[[1]](#footnote-1)

Examen Corto #1

Francisco Alberto, Navarro Orozco, 202004752

*Escuela de Mecánica Eléctrica, Facultad de Ingeniería, Universidad de San Carlos de Guatemala*

*Para el examen corto se realizaron una serie de diversos programas en octave y se conectaron con Postgres*

# **Programa 1**

pkg load database

% Establecer la conexión con la base de datos PostgreSQL

conn = pq\_connect(setdbopts('dbname', '0980 Proyectos', 'host', 'localhost', 'port', '5433', 'user', 'postgres', 'password', '2405'));

try

% Solicitar al usuario que ingrese tres números

primer\_numero = input("Ingrese el primer número: ");

segundo\_numero = input("Ingrese el segundo número: ");

tercer\_numero = input("Ingrese el tercer número: ");

% Verificar y realizar las operaciones según las condiciones especificadas

if primer\_numero > segundo\_numero && primer\_numero > tercer\_numero

resultado = primer\_numero + segundo\_numero + tercer\_numero;

fprintf("El resultado es la suma de los tres números: %d\n", resultado);

elseif segundo\_numero > primer\_numero && segundo\_numero > tercer\_numero

resultado = primer\_numero \* segundo\_numero \* tercer\_numero;

fprintf("El resultado es la multiplicación de los tres números: %d\n", resultado);

elseif tercer\_numero > primer\_numero && tercer\_numero > segundo\_numero

resultado = strcat(num2str(primer\_numero), num2str(segundo\_numero), num2str(tercer\_numero));

fprintf("El resultado es la concatenación de los tres números: %s\n", resultado);

elseif primer\_numero == segundo\_numero

fprintf("El número diferente es: %d\n", tercer\_numero);

elseif primer\_numero == tercer\_numero

fprintf("El número diferente es: %d\n", segundo\_numero);

elseif segundo\_numero == tercer\_numero

fprintf("El número diferente es: %d\n", primer\_numero);

else

fprintf("Todos los números son iguales: %d\n", primer\_numero);

end

% Insertar los resultados en la tabla "Uno" de la base de datos

Ins1 = 'INSERT INTO Uno (Primer\_Numero, Segundo\_Numero, Tercer\_Numero) VALUES (';

Ins2 = ');';

Instruccion = strcat(Ins1, num2str(primer\_numero), ', ', num2str(segundo\_numero), ', ', num2str(tercer\_numero), Ins2);

Registro = pq\_exec\_params(conn, Instruccion);

catch e

disp(['Error durante la conexión a la DB, Consulte el error: ' e.message]);

end

% Cerrar la conexión con la base de datos

close(conn);

**Programa 2**

% Cargar el paquete de base de datos

pkg load database

% Establecer la conexión con la base de datos PostgreSQL

conn = pq\_connect(setdbopts('dbname', '0980 Proyectos', 'host', 'localhost', 'port', '5433', 'user', 'postgres', 'password', '2405'));

% Solicitar al usuario que ingrese un número entero

numero\_ingresado = input("Ingrese un número entero: ");

% Calcular los divisores del número ingresado

divisores = find(mod(numero\_ingresado, 1:numero\_ingresado) == 0);

% Mostrar los divisores en la consola

fprintf("Los divisores de %d son: %s\n", numero\_ingresado, num2str(divisores));

% Contar los divisores en total

total\_divisores = numel(divisores);

% Mostrar el número total de divisores

fprintf("El número total de divisores de %d es: %d\n", numero\_ingresado, total\_divisores);

% Insertar el número ingresado y el total de divisores en la tabla "Dos" de la base de datos

try

Ins1 = 'INSERT INTO Dos (numero\_ingresado, Total\_de\_Divisores) VALUES (';

Ins2 = ');';

Instruccion = strcat(Ins1, num2str(numero\_ingresado), ', ', num2str(total\_divisores), Ins2);

Registro = pq\_exec\_params(conn, Instruccion);

catch e

disp(['Error durante la conexión a la DB, Consulte el error: ' e.message]);

end

% Cerrar la conexión con la base de datos

close(conn);

**Programa 3**

% Cargar el paquete de base de datos

pkg load database

% Establecer la conexión con la base de datos PostgreSQL

conn = pq\_connect(setdbopts('dbname', '0980 Proyectos', 'host', 'localhost', 'port', '5433', 'user', 'postgres', 'password', '2405'));

