

Nombre: Edgar David Barrios Franco.      Carné: 201906465.

Serie I.

Responda las siguientes preguntas de forma manuscrita:

1. ¿Qué función de Octave se utiliza para crear una matriz identidad?  
a) zeros()  
b) ones()  
c) eye()  
d) rand()
2. ¿Qué comando se utiliza para guardar variables en un archivo en Octave?  
a) save()  
b) load()  
c) delete()  
d) clear()
3. ¿Cuál es la sintaxis correcta para acceder al segundo elemento de un vector llamado "x" en Octave?  
a) x(2)  
b) x[2]  
c) x{2}  
d) x(1,2)
4. ¿Qué comando se utiliza para calcular la raíz cuadrada de un número en Octave?  
a) sqrt()  
b) log()  
c) exp()  
d) sin()
5. ¿Cuál es el resultado de la operación 3^2 en Octave?  
a) 6  
b) 9  
c) 12  
d) 6^2
6. ¿Qué función se utiliza para redondear un número en Octave?  
a) round()  
b) floor()  
c) ceil()  
d) all()
7. ¿Cuál es la sintaxis correcta para crear una matriz de 3 filas y 4 columnas en Octave?  
a) A = [3,4]  
b) A = [3;4]  
c) A = zeros(3,4)  
d) A = ones(3,4)
8. ¿Qué comando se utiliza para calcular la media de un vector en Octave?  
a) mean()  
b) median()  
c) mode()  
d) std()
9. ¿Cuál es el resultado de la operación "Hello " . "world" en Octave?  
a) "Hello world"  
b) "Hello"  
c) "world"  
d) "Hello world"
10. ¿Qué función se utiliza para calcular la autocorrelación de un vector en Octave?  
a) corr()  
b) xcorr()  
c) acorr()  
d) autocorr()
11. ¿Qué comando se utiliza para asignar un valor a una variable en Octave?  
a) =  
b) :  
c) ==  
d) :=
12. ¿Cuál de los siguientes comandos permite conocer las dimensiones de una matriz en Octave?  
a) size()  
b) dim()  
c) length()  
d) ndims()
13. ¿Cuál es el comando para generar una secuencia de números equidistantes en Octave?  
a) arange()  
b) linspace()  
c) seq()  
d) range()
14. ¿Qué comando se utiliza para calcular la media de una matriz en Octave?  
a) mean()  
b) median()  
c) mode()  
d) average()
15. ¿Qué comando se utiliza para calcular la desviación estándar de una matriz en Octave?  
a) std()  
b) var()  
c) cov()  
d) corr()
16. ¿Qué comando se utiliza para redondear un número al entero más cercano en Octave?  
a) round()

- b) floor()  
c) ceil()  
d) fix()
17. ¿Qué comando se utiliza para graficar una función en Octave?  
a) **plot()**  
b) graph()  
c) draw()  
d) chart()
18. ¿Cuál es el comando para calcular la transformada de Fourier de una señal en Octave?  
a) dft()  
b) **fft()**  
c) ifft()  
d) fft2()
19. ¿Qué comando se utiliza para leer un archivo de texto en Octave?  
a) readfile()  
b) load()  
c) readtable()  
d) **textread()**
20. ¿Cuál es el comando para guardar una variable en un archivo en Octave?  
a) **save()**  
b) store()  
c) write()  
d) dump()
21. ¿Cuál es el comando para conectarse a una base de datos en PostgreSQL?  
a) connect  
b) use  
c) connect to  
d) **\connect**
22. ¿Cuál es la herramienta de línea de comandos para interactuar con PostgreSQL?  
a) pgAdmin  
b) phpPgAdmin  
c) **psql**  
d) Postico
23. ¿Qué comando se utiliza para crear una nueva base de datos en PostgreSQL?  
a) CREATE TABLE  
b) **CREATE DATABASE**  
c) CREATE SCHEMA  
d) CREATE VIEW
24. ¿Cuál es el comando para listar todas las bases de datos en PostgreSQL?  
a) LIST DATABASES  
b) SHOW DATABASES  
c) **\l**  
d) SELECT DATABASES
25. ¿Cuál es el tipo de dato en PostgreSQL que almacena valores numéricos de alta precisión?  
a) int  
b) float  
c) **decimal**  
d) double precision
26. ¿Qué comando se utiliza para crear una tabla en PostgreSQL?  
a) **CREATE TABLE**  
b) CREATE DATABASE  
c) CREATE SCHEMA  
d) CREATE VIEW
27. ¿Cuál es el comando para borrar una tabla en PostgreSQL?  
a) DELETE TABLE  
b) **DROP TABLE**  
c) REMOVE TABLE  
d) DESTROY TABLE
28. ¿Qué comando se utiliza para insertar datos en una tabla en PostgreSQL?  
a) ADD  
b) UPDATE  
c) **INSERT INTO**  
d) SET
29. ¿Cuál es la sintaxis correcta para hacer una consulta en PostgreSQL que seleccione todas las columnas de una tabla llamada "users"?  
a) SELECT ALL FROM users  
b) **SELECT \* FROM users**  
c) SELECT users.\*  
d) Todas las anteriores
30. ¿Cuál es el comando para actualizar datos en una tabla en PostgreSQL?  
a) **UPDATE**  
b) SET  
c) WHERE  
d) Todas las anteriores
31. ¿Qué significa PostgreSQL?  
a) Post Global Structural Query Language  
b) Postgres Graphical SQL  
c) **Postgres Relational Database Management System**  
d) Post Global Sequence Query Language
32. ¿Qué tipo de licencia tiene PostgreSQL?  
a) **GNU General Public License**  
b) Apache License 2.0  
c) MIT License  
d) BSD 3-Clause License
33. ¿Cuál es el lenguaje de programación utilizado para escribir funciones en PostgreSQL?  
a) PHP  
b) Python  
c) **C**  
d) Todos los anteriores
34. ¿Qué comando se utiliza para crear una nueva base de datos en PostgreSQL?  
a) **CREATE DATABASE**

