creado en Github

#### F. Videos del funcionamiento

Video del funcionamiento del programa

### III. CONCLUSIONES

- \* La implementación de las soluciones en Python y Octave demuestra la adaptabilidad de los algoritmos matemáticos y de gestión de datos en distintos lenguajes de programación. Python aporta un enfoque moderno y versátil, mientras que Octave se destaca por sus capacidades en cálculos matemáticos y científicos.
- \* En Python, la integración con PostgreSQL muestra cómo interactuar con bases de datos de manera eficiente, garantizando la persistencia de los datos y la gestión estructurada de la información de los usuarios. En Octave, aunque más limitado en cuanto a bases de datos, se demostró cómo procesar datos y realizar verificaciones matemáticas.
- \* Tanto en Python como en Octave, se desarrollaron funciones para verificar si un número es primo, perfecto o palíndromo. Esto subraya la importancia de adaptar algoritmos matemáticos a diferentes entornos de programación, destacando las fortalezas de Octave en operaciones numéricas y la flexibilidad de Python.
- \* El programa en Python no solo se conecta a la base de datos, sino que también gestiona archivos de texto como respaldo, lo que proporciona una solución híbrida para la persistencia de datos. En Octave, la lógica de almacenamiento fue más enfocada a la demostración y procesamiento de datos sin conexión con bases de datos.
- \* La inclusión de validaciones de entrada tanto en Python como en Octave asegura la robustez y seguridad de los programas, evitando errores comunes y mejorando la experiencia del usuario.