% Solicitar al usuario que ingrese una palabra

palabra\_ingresada = input("Ingrese una palabra: ", 's');

% Contar el número de vocales en la palabra

numero\_vocales = sum(ismember(lower(palabra\_ingresada), ['a', 'e', 'i', 'o', 'u']));

% Mostrar el resultado en la consola

fprintf("La palabra '%s' tiene %d vocales.\n", palabra\_ingresada, numero\_vocales);

% Insertar los resultados en la tabla "Tres" de la base de datos

try

Ins1 = 'INSERT INTO Tres (Palabra\_Ingresada, numero\_vocales) VALUES (''';

Ins2 = ''', ';

Ins3 = ');';

Instruccion = strcat(Ins1, palabra\_ingresada, Ins2, num2str(numero\_vocales), Ins3);

Registro = pq\_exec\_params(conn, Instruccion);

catch e

disp(['Error durante la conexión a la DB, Consulte el error: ' e.message]);

end

% Cerrar la conexión con la base de datos

close(conn);

**Programa 4**

% Cargar el paquete de base de datos

pkg load database

% Establecer la conexión con la base de datos PostgreSQL

conn = pq\_connect(setdbopts('dbname', '0980 Proyectos', 'host', 'localhost', 'port', '5433', 'user', 'postgres', 'password', '2405'));

% Solicitar al usuario que ingrese un número

numero\_ingresado = input("Ingrese un número: ");

% Calcular la suma de los números desde 0 hasta el número ingresado

resultado\_suma = sum(0:numero\_ingresado);

% Mostrar el resultado de la suma en la consola

fprintf("La suma de los números desde 0 hasta %d es: %d\n", numero\_ingresado, resultado\_suma);

% Insertar el número ingresado y el resultado de la suma en la tabla "Cuatro" de la base de datos

try

Ins1 = 'INSERT INTO Cuatro (Numero\_Ingresado, Resultado\_Suma) VALUES (';

Ins2 = ');';

Instruccion = strcat(Ins1, num2str(numero\_ingresado), ', ', num2str(resultado\_suma), Ins2);

Registro = pq\_exec\_params(conn, Instruccion);

catch e

disp(['Error durante la conexión a la DB, Consulte el error: ' e.message]);

end

% Cerrar la conexión con la base de datos

close(conn);

**Programa 5**

% Cargar el paquete de base de datos

pkg load database

% Establecer la conexión con la base de datos PostgreSQL

conn = pq\_connect(setdbopts('dbname', '0980 Proyectos', 'host', 'localhost', 'port', '5433', 'user', 'postgres', 'password', '2405'));

% Solicitar al usuario que ingrese un número de inicio

inicio = input("Ingrese un número de inicio: ");

% Solicitar al usuario que ingrese un número de fin

fin = input("Ingrese un número de fin: ");

% Mostrar los números de dos en dos desde inicio hasta fin

fprintf("Números de dos en dos desde %d hasta %d:\n", inicio, fin);

for i = inicio:2:fin

fprintf("%d\n", i);

end

% Insertar el número de inicio y el de fin en la tabla "Cinco" de la base de datos

try

Ins1 = 'INSERT INTO Cinco (Inicio, Fin) VALUES (';

Ins2 = ');';

Instruccion = strcat(Ins1, num2str(inicio), ', ', num2str(fin), Ins2);

Registro = pq\_exec\_params(conn, Instruccion);

catch e

disp(['Error durante la conexión a la DB, Consulte el error: ' e.message]);

end

% Cerrar la conexión con la base de datos

close(conn);

**Programa 6**

% Cargar el paquete de base de datos

pkg load database

% Establecer la conexión con la base de datos PostgreSQL

conn = pq\_connect(setdbopts('dbname', '0980 Proyectos', 'host', 'localhost', 'port', '5433', 'user', 'postgres', 'password', '2405'));

% Solicitar al usuario que ingrese el primer número

primer\_numero = input("Ingrese el primer número: ");

% Solicitar al usuario que ingrese el segundo número

segundo\_numero = input("Ingrese el segundo número: ");

% Determinar cuál es el número mayor

numero\_mayor = max(primer\_numero, segundo\_numero);