- b) NEW DATABASE
  - c) ADD DATABASE
  - d) POSTGRES DATABASE
35. ¿Qué comando se utiliza para conectarse a una base de datos en PostgreSQL?
- a) CONNECT TO
  - b) OPEN DATABASE
  - c) USE DATABASE
  - d) \c
36. ¿Qué es un índice en PostgreSQL?
- a) Una tabla con valores únicos
  - b) Una estructura de datos que acelera las consultas
  - c) Un archivo que almacena los datos de la base de datos
  - d) Una función que realiza cálculos complejos
37. ¿Qué es un trigger en PostgreSQL?
- a) Una función que se ejecuta automáticamente cuando se inserta, actualiza o elimina datos de una tabla
  - b) Una tabla que almacena datos históricos
  - c) Una consulta que combina datos de varias tablas
  - d) Un archivo que contiene instrucciones SQL
38. ¿Qué es una vista en PostgreSQL?
- a) Una tabla virtual que se genera a partir de una consulta
  - b) Una tabla que almacena datos históricos
  - c) Una función que realiza cálculos complejos
  - d) Un archivo que contiene instrucciones SQL
39. ¿Qué es una transacción en PostgreSQL?
- a) Un grupo de operaciones que se realizan de manera atómica
  - b) Una tabla que almacena datos históricos
  - c) Una función que se ejecuta automáticamente cuando se inserta, actualiza o elimina datos de una tabla
  - d) Un archivo que contiene instrucciones SQL
40. ¿Qué es un deadlock en PostgreSQL?
- a) Una situación en la que dos o más transacciones quedan bloqueadas indefinidamente
  - b) Una tabla que almacena datos históricos
  - c) Una función que se ejecuta automáticamente cuando se inserta, actualiza o elimina datos de una tabla
  - d) Un archivo que contiene instrucciones SQL
41. ¿Cuál es la sentencia SQL utilizada para recuperar datos de una tabla?
- a) UPDATE
  - b) DELETE
  - c) SELECT
  - d) INSERT
42. ¿Cuál es la sentencia SQL utilizada para actualizar datos en una tabla?
- a) UPDATE
  - b) DELETE
  - c) SELECT
  - d) INSERT
43. ¿Cuál es la sentencia SQL utilizada para eliminar datos de una tabla?
- a) UPDATE
  - b) DELETE
  - c) SELECT
  - d) INSERT
44. ¿Cuál es la sentencia SQL utilizada para insertar datos en una tabla?
- a) UPDATE
  - b) DELETE
  - c) SELECT
  - d) INSERT
45. ¿Cuál es el comando SQL utilizado para crear una nueva tabla?
- a) CREATE
  - b) ALTER
  - c) DROP
  - d) UPDATE
46. ¿Cuál es el comando SQL utilizado para agregar una columna a una tabla existente?
- a) CREATE
  - b) ALTER
  - c) DROP
  - d) UPDATE
47. ¿Cuál es el comando SQL utilizado para eliminar una tabla?
- a) CREATE
  - b) ALTER
  - c) DROP
  - d) UPDATE
48. ¿Cuál es el comando SQL utilizado para seleccionar todos los registros de una tabla?
- a) SELECT \*
  - b) SELECT ALL
  - c) SELECT RECORDS
  - d) SELECT ROWS
49. ¿Cuál es el comando SQL utilizado para seleccionar registros específicos de una tabla?
- a) SELECT DISTINCT
  - b) SELECT WHERE
  - c) SELECT GROUP BY
  - d) SELECT ORDER BY
50. ¿Cuál es el comando SQL utilizado para ordenar los registros de una tabla?
- a) ORDER
  - b) SORT
  - c) GROUP
  - d) ORDER BY

