
Proyectos de computación aplicados a I.E.

David Antonio Rodas Alvarez
David Steven Solís Sosa

202010039
202001569

Fecha: Guatemala, 7 de noviembre del 2024

Examen Final*

David Antonio, Rodas Alvarez, 202010039^{1, **}

¹Escuela de Ingeniería Mecánica Eléctrica, Universidad de San Carlos, Guatemala.

(Dated: 8 de noviembre de 2024)

I. RESUMEN

El propósito de este examen final es evaluar la capacidad del estudiante para integrar conceptos clave de programación en Python, tales como el manejo de cadenas y números, estructuras de control de flujo y la conexión a bases de datos. El programa también enfatiza el uso de funciones que cumplen con un objetivo específico, y la implementación de un menú interactivo que permite al usuario navegar y ejecutar las funcionalidades del programa de manera fluida. A través de este ejercicio, se destaca la habilidad del estudiante para implementar y manejar múltiples funciones, estructuras de datos y métodos de persistencia de información, habilidades fundamentales para el desarrollo de aplicaciones de software en la vida real.

ALGORITMO

Este algoritmo implementa un programa en Python que permite a un usuario detectar si una oración es un palíndromo, si un número es primo, o si un número es perfecto. Además, el programa guarda un historial de las operaciones realizadas en una base de datos PostgreSQL y en un archivo de texto.

1. Inicialización de la conexión a la base de datos

- Conéctate a una base de datos PostgreSQL llamada `detector` usando los credenciales proporcionados.
- Crea un cursor para realizar operaciones en la base de datos.

2. Función para almacenar en archivo de texto

- La función `save_to_history(data)` guarda una línea de texto en el archivo `history.txt` con la información de cada operación.

3. Función para detectar palíndromos

- La función `is_palindrome(sentence)` normaliza una oración eliminando caracteres no alfabéticos y tildes, y verifica si la oración es un palíndromo.

4. Función para detectar números primos

- La función `is_prime(number)` verifica si un número es primo, devolviendo `True` si el número es primo y `False` en caso contrario.

5. Función para detectar números perfectos

- La función `is_perfect(number)` verifica si un número es perfecto, devolviendo `True` si el número es perfecto y `False` en caso contrario.

6. Menú del detector

- La función `detector_menu()` presenta un menú con tres opciones:

a) Opción 1: Detector de Palíndromos

- Solicita al usuario una oración y verifica si es palíndroma usando `is_palindrome`.
- Almacena el resultado en la base de datos y en el archivo de texto.

b) Opción 2: Detector de Números Primos

- Solicita al usuario un número y verifica si es primo usando `is_prime`.
- Almacena el resultado en la base de datos y en el archivo de texto.

c) Opción 3: Detector de Números Perfectos

- Solicita al usuario un número y verifica si es perfecto usando `is_perfect`.
- Almacena el resultado en la base de datos y en el archivo de texto.

- Después de cada operación, se pregunta al usuario si desea realizar otra operación o salir del menú.

7. Menú principal

- La función `main_menu()` solicita al usuario su nombre de usuario y presenta cuatro opciones en el menú principal:

a) Opción 1: Detector

- Inicia el menú del detector (`detector_menu()`).

b) Opción 2: Historial de Datos Ingresados

* Escuela de Ingeniería Mecánica Eléctrica

** e-mail: 3711111370101@ingenieria.usac.edu.gt

- Consulta y muestra todos los registros en la tabla **detect** de la base de datos.

c) Opción 3: Borrar Datos

- Solicita el nombre de usuario cuyos datos serán eliminados y borra esos registros de la base de datos.

d) Opción 4: Salir

- Cierra la conexión a la base de datos y finaliza el programa.

II. DIAGRAMA DE GRANT

	9:00	10:00	11:00	12:00
Ideas Texto escrito en tiempo de examen final				
Elaboración Código en Python y Octave				
Preparación Reporte para subir a UEDI				
Elaboración Modelo al proyecto				
Preparación Reporte para subir a UEDI				
Publicación de reporte a UEDI				