% Mostrar la lista de números desde el mayor hasta el menor

fprintf("Lista de números desde el mayor hasta el menor:\n");

for i = numero\_mayor:-1:min(primer\_numero, segundo\_numero)

fprintf("%d\n", i);

end

% Insertar los números ingresados y el número mayor en la tabla "Seis" de la base de datos

try

Ins1 = 'INSERT INTO Seis (Primer\_Numero, Segundo\_Numero, Numero\_Mayor) VALUES (';

Ins2 = ');';

Instruccion = strcat(Ins1, num2str(primer\_numero), ', ', num2str(segundo\_numero), ', ', num2str(numero\_mayor), Ins2);

Registro = pq\_exec\_params(conn, Instruccion);

catch e

disp(['Error durante la conexión a la DB, Consulte el error: ' e.message]);

end

% Cerrar la conexión con la base de datos

close(conn);

**Programa 7**

% Cargar el paquete de base de datos

pkg load database

% Establecer la conexión con la base de datos PostgreSQL

conn = pq\_connect(setdbopts('dbname', '0980 Proyectos', 'host', 'localhost', 'port', '5433', 'user', 'postgres', 'password', '2405'));

% Solicitar al usuario que ingrese una palabra

palabra\_ingresada = input("Ingrese una palabra: ", 's');

% Inicializar contadores para cada vocal

contador\_a = 0;

contador\_e = 0;

contador\_i = 0;

contador\_o = 0;

contador\_u = 0;

% Contar el número de veces que aparece cada vocal en la palabra

for letra = palabra\_ingresada

letra = lower(letra);

switch letra

case 'a'

contador\_a = contador\_a + 1;

case 'e'

contador\_e = contador\_e + 1;

case 'i'

contador\_i = contador\_i + 1;

case 'o'

contador\_o = contador\_o + 1;

case 'u'

contador\_u = contador\_u + 1;

end

end

% Mostrar el resultado en la consola

fprintf("A=%d, E=%d, I=%d, O=%d, U=%d.\n", contador\_a, contador\_e, contador\_i, contador\_o, contador\_u);

% Insertar los resultados en la tabla "Siete" de la base de datos

try

Ins1 = 'INSERT INTO Siete (Palabra, Letra\_A, Letra\_E, Letra\_I, Letra\_O, Letra\_U) VALUES (''';

Ins2 = ''', ';

Ins3 = ');';

Instruccion = strcat(Ins1, palabra\_ingresada, Ins2, num2str(contador\_a), ', ', num2str(contador\_e), ', ', num2str(contador\_i), ', ', num2str(contador\_o), ', ', num2str(contador\_u), Ins3);

Registro = pq\_exec\_params(conn, Instruccion);

catch e

disp(['Error durante la conexión a la DB, Consulte el error: ' e.message]);

end

% Cerrar la conexión con la base de datos

close(conn);

**Programa 8**

% Cargar el paquete de base de datos

pkg load database

% Establecer la conexión con la base de datos PostgreSQL

conn = pq\_connect(setdbopts('dbname', '0980 Proyectos', 'host', 'localhost', 'port', '5433', 'user', 'postgres', 'password', '2405'));

try

% Mostrar todos los números impares del 1 al 100

fprintf("Números impares del 1 al 100:\n");

total\_impares = 0;

for num = 1:2:100

fprintf("%d ", num);

total\_impares = total\_impares + 1;

end

% Mostrar el total de números impares

fprintf("\nEl total de números impares es: %d\n", total\_impares);

% Insertar el total de números impares en la tabla "Ocho" de la base de datos

Ins1 = 'INSERT INTO Ocho (Total\_numeros\_impares) VALUES (';

Ins2 = ');';

Instruccion = strcat(Ins1, num2str(total\_impares), Ins2);

Registro = pq\_exec\_params(conn, Instruccion);

catch e

disp(['Error durante la conexión a la DB, Consulte el error: ' e.message]);

end

% Cerrar la conexión con la base de datos

close(conn);

**Programa 9**

% Cargar el paquete de base de datos

pkg load database

% Establecer la conexión con la base de datos PostgreSQL

conn = pq\_connect(setdbopts('dbname', '0980 Proyectos', 'host', 'localhost', 'port', '5433', 'user', 'postgres', 'password', '2405'));

try

% Solicitar al usuario que ingrese tres números enteros positivos

lado1 = input("Ingrese el primer lado del triángulo: ");

lado2 = input("Ingrese el segundo lado del triángulo: ");

lado3 = input("Ingrese el tercer lado del triángulo: ");