**Serie II.**

Responda las siguientes preguntas de forma manuscrita:

1. ¿Qué es SQL?

El lenguaje de consulta estructurada, es un lenguaje de programación utilizado para almacenar y procesar información en una base de datos.

2. ¿Qué es una clave primaria en una tabla?

Es un campo o conjunto de campos que identifica de forma única cada fila en una tabla.

3. ¿Qué operación SQL se utiliza para insertar datos en una tabla?

Posteriormente a insertar datos a una tabla, debimos haber creado la base de datos y la propia tabla en donde almacenaremos información. Hecho esto debemos recordar el nombre con el que hemos creado la tabla, luego escribimos la siguiente línea de código: **INSER INTO nombredelatabla(columna1, columna2.....) VALUES(valor1, valor2.....).**

4. ¿Qué operación SQL se utiliza para eliminar datos de una tabla?

Tras crear y almacenar cierta cantidad de información en nuestra tabla, usamos la sentencia DELETE para eliminar información almacenada en un registro de la tabla, con lo cual debemos ser más específicos del registro que quiere eliminarse.

5. ¿Qué operación SQL se utiliza para actualizar datos en una tabla?

Para actualizar datos almacenados en nuestra tabla se utiliza la sentencia UPDATE, en este caso también debemos ser específicos del registro que se desea actualizar.

6. ¿Qué es una cláusula WHERE en una consulta SQL?

Esta cláusula permite especificar criterios de búsqueda que deben cumplir las filas para ser incluidas en el resultado de una consulta.

7. ¿Qué es una cláusula JOIN en una consulta SQL?

Se utiliza para combinar filas de dos o más tablas, permitiendo unir datos de diferentes tablas en una única consulta, lo que facilita la recuperación de información relacionada almacenada en tablas separadas.

8. ¿Qué es una vista en SQL?

Es un conjunto de datos derivados de una o más tablas, permiten simplificar consultas complejas al proporcionar una capa de abstracción sobre los datos subyacentes, lo que simplifica el acceso y la manipulación de los datos en la base de datos.

9. ¿Qué es una transacción en SQL?

Es una secuencia de operaciones que se ejecutan como una unidad atómica e indivisible.

10. ¿Qué es una función en SQL?

Es un objeto que encapsula una lógica específica y puede aceptar parámetros de entrada, realizar operaciones y devolver un resultado.

11. ¿Cuál es el comando para crear una matriz de ceros en Octave?

El comando utilizado para crear una matriz sería: zeros(), en donde el número de filas y columnas se especifica dentro del paréntesis.

12. ¿Cómo se realiza una multiplicación de matrices en Octave?

Primera debemos identificar a cada una de nuestras matrices “A=[], B=[]”, posteriormente podemos usar el operador “\*” o el comando “mul(A, B)”, y obtendremos como resultado nuestra matriz resultante.

13. ¿Cómo se calcula el valor absoluto de un número en Octave?

Para calcular el valor absoluto de un número se utiliza el comando “abs()”.

14. ¿Cómo se calcula la media aritmética de un vector en Octave?

El resultado se obtiene sumando todos los elementos del vector y dividiendo el resultado entre el número total de elementos del vector. Puede usarse el comando “mean(2,6,5,7,9,8)” para obtener la media aritmética de ese conjunto de valores.

15. ¿Cómo se obtiene la diagonal de una matriz en Octave?