III. CÓDIGO

A. Octave

```
pkg load database; # Cargar el paquete necesario para trabajar
con bases de datos

function conn = conectar_db()
    try
        conn = pq_connect(setdbopts('dbname', 'detect', 'host',
            'localhost', 'port', '5432', 'user', 'postgres',
            'password', 'Dali6478'));
    catch
        disp("Error al conectar a la base de datos");
        return;
    end
end

% Funcin para almacenar en el archivo .txt
function save_to_history(data)
    fid = fopen("history.txt", "a");
    fprintf(fid, "%s\n", data);
```

```
fclose(fid);
endfunction

% Funcin para verificar si una palabra es palndroma
function result = is_palindrome(sentence)
    normalized = lower(strrep(sentence, ' ', ''));
    normalized = unideblank(normalized); % Elimina tildes
    result = strcmp(normalized, fliplr(normalized)); % Compara si
    es igual al reverso
endfunction

% Funcin para verificar si un nmero es primo
function result = is_prime(number)
    if number <= 1
        result = false;
    else
        result = true;
        for i = 2:sqrt(number)
            if mod(number, i) == 0
                result = false;
                break;
            end
        end
    end
endfunction

% Funcin para verificar si un nmero es perfecto
function result = is_perfect(number)
    if number <= 1
        result = false;
    else
        sum_divisors = sum(find(mod(number, 1:number-1) == 0));
        result = sum_divisors == number;
    end
endfunction

% Funcin principal del men del detector
function detector_menu()
    while true
        printf("\n1) Detector de palndromos\n2) Detector de
            nmeros primos\n3) Detector de nmeros perfectos\n");
        option = input("Selecciona una opcin: ", "s");

        if option == '1'
            sentence = input("Ingresa una oracin: ", "s");
            if isempty(regexpi(sentence, '[a-zA-Z]', 'once'))
                printf("Error: Solo se permiten letras en la
                    oracin.\n");
                continue;
            endif
            result = is_palindrome(sentence);
            result_str = 'NO';
            if result
                result_str = 'SI';
            endif
            query = sprintf("INSERT INTO detect (palindromos,
                resultado) VALUES ('%s', '%s')", sentence,
                result_str);

            message = sprintf("La oracin '%s' es palndroma",
                result_str);
            printf("%s\n", message);
            save_to_history(sprintf("%s - Palndromo: %s -
                Resultado: %s", datestr(now, 'yyyy-mm-dd
                HH:MM:SS'), sentence, result_str));

        elseif option == '2'
            number = input("Ingresa un nmero: ");
            if !isnumeric(number)
                printf("Error: Solo se permiten nmeros.\n");
                continue;
            endif
            result = is_prime(number);
            result_str = 'NO';
            if result
                result_str = 'SI';
            endif
            query = sprintf("INSERT INTO detect (numero,
                resultado) VALUES (%d, '%s')", number,
                result_str);

            message = sprintf("El nmero '%s' es primo", result_str);
            printf("%s\n", message);
```

```

save_to_history(sprintf("%s - Nmero primo: %d -
Resultado: %s", datestr(now, 'yyyy-mm-dd
HH:MM:SS'), number, result_str));

elseif option == '3'
    number = input("Ingresa un nmero: ");
    if !isnumeric(number)
        printf("Error: Solo se permiten nmeros.\n");
        continue;
    endif
    result = is_perfect(number);
    result_str = 'NO';
    if result
        result_str = 'SI';
    endif
    query = sprintf("INSERT INTO detect (numero,
        resultado) VALUES (%d, '%s')", number,
        result_str);
    pq_exec(conn, query);
    message = sprintf("El nmero %s es perfecto",
        result_str);
    printf("%s\n", message);
    save_to_history(sprintf("%s - Nmero perfecto: %d -
        Resultado: %s", datestr(now, 'yyyy-mm-dd
        HH:MM:SS'), number, result_str));

else
    printf("Opcin invlida. Intntalo de nuevo.\n");
    continue;
endif

again = input("Deseas realizar otra operacin? (s/n): ",
    "s");
if again == 's'
    return; % Si el usuario elige 's', reiniciar el
    programa
elseif again == 'n'

    printf("Saliendo del programa...\n");
    return; % Detener el programa
endif
endwhile
endfunction

% Men principal
function main_menu()
    while true
        user = input("Ingresa tu nombre de usuario: ", "s");
        query = sprintf("INSERT INTO detect (usuario) VALUES
            ('%s')", user);

        while true
            printf("\n1) Detector\n2) Historial de datos
                ingresados\n3) Borrar datos\n4) Salir\n");
            option = input("Selecciona una opcin: ", "s");

            if option == '1'
                detector_menu();
            elseif option == '2'
                result = pq_exec(conn, "SELECT * FROM detect");
                disp(result);
            elseif option == '3'
                user_to_delete = input("Ingresa el nombre de
                    usuario cuyos datos deseas borrar: ", "s");
                query = sprintf("DELETE FROM detect WHERE usuario
                    = '%s'", user_to_delete);

                printf("Datos de %s eliminados.\n",
                    user_to_delete);
            elseif option == '4'
                pq_close(conn); % Cerrar la conexin
                printf("Saliendo del programa...\n");
                return; % Detener el programa
            else
                printf("Opcin invlida. Intntalo de nuevo.\n");
            endif
        endwhile
    endwhile
endfunction

% Ejecucin del programa