% Verificar el tipo de triángulo

if lado1 == lado2 && lado2 == lado3

tipo\_triangulo = 'Equilátero';

elseif lado1 == lado2 || lado2 == lado3 || lado1 == lado3

tipo\_triangulo = 'Isósceles';

else

tipo\_triangulo = 'Escaleno';

end

% Mostrar el tipo de triángulo en la consola

fprintf("El triángulo con lados %d, %d y %d es %s.\n", lado1, lado2, lado3, tipo\_triangulo);

% Insertar los resultados en la tabla "Nueve" de la base de datos

Ins1 = 'INSERT INTO Nueve (Primer\_Lado, Segundo\_Lado, Tercer\_Lado, Tipo\_De\_Triangulo) VALUES (';

Ins2 = ');';

Instruccion = strcat(Ins1, num2str(lado1), ', ', num2str(lado2), ', ', num2str(lado3), ', ''', tipo\_triangulo, '''', Ins2);

Registro = pq\_exec\_params(conn, Instruccion);

catch e

disp(['Error durante la conexión a la DB, Consulte el error: ' e.message]);

end

% Cerrar la conexión con la base de datos

close(conn);

**Programa 10**

% Cargar el paquete de base de datos

pkg load database

% Establecer la conexión con la base de datos PostgreSQL

conn = pq\_connect(setdbopts('dbname', '0980 Proyectos', 'host', 'localhost', 'port', '5433', 'user', 'postgres', 'password', '2405'));

try

% Solicitar al usuario que ingrese un número

numero\_ingresado = input("Ingrese un número: ");

% Verificar si el número es divisible por 7

if mod(numero\_ingresado, 7) == 0

% Calcular el factorial del número

factorial\_resultado = factorial(numero\_ingresado);

fprintf("El factorial de %d es %d.\n", numero\_ingresado, factorial\_resultado);

resultado\_mensaje = num2str(factorial\_resultado);

else

% Mostrar mensaje de número incorrecto

disp("Número incorrecto.");

resultado\_mensaje = "Número incorrecto";

end

% Insertar los resultados en la tabla "Diez" de la base de datos

Ins1 = 'INSERT INTO Diez (Numero\_Ingresado, Resultado) VALUES (';

Ins2 = ');';

Instruccion = strcat(Ins1, num2str(numero\_ingresado), ', ''', resultado\_mensaje, '''', Ins2);

Registro = pq\_exec\_params(conn, Instruccion);

catch e

disp(['Error durante la conexión a la DB, Consulte el error: ' e.message]);

end

% Cerrar la conexión con la base de datos

close(conn);

**Programa 11**

% Cargar el paquete de base de datos

pkg load database

% Establecer la conexión con la base de datos PostgreSQL

conn = pq\_connect(setdbopts('dbname', '0980 Proyectos', 'host', 'localhost', 'port', '5433', 'user', 'postgres', 'password', '2405'));

try

% Despliega el menú

fprintf("Seleccione la figura para calcular el área:\n");

fprintf("1. Circulo\n");

fprintf("2. Triangulo\n");

fprintf("3. Cuadrado\n");

fprintf("4. Rectangulo\n");

% Solicita la opción al usuario

opcion = input("Ingrese el número de la figura: ");

% Realiza el cálculo del área según la opción

switch opcion

case 1 % Circulo

radio = input("Ingrese el radio del circulo: ");

area = pi \* radio^2;

figura = 'Circulo';

case 2 % Triangulo

base = input("Ingrese la base del triangulo: ");

altura = input("Ingrese la altura del triangulo: ");

area = 0.5 \* base \* altura;

figura = 'Triangulo';

case 3 % Cuadrado

lado = input("Ingrese el lado del cuadrado: ");

area = lado^2;

figura = 'Cuadrado';

case 4 % Rectangulo

base = input("Ingrese la base del rectangulo: ");

altura = input("Ingrese la altura del rectangulo: ");

area = base \* altura;

figura = 'Rectangulo';

otherwise

error("Opción no válida");

end

% Muestra el resultado del cálculo del área

fprintf("El área del %s es: %.2f\n", figura, area);