Debemos establecer nuestra matriz “A=[1,2,3;4,5,6]” luego utilizamos el comando “diag()”, que nos devolverá los elementos principales de la matriz que establecimos previamente.

16. ¿Cómo se calcula la raíz cuadrada de un número en Octave?

Para calcular la raíz cuadrada de un número usamos el comando “sqrt()”, el cual nos devolverá la raíz cuadrada del número ingresado.

17. ¿Cómo se realiza una operación de exponenciación en Octave?

Para elevar un número a la “n” potencia, únicamente usamos “^”.

18. ¿Cómo se calcula el máximo valor de un vector en Octave?

Previamente establecemos el conjunto de números que conformarán nuestro vector, luego utilizamos el comando “max()” para que nos muestre el valor máximo de ese conjunto de datos.

19. ¿Cómo se realiza una suma acumulada en Octave?

Como primer paso debemos establecer un conjunto de datos “vector=[1,2,3,4,5,6]”, una vez establecido el vector utilizamos el comando “cumsum()”, entregando la suma acumulada de ese conjunto de números.

20. ¿Cómo se genera un número aleatorio en Octave?

Para generar números aleatorios se utiliza el comando “rand()”.

**Serie III. (Octave o Python)**

Realice los siguientes programas utilizando try-catch, que tenga un menú con las opciones de ver historial, borrar historial y hacer búsquedas en el historial.

1. Programa de registro de estudiantes: Este programa utiliza Octave y una base de datos para almacenar información de estudiantes, como su nombre, edad, género y dirección. Ofrece opciones para agregar nuevos estudiantes, editar la información de estudiantes existentes y eliminar estudiantes de la base de datos.
2. Programa de seguimiento de presupuesto personal: Este programa utiliza Octave y una base de datos para ayudar a los usuarios a realizar un seguimiento de sus gastos y presupuestos personales. Ofrece opciones para ingresar nuevos gastos, ver un resumen de los gastos acumulados y ajustar los presupuestos según sea necesario.
3. Programa de gestión de inventario: Este programa utiliza Octave y una base de datos para gestionar un inventario de productos. Ofrece opciones para agregar nuevos productos, actualizar la información de los productos existentes y eliminar productos que ya no se necesiten.
4. Programa de seguimiento de pedidos: Este programa utiliza Octave y una base de datos para realizar un seguimiento de los pedidos de los clientes. Ofrece opciones para agregar nuevos pedidos, actualizar la información de los pedidos existentes y eliminar pedidos que ya se hayan completado.
5. Programa de monitoreo de ventas: Este programa utiliza Octave y una base de datos para monitorear las ventas de una empresa. Ofrece opciones para agregar nuevos datos de ventas, generar informes sobre las ventas y analizar los datos para encontrar patrones y tendencias de ventas.

Plantear la mejor solución con diagrama de flujo para los problemas siguientes:

1. Programa de análisis de datos de sensores: Este programa utiliza Octave y una base de datos para analizar los datos de un conjunto de sensores. Incluye un menú que permite al usuario seleccionar el sensor que desea analizar y luego ver los resultados en gráficos y tablas.
2. Sistema de recomendación de películas: Este programa utiliza Octave y una base de datos de películas para crear un sistema de recomendación personalizado para el usuario. El usuario puede ingresar información sobre sus gustos en películas y el programa utiliza algoritmos de aprendizaje automático para recomendar películas similares.
3. Programa de análisis financiero: Este programa utiliza Octave y una base de datos financiera para analizar los datos de una empresa. Incluye un menú que permite al usuario seleccionar la empresa que desea analizar y luego ver los resultados en gráficos y tablas.
4. Sistema de gestión de inventario: Este programa utiliza Octave y una base de datos para crear un sistema de gestión de inventario para una tienda. Incluye un menú que permite al usuario agregar nuevos productos, actualizar la cantidad de inventario y generar informes de ventas.
5. Sistema de planificación de producción: Este programa utiliza Octave y una base de datos para crear un sistema de planificación de producción para una fábrica. Incluye un menú que permite al usuario ingresar información sobre los productos que desea producir y luego utiliza algoritmos de optimización para crear un plan de producción eficiente.