```

```
main_menu();
```

B. Python

```

import psycopg2
import re
from datetime import datetime
from unicode import unicode # Importamos la libreria para
    eliminar tildes

# Configuracin de la conexin a la base de datos
conn = psycopg2.connect(
    dbname="detector",
    user="postgres",
    password="Dali6478",
    host="localhost",
    port="5432"
)
cursor = conn.cursor()

# Funcin para almacenar en el archivo .txt
def save_to_history(data):
    with open("history.txt", "a") as file:
        file.write(data + "\n")

# Funciones para el detector
def is_palindrome(sentence):
    # Normalizar la oracin: eliminar caracteres no alfabticos,
    # convertir a minsculas y eliminar tildes
    normalized = re.sub(r'[^a-zA-Z]', '',
        unicode(sentence).lower())
    return normalized == normalized[::-1]

def is_prime(number):
    if number <= 1:
        return False
    for i in range(2, int(number ** 0.5) + 1):
        if number % i == 0:
            return False
    return True

def is_perfect(number):
    return number > 1 and sum(i for i in range(1, number) if
        number % i == 0) == number

# Funcin principal del men del detector
def detector_menu():
    while True:
        print("\n1) Detector de palndromos\n2) Detector de nmeros
            primos\n3) Detector de nmeros perfectos")
        option = input("Selecciona una opcin: ")

        if option == '1':
            sentence = input("Ingresa una oracin: ")
            if not re.search(r'[a-zA-Z]', sentence):
                print("Error: Solo se permiten letras en la
                    oracin.")
                continue

            result = "SI" if is_palindrome(sentence) else "NO"
            cursor.execute("INSERT INTO detect (palindromos,
                resultado) VALUES (%s, %s)", (sentence, result))
            conn.commit()
            message = f"La oracin {'SI' if result == 'SI' else
                'NO'} es palndroma"
            print(message)
            save_to_history(f"{datetime.now()} - Palndromo:
                {sentence} - Resultado: {result}")

        elif option == '2':
            try:
                number = int(input("Ingresa un nmero: "))
            except ValueError:
                print("Error: Solo se permiten nmeros.")
                continue

            result = "SI" if is_prime(number) else "NO"

```

```

cursor.execute("INSERT INTO detect (numero, resultado)
VALUES (%s, %s)", (number, result))
conn.commit()
message = f"El nmero {'SI' if result == 'SI' else
'NO'} es primo"
print(message)
save_to_history(f"{datetime.now()} - Nmero primo:
{number} - Resultado: {result}")

elif option == '3':
    try:
        number = int(input("Ingresa un nmero: "))
    except ValueError:
        print("Error: Solo se permiten nmeros.")
        continue

    result = "SI" if is_perfect(number) else "NO"
    cursor.execute("INSERT INTO detect (numero, resultado)
VALUES (%s, %s)", (number, result))
    conn.commit()
    message = f"El nmero {'SI' if result == 'SI' else
'NO'} es perfecto"
    print(message)
    save_to_history(f"{datetime.now()} - Nmero perfecto:
{number} - Resultado: {result}")

else:
    print("Opcin invlida. Intntalo de nuevo.")
    continue

again = input("Deseas realizar otra operacin? (s/n):
").lower()
if again == 's':
    # Si el usuario elige 's', reiniciar el programa
    pidiendo el nombre de usuario
    return False # Esto vuelve al men principal

elif again == 'n':
    # Si el usuario elige 'n', cerrar la conexin y salir
    conn.close()
    conn.close() # Cerramos la conexin antes de salir.
    print("Saliendo del programa...")
    return True # Detiene el programa

# Men principal
def main_menu():
    while True:
        user = input("Ingresa tu nombre de usuario: ")
        cursor.execute("INSERT INTO detect (usuario) VALUES
(%s)", (user,))
        conn.commit()

    while True:
        print("\n1) Detector\n2) Historial de datos
ingresados\n3) Borrar datos\n4) Salir")
        option = input("Selecciona una opcin: ")

        if option == '1':
            if not detector_menu(): # Si el detector retorna
False, se detiene la ejecucin.
                break # Si el detector devuelve True, es que
se termin el proceso.

        elif option == '2':
            cursor.execute("SELECT * FROM detect")
            records = cursor.fetchall()
            for record in records:
                print(record)

        elif option == '3':
            user_to_delete = input("Ingresa el nombre de
usuario cuyos datos deseas borrar: ")
            cursor.execute("DELETE FROM detect WHERE usuario =
%s", (user_to_delete,))
            conn.commit()
            print(f"Datos de {user_to_delete} eliminados.")

        elif option == '4':
            print("Saliendo del programa...")
            cursor.close()
            conn.close() # Cerramos la conexin antes de salir.
            return # Detenemos el programa al seleccionar la
opcin de salir.

        else:
            print("Opcin invlida. Intntalo de nuevo.")