% Inserta los resultados en la tabla "Once" de la base de datos

Ins1 = 'INSERT INTO Once (Opcion\_Elegida, Figura, Area) VALUES (';

Ins2 = ');';

Instruccion = strcat(Ins1, num2str(opcion), ', ''', figura, ''', ', num2str(area), Ins2);

Registro = pq\_exec\_params(conn, Instruccion);

catch e

disp(['Error durante la conexión a la DB, Consulte el error: ' e.message]);

end

% Cerrar la conexión con la base de datos

close(conn);

**Programa 12**

% Cargar el paquete de base de datos

pkg load database

% Establecer la conexión con la base de datos PostgreSQL

conn = pq\_connect(setdbopts('dbname', '0980 Proyectos', 'host', 'localhost', 'port', '5433', 'user', 'postgres', 'password', '2405'));

try

% Solicitar al usuario ingresar 3 notas

nota1 = input("Ingrese la primera nota: ");

nota2 = input("Ingrese la segunda nota: ");

nota3 = input("Ingrese la tercera nota: ");

% Calcular el promedio

promedio = (nota1 + nota2 + nota3) / 3;

% Mostrar el promedio

fprintf("El promedio es: %.2f\n", promedio);

% Determinar si está aprobado o desaprobado

mensaje = '';

if promedio > 60

mensaje = 'Aprobado';

else

mensaje = 'Desaprobado';

end

% Mostrar el mensaje

fprintf("Mensaje: %s\n", mensaje);

% Insertar los resultados en la tabla "Doce" de la base de datos

Ins1 = 'INSERT INTO Doce (Primer\_Nota, Segunda\_Nota, Tercera\_Nota, Promedio, Mensaje) VALUES (';

Ins2 = strcat(num2str(nota1), ', ', num2str(nota2), ', ', num2str(nota3), ', ', num2str(promedio), ', ''', mensaje, ''');');

Instruccion = strcat(Ins1, Ins2);

Registro = pq\_exec\_params(conn, Instruccion);

catch e

disp(['Error durante la conexión a la DB, Consulte el error: ' e.message]);

end

% Cerrar la conexión con la base de datos

close(conn);

**Programa 13**

% Cargar el paquete de base de datos

pkg load database

% Establecer la conexión con la base de datos PostgreSQL

conn = pq\_connect(setdbopts('dbname', '0980 Proyectos', 'host', 'localhost', 'port', '5433', 'user', 'postgres', 'password', '2405'));

% Solicitar al usuario que ingrese su año de nacimiento

ano\_nacimiento = input("Ingrese su año de nacimiento: ");

% Verificar si el año ingresado es bisiesto

if mod(ano\_nacimiento, 4) == 0 && (mod(ano\_nacimiento, 100) ~= 0 || mod(ano\_nacimiento, 400) == 0)

mensaje = "El año es bisiesto.";

else

mensaje = "El año no es bisiesto.";

end

% Mostrar el resultado en la consola

fprintf('%s\n', mensaje);

% Insertar los resultados en la tabla "Trece" de la base de datos

try

Ins1 = 'INSERT INTO Trece (Año\_Ingresado, Tipo\_De\_año) VALUES (';

Ins2 = ', ''';

Ins3 = ''');';

Instruccion = strcat(Ins1, num2str(ano\_nacimiento), Ins2, mensaje, Ins3);

Registro = pq\_exec\_params(conn, Instruccion);

catch e

disp(['Error durante la conexión a la DB, Consulte el error: ' e.message]);

end

% Cerrar la conexión con la base de datos

close(conn);

**Programa 14**

% Cargar el paquete de base de datos

pkg load database

% Establecer la conexión con la base de datos PostgreSQL

conn = pq\_connect(setdbopts('dbname', '0980 Proyectos', 'host', 'localhost', 'port', '5433', 'user', 'postgres', 'password', '2405'));

try

% Solicitar al usuario que ingrese el modelo del auto y el kilometraje

modelo\_auto = input("Ingrese el modelo del auto (año): ");

kilometraje\_auto = input("Ingrese el kilometraje del auto en km: ");