6

**Problema Buscador de Canciones**

Realice un buscador de canciones, el cual despliegue un menú con las opciones:

1. Desplegar el listado de canciones
2. Buscar por artista
3. Buscar por canción
4. Salir

Con la opción 1 deberá de poder ver todas las canciones almacenadas en su base de datos con su respectivo artista y letra. Con la opción 2 y 3 deberá de mostrar las canciones que correspondan según el tipo de filtrado, ya sea por artista o por el nombre de la canción. Deberá utilizar una tabla en PostgreSQL con las columnas “artista”, “canción” y “letra” con al menos 5 canciones de diferente nombre y artista.

7

**Problema Concurso**

Realice un programa en el que se le harán una serie de preguntas al usuario, el cual despliegue un menú con las opciones:

1. Jugar
2. Instrucciones
3. Ver preguntas
4. Salir

Con la opción 1 deberá de comenzar el concurso, para ello se presentarán una a una las preguntas previamente guardadas en su base de datos, el usuario tendrá inicialmente 3 vidas y deberá ingresar su respuesta para cada pregunta, en caso de acertar se sumará un punto al usuario, en caso de fallar se quitará una vida al usuario. Entre cada pregunta se deberá presentar el punteo actual. Tras haber realizado 5 preguntas se presentará la puntuación final y se regresará al menú principal. Deberá utilizar una tabla en PostgreSQL con las columnas “pregunta” y “respuesta” con al menos 5 preguntas. Cada vez que el usuario elija “jugar” deberá presentarse una vez cada pregunta.



Serie IV.

Fase practica:

Realizar los programas usando octave, grabar por 10 segundos contar de 1 al 10:

```
1.
% Comprueba si estamos ejecutando en MATLAB o en Octave
if (exist('OCTAVE_VERSION', 'builtin') ~= 0)
    % Estamos en Octave
    pkg load signal;
end
% Menú principal
opcion = 0;
while opcion ~= 5
    %opcion = input('Seleccione una opción:\n 1. Grabar audio\n 2. Reproducir audio\n 3. Graficar audio\n 4. Salir\n');
    % Menú de opciones
    disp('Seleccione una opción:')
    disp('1. Grabar')
    disp('2. Reproducir')
    disp('3. Graficar')
    disp('4. Graficar densidad')
    disp('5. Salir')
    opcion = input('Ingrese su elección: ');
    switch opcion
        case 1
            % Grabación de audio
            try
                duracion = input('Ingrese la duración de la grabación en segundos: ');
                disp('Comenzando la grabación...');
                recObj = audiorecorder;
                recordblocking(recObj, duracion);
                disp('Grabación finalizada. ');
                data = getaudiodata(recObj);
                audiowrite('audio.wav', data, recObj.SampleRate);
                disp('Archivo de audio grabado correctamente. ');
            catch
                disp('Error al grabar el audio. ');
            end
        case 2
            % Reproducción de audio
            try
                [data, fs] = audioread('audio.wav');
                sound(data, fs);
            catch
                disp('Error al reproducir el audio. ');
            end
        case 3
            % Gráfico de audio
            try
                [data, fs] = audioread('audio.wav');
                tiempo = linspace(0, length(data)/fs, length(data));
                plot(tiempo, data);
                xlabel('Tiempo (s)');
                ylabel('Amplitud');
                title('Audio');
            catch
                disp('Error al graficar el audio. ');
            end
        case 4
            % Graficando espectro de frecuencia
            try
                disp('Graficando espectro de frecuencia...');
                [audio, Fs] = audioread('audio.wav'); % Lee la señal desde el archivo .wav
                N = length(audio); % Número de muestras de la señal
                f = linspace(0, Fs/2, N/2+1); % Vector de frecuencias
                ventana = hann(N); % Ventana de Hann para reducir el efecto de las discontinuidades al calcular la FFT
                Sxx = pwelch(audio, ventana, 0, N, Fs); % Densidad espectral de potencia
                plot(f, 10*log10(Sxx(1:N/2+1))); % Grafica el espectro de frecuencia en dB
                xlabel('Frecuencia (Hz)');
                ylabel('Densidad espectral de potencia (dB/Hz)');
                title('Espectro de frecuencia de la señal grabada');
            catch
                disp('Error al graficar el audio. ');
            end
        case 5
            % Salir
            disp('Saliendo del programa...');
        otherwise
            disp('Opción no válida. ');
    end
end
end
```

Notas:

- Realizar comentarios y observaciones de la salida del programa.
- Presentar reporte de resultados en formato IEEE en único documento pdf con el código y pantallazo que demuestre su funcionamiento.
- Puntos extras realizar el mismo programa en Python.