```

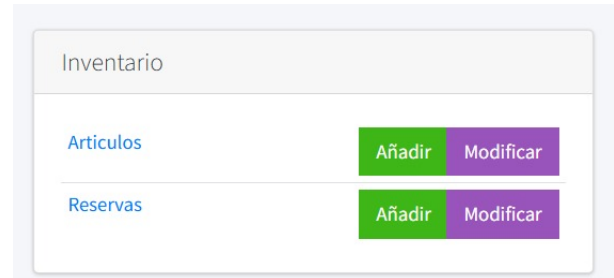
```

# Ejecucin del programa
main_menu()

```

IV. MODULO DE CONTABILIDAD

Figura 1: Django - Inventario



Fuente: Elaboración Propia, 2024.

Figura 2: Django - Historial Contabilidad

fecha reserva

profesor

Buscar

fecha reserva

0 de 6 seleccionado

	Nombre del profesor	Fecha reserva	Hora reserva	Producto	Usuario	
<input type="checkbox"/>	David María	Good Doctor	8 de noviembre de 2024	08:00	Cheques Cardiológicos	mmes12
<input type="checkbox"/>	Johán Pérez	Good Doctor	18 de octubre de 2024	08:00	Vacunación	jroba1
<input type="checkbox"/>	Johán Pérez	Good Doctor	18 de octubre de 2024	08:00	Cheques Preoperator	mmes
<input type="checkbox"/>	José Pérez	Good Doctor	12 de octubre de 2024	08:00	Vacunación	mmes12
<input type="checkbox"/>	David Pérez	Good Doctor	14 de octubre de 2024	09:00	Cheques Cardiológicos	mmes12
<input type="checkbox"/>	David María	Good Doctor	14 de octubre de 2024	08:00	Orientación Nutricional	mmes12

Entrenapas

Fuente: Elaboración Propia, 2024.

A. Link de GitHub

<https://github.com/dvd-r16/proyectos>

B. Link de Drive - Videos

https://drive.google.com/drive/folders/1YIWUz2xakxxgVof48AQozlCIyDIHpfXA?usp=drive_link

V. CONCLUSIÓN

1. Este código implementa un sistema de detección multifuncional que permite a los usuarios identificar palíndromos, números primos y números perfectos. Utiliza Python junto con una base de datos PostgreSQL para almacenar registros y un archivo de texto para crear un historial de las operaciones realizadas.

El programa presenta una estructura de menús intuitiva. En el menú principal, el usuario puede ingresar su nombre, acceder al detector, visualizar el historial de operaciones almacenado o eliminar datos específicos de usuarios. Dentro del submenú del detector, el usuario puede seleccionar entre tres funcionalidades: verificar si una oración es un palíndromo, si un número es primo o

si es perfecto. Cada operación almacena los resultados tanto en la base de datos como en un archivo de texto para consulta futura.

Al finalizar, el programa garantiza que la conexión a

la base de datos se cierra de manera segura. En general, este código combina técnicas de manipulación de cadenas, cálculos numéricos, manejo de base de datos y control de archivos, proporcionando una aplicación robusta y persistente para la gestión de datos ingresados.

[1] Ing. Jose Anibal Silva de los Angeles. Proyectos Aplicados para I.E. *Programa del curso*. Ciudad de Guatemala:

Universidad de San Carlos, Facultad de Ingeniería, Escuela de Ingeniería Mecánica Eléctrica.