% Calcular el diagnóstico según las condiciones especificadas

if modelo\_auto < 2007 && kilometraje\_auto > 20000.0

diagnostico = "Renovarse";

elseif modelo\_auto >= 2007 && modelo\_auto <= 2013 && kilometraje\_auto > 20000

diagnostico = "Debe recibir mantenimiento";

elseif modelo\_auto > 2013 && kilometraje\_auto < 10000

diagnostico = "En óptimas condiciones";

else

diagnostico = "Mecánico";

end

% Mostrar el diagnóstico en la consola

fprintf("El diagnóstico del carro es: %s\n", diagnostico);

% Insertar los resultados en la tabla "Catorce" de la base de datos

Instruccion = sprintf("INSERT INTO Catorce (modelo, kilometraje\_recorrido, diagnostico) VALUES (%d, %f, '%s');", modelo\_auto, kilometraje\_auto, diagnostico);

Registro = pq\_exec\_params(conn, Instruccion);

catch e

disp(['Error durante la conexión a la DB, Consulte el error: ' e.message]);

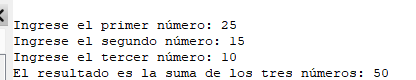
end

% Cerrar la conexión con la base de datos

close(conn);

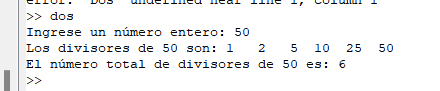
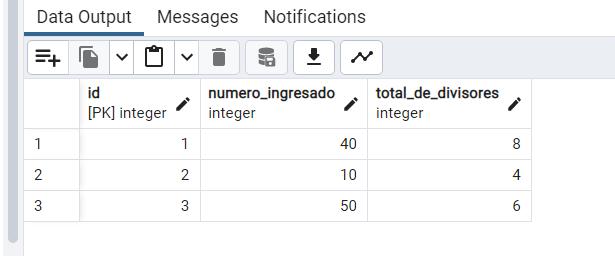
# Resultados

# **Programa 1**

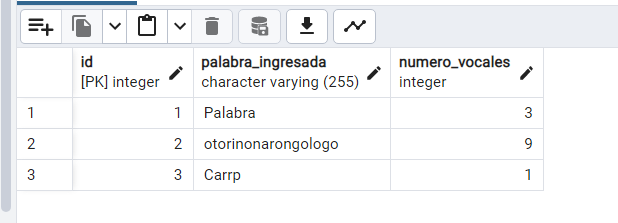




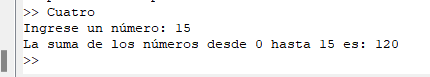
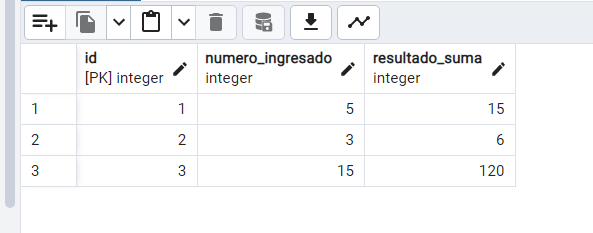
**Programa 2**

**** ****

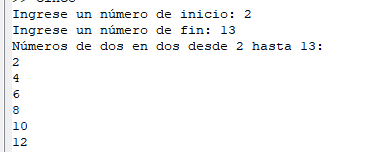
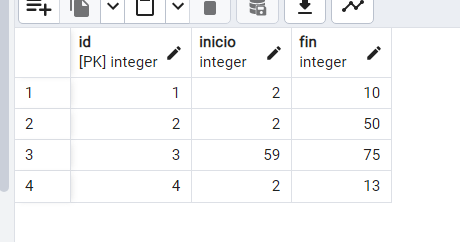
**Programa 3**