## SERIE II.

### I. PROGRAMA 1: REGISTRO DE ESTUDIANTES.

```
1 pkg load database
2
3 % Establecer la conexión a la base de datos
4 conn = pg_connect(sprintf('dbname = "REPASO1P1", host = "localhost", port = "5432", user = "postgres", password = "centenario"'));
5
6 while true
7     % Mostrar menú
8     disp("1. Agregar estudiante");
9     disp("2. Editar información de estudiante");
10    disp("3. Eliminar estudiante");
11    disp("4. Ver lista de estudiantes");
12    disp("5. Salir");
13
14    % Solicitar opción al usuario
15    opcion = input("Seleccione una opción: ");
16
17    switch opcion
18        case 1
19            % Agregar estudiante
20            Identificacion = input("Ingrese la identificación del estudiante: ");
21            Nombre = input("Ingrese el nombre del estudiante: ", 's');
22            Edad = input("Ingrese la edad del estudiante: ");
23            Genero = input("Ingrese el género del estudiante: ", 's');
24            Direccion = input("Ingrese la dirección del estudiante: ", 's');
25
26            % Crear y ejecutar la consulta de inserción
27            insert_query = sprintf("INSERT INTO Estudiantes (Identificacion, Nombre, Edad, Genero, Direccion) VALUES ('%s', '%s', %d, '%s', '%s');", num2str(
28                pg_exec_params(conn, insert_query);
29
30            disp("Estudiante agregado exitosamente.");
```

Figura 1: Código en Octave.

```
32 case 2
33     % Editar información de estudiante
34     Identificacion = input("Ingrese la identificación del estudiante a editar: ");
35
36     % Verificar si el estudiante existe
37     select_query = sprintf("SELECT * FROM Estudiantes WHERE Identificacion = '%s'", Identificacion);
38     result = pg_exec_params(conn, select_query);
39
40     if isempty(result.data)
41         disp("Información actual del estudiante:");
42         disp(result.data);
43
44         % Solicitar nueva información
45         nombre = input("Ingrese el nuevo nombre del estudiante: ", 's');
46         edad = input("Ingrese la nueva edad del estudiante: ");
47         genero = input("Ingrese el nuevo género del estudiante: ", 's');
48         direccion = input("Ingrese la nueva dirección del estudiante: ", 's');
49
50         % Crear y ejecutar la consulta de actualización
51         update_query = sprintf("UPDATE Estudiantes SET Nombre = '%s', Edad = %d, Genero = '%s', Direccion = '%s' WHERE Identificacion = '%s';",
52             pg_exec_params(conn, update_query);
53
54         disp("Información del estudiante actualizada exitosamente.");
55     else
56         disp("El estudiante no existe en la base de datos.");
57     end
58
59 case 3
60     % Eliminar estudiante
61     Identificacion = input("Ingrese la identificación del estudiante a eliminar: ");
```

Figura 2: Código en Octave.

```
61
62     Identificacion = input("Ingrese la identificación del estudiante a eliminar: ");
63
64     % Verificar si el estudiante existe
65     select_query = sprintf("SELECT * FROM Estudiantes WHERE Identificacion = '%s'", Identificacion);
66     result = pg_exec_params(conn, select_query);
67
68     if isempty(result.data)
69         % Crear y ejecutar la consulta de eliminación
70         delete_query = sprintf("DELETE FROM Estudiantes WHERE Identificacion = '%s'", Identificacion);
71         pg_exec_params(conn, delete_query);
72
73         disp("Estudiante eliminado exitosamente.");
74     else
75         disp("El estudiante no existe en la base de datos.");
76     end
77
78 case 4
79     % Ver lista de estudiantes
80     select_all_query = "SELECT * FROM Estudiantes;";
81     result = pg_exec_params(conn, select_all_query);
82
83     disp("Lista de estudiantes:");
84     disp(result.data);
85
86 case 5
87     % Salir del programa
88     pg_close(conn);
89     disp("¡Hasta Luego!");
90     return;
91
92 otherwise
```

