

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'));

% Mostrar todos los números impares del 1 al 100
    try
        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_triángulo = 'Equilátero';
elseif lado1 == lado2 || lado2 == lado3 || lado1 == lado3
    tipo_triángulo = 'Isósceles';
else
    tipo_triángulo = '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_triángulo);

% 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_triángulo, '"',
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
    end

```

```

    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

```
Ingrese el primer número: 25
Ingrese el segundo número: 15
Ingrese el tercer número: 10
El resultado es la suma de los tres números: 50
```

	id [PK] integer	primer_numero integer	segundo_numero integer	tercer_numero integer
1	1	25	15	10

Programa 2

```
Error: DOS undefined near line 1, column 1
>> dos
Ingrese un número entero: 50
Los divisores de 50 son: 1 2 5 10 25 50
El número total de divisores de 50 es: 6
>>
```

	id [PK] integer	numero_ingresado integer	total_de_divisores integer
1	1	40	8
2	2	10	4
3	3	50	6

Programa 3

```
>> tres
Ingrese una palabra: Carrp
La palabra 'Carrp' tiene 1 vocales.
>>
```

	id [PK] integer	palabra_ingresada character varying (255)	numero_vocales integer
1	1	Palabra	3
2	2	otorinonarangologo	9
3	3	Carrp	1

Programa 4

```
>> Cuatro
Ingrese un número: 15
La suma de los números desde 0 hasta 15 es: 120
>>
```

	id [PK] integer	numero_ingresado integer	resultado_suma integer
1	1	5	15
2	2	3	6
3	3	15	120

Programa 5

```
>> cinco
Ingrese un número de inicio: 2
Ingrese un número de fin: 13
Números de dos en dos desde 2 hasta 13:
2
4
6
8
10
12
```

	id [PK] integer	inicio integer	fin integer
1	1	2	10
2	2	2	50
3	3	59	75
4	4	2	13

Programa 6

```
>> Seis
Ingrese el primer número: 2
Ingrese el segundo número: 26
Lista de números desde el mayor hasta el menor:
26
25
24
23
22
21
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
>>
```

Data Output Messages Notifications

	id [PK] integer	primer_numero integer	segundo_numero integer	numero_mayor integer
1	1	50	150	150
2	2	80	60	80
3	3	50	85	85
4	4	2	26	26

Programa 7

```
>> Siete
Ingrese una palabra: Vaca
A=2, E=0, I=0, O=0, U=0.
>>
```

	id [PK] integer	palabra character varying (255)	letra_a integer	letra_e integer	letra_i integer	letra_o integer	letra_u integer
1	1	Pato	1	0	0	1	0
2	2	Esternocleidomastoideo	1	4	2	4	0
3	3	Vaca	2	0	0	0	0

Programa 8

```
>> Ocho
Números impares del 1 al 100:
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49 51 53 55 57 59 61 63 65 67 69 71 73 75 77 79 81 83 85 87 89 91 93 95 97 99
El total de números impares es: 50
>>
```

Data Output Messages Notifications

	id [PK] integer	total_numeros_impares integer
1	1	50

Programa 9

```
>> nueve
Ingrese el primer lado del triángulo: 6
Ingrese el segundo lado del triángulo: 3
Ingrese el tercer lado del triángulo: 6
El triángulo con lados 6, 3 y 6 es Isósceles.
```

	id [PK] integer	primer_lado integer	segundo_lado integer	tercer_lado integer	tipo_de_triángulo character varying (255)
1	1	15	15	15	Equilátero
2	2	6	3	6	Isósceles

Programa 10

```
>> Diez
Ingrese un número: 26
Número incorrecto.
>> Diez
Ingrese un número: 14
El factorial de 14 es 87178291200.
>>
```

Data Output Messages Notifications

	id [PK] integer	numero_ingresado character varying (255)	resultado character varying (255)
1	1	7	5040
2	2	8	Número incorrecto
3	3	7	5040
4	4	26	Número incorrecto
5	5	14	87178291200

Programa 11

```
>> Once
Seleccione la figura para calcular el área:
1. Circulo
2. Triangulo
3. Cuadrado
4. Rectangulo
Ingrese el número de la figura: 2
Ingrese la base del triángulo: 3
Ingrese la altura del triángulo: 6
El área del Triangulo es: 9.00
>>
```

Data Output Messages Notifications

	id [PK] integer	opcion_elegida character varying (255)	figura character varying (255)	area double precision
1	1	1	Circulo	28.2743
2	2	2	Triangulo	6
3	3	2	Triangulo	9

Programa 12

```

>> Doce
Ingrese la primera nota: 96
Ingrese la segunda nota: 70
Ingrese la tercera nota: 50
El promedio es: 72.00
Mensaje: Aprobado
>>

```

Data Output Messages Notifications						
	id [PK] integer	primer_nota double precision	segunda_nota double precision	tercera_nota double precision	promedio double precision	mensaje character varying (255)
1	1	96	70	50	71.6667	Aprobado
2	2	69	70	50	63	Aprobado
3	3	60	55	36	50.3333	Desaprobado
4	4	96	70	50	72	Aprobado

Programa 13

```

>> Trece
Ingrese su año de nacimiento: 2016
El año es bisiesto.
>>

```

Data Output Messages Notifications			
	id [PK] integer	año_ingresado integer	tipo_de_año character varying (255)
1	1	2002	El año no es bisiesto.
2	2	2020	El año es bisiesto.
3	3	2016	El año es bisiesto.

Programa 14

```

>> Catorce
Ingrese el modelo del auto (año): 2018
Ingrese el kilometraje del auto en km: 5000
El diagnóstico del carro es: En óptimas condiciones
>>

```

Data Output Messages Notifications				
	id [PK] integer	modelo integer	kilometraje_recorrido integer	diagnostico text
1	1	2013	25000	Debe recibir mantenimiento
2	2	2006	5600	Mecánico
3	3	2007	15	Mecánico
4	4	2018	5000	En óptimas condiciones