**** ****

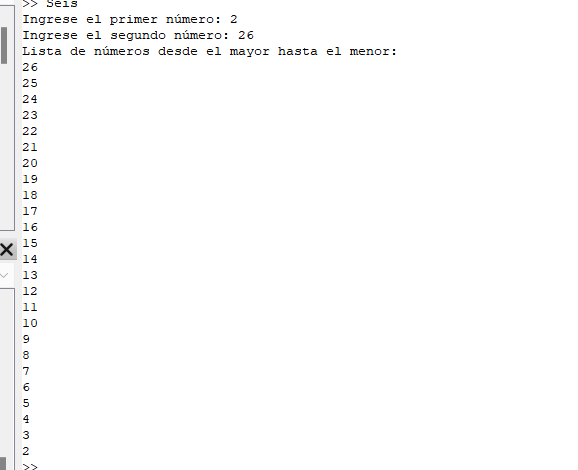
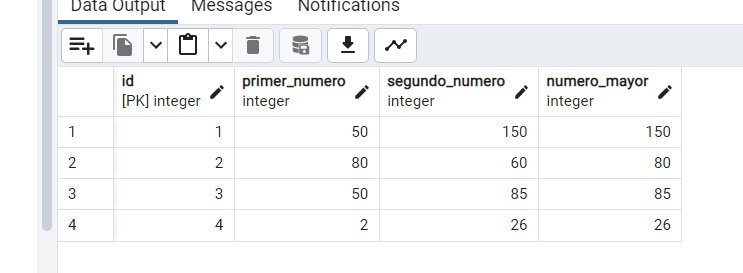
**Programa 4**

**** ****

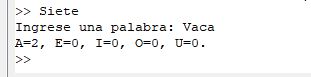
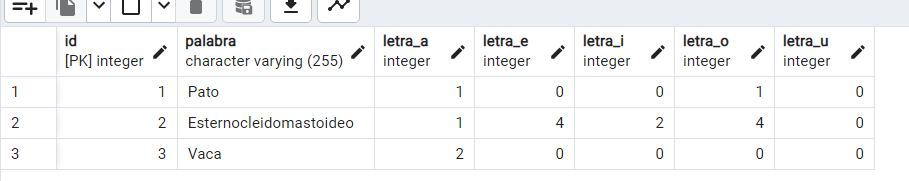
**Programa 5**

**** ****

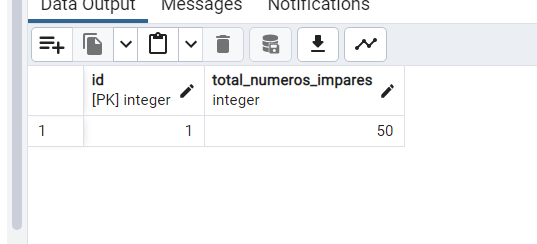
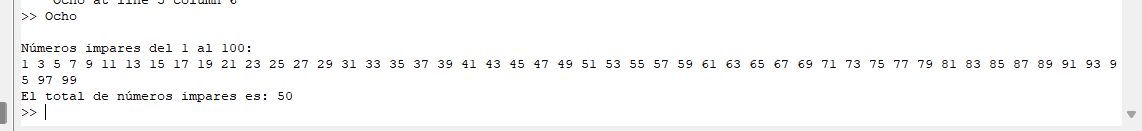
**Programa 6**

**** ****

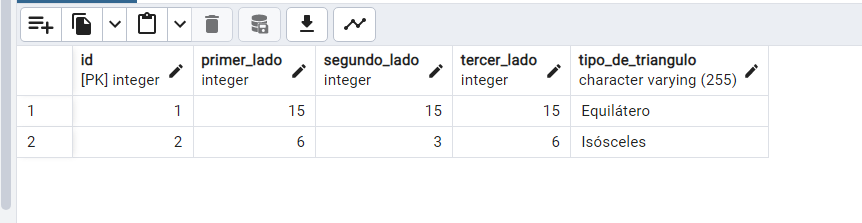
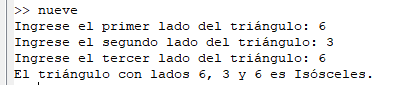
**Programa 7**

**** ****

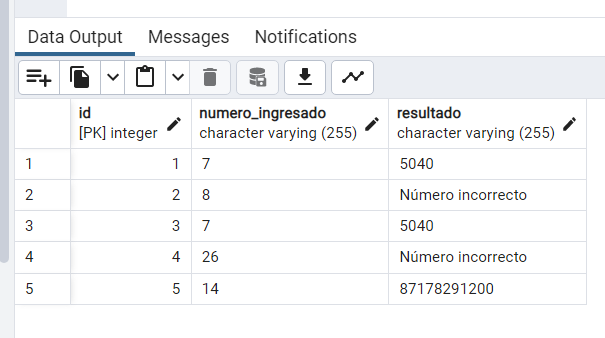
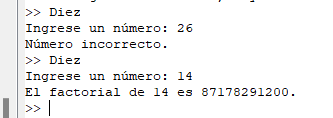
**Programa 8**