Figura 3: Código en Octave.

```
65 result = pg_exec_params(conn, select_query);
66
67 if isempty(result.data)
68     % Crear y ejecutar la consulta de eliminación
69     delete_query = sprintf("DELETE FROM Estudiantes WHERE Identificacion = '%s'", Identificacion);
70     pg_exec_params(conn, delete_query);
71
72     disp("Estudiante eliminado exitosamente.");
73 else
74     disp("El estudiante no existe en la base de datos.");
75 end
76
77 case 4
78     % Ver lista de estudiantes
79     select_all_query = "SELECT * FROM Estudiantes;";
80     result = pg_exec_params(conn, select_all_query);
81
82     disp("Lista de estudiantes:");
83     disp(result.data);
84
85 case 5
86     % Salir del programa
87     pg_close(conn);
88     disp("¡Hasta Luego!");
89     return;
90
91 otherwise
92     disp("Opción no válida. Por favor, seleccione una opción válida.");
93 end
94
95
```

Figura 4: Código en Octave.

```
Ingrese la identificación del estudiante: 200701415
Ingrese el nombre del estudiante: Carlos Guzmán
Ingrese la edad del estudiante: 33
Ingrese el género del estudiante: Masculino
Ingrese la dirección del estudiante: Santa Cruz del Quiche
Estudiante agregado exitosamente.
1. Agregar estudiante
2. Editar información de estudiante
3. Eliminar estudiante
4. Ver lista de estudiantes
5. Salir
Seleccione una opción: 4
Lista de estudiantes:
{
    (1,1) = 201906465
    (2,1) = 202207845
    (3,1) = 201509852
    (4,1) = 200701415
    (1,2) = Edgar Franco
    (2,2) = Juan Franco
    (3,2) = Nahum Franco
    (4,2) = Carlos Guzmán
    (1,3) = 23
    (2,3) = 20
    (3,3) = 28
    (4,3) = 33
    (1,4) = Masculino
    (2,4) = Masculino
    (3,4) = Masculino
    (4,4) = Masculino
    (1,5) = Chichicastenango
    (2,5) = Chichicastenango
    (3,5) = Chichicastenango
    (4,5) = Santa Cruz del Quiche
}
1. Agregar estudiante
2. Editar información de estudiante
3. Eliminar estudiante
4. Ver lista de estudiantes
5. Salir
```

Figura 5: Resultados en Octave.

```
1 --Tabla Registro de estudiantes.
2
3 create table Estudiantes(
4     Identificacion varchar(200),
5     Nombre varchar(200),
6     Edad varchar(200),
7     Genero varchar(200),
8     Direccion varchar(200)
9 );
10
11 Select * from Estudiantes;
12
```

Figura 6: Creación de base de datos en Postgres.

	identificacion character varying (200)	nombre character varying (200)	edad character varying (10)	genero character varying (50)	direccion character varying (200)
1	201906465	Edgar Franco	23	Masculino	Chichicastenango
2	202207845	Juan Franco	20	Masculino	Chichicastenango

Figura 7: Resultados en Postgres.

## II. PROGRAMA 2: SEGUIMIENTO DE PRESUPUESTO PERSONAL.

```

1 pkg load database
2
3 function conn = conectar_bd(Descripcion, Monto, Fecha)
4     dbname = "REPASOIP";
5     host = "localhost";
6     port = 5432;
7     user = "postgres";
8     password = "centenario";
9
10    conn = pg_connect(testdbopts('dbname', 'REPASOIP', 'host', 'localhost', 'port', '5432', 'user', 'postgres', 'password', 'centenario'));
11    query = sprintf("INSERT INTO PresP (Descripcion, Monto, Fecha) VALUES ('%s','%s','%s')", Descripcion, Monto, Fecha);
12    pg_exec(conn, query);
13 end
14
15 function Presupuesto()
16     historial = struct('Descripcion', {}, 'Monto', {}, 'Fecha', {});
17
18     while true
19         fprintf('Bienvenido al Programa de Seguimiento de Presupuesto Personal\n');
20         fprintf('1. Ingresar nuevo gasto\n');
21         fprintf('2. Ver resumen de gastos acumulados\n');
22         fprintf('3. Ajustar presupuestos\n');
23         fprintf('4. Salir\n');
24
25         opcion = input('Seleccione una opción: ');
26
27         try
28             switch opcion
29                 case 1
30                     Descripcion = input('Ingrese la descripción del gasto: ');
31                     Monto = input('Ingrese el monto del gasto: ');
32

```

Figura 8: Código en Octave.

```

-- Presupuesto