****

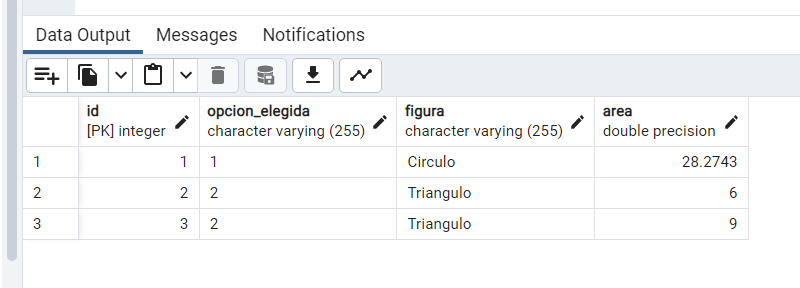
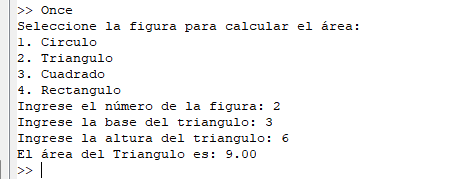
**Programa 9**

****

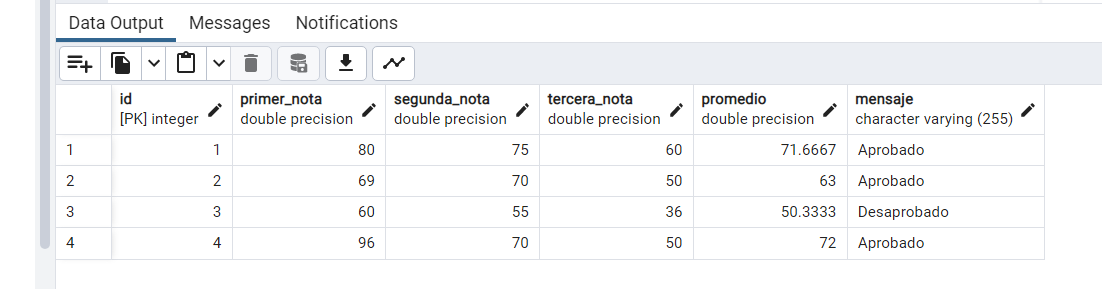
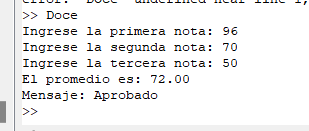
**Programa 10**

****

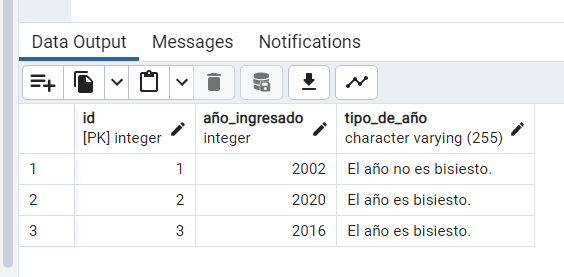
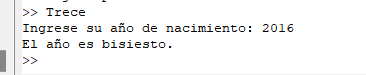
**Programa 11**

****

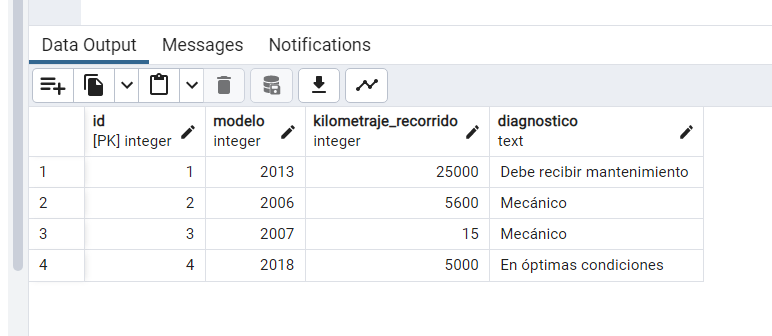
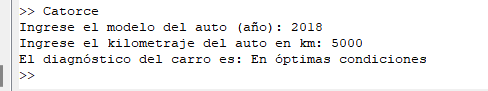
**Programa 12**

****

**Programa 13**

****

**Programa 14**



1. [↑](#footnote-ref-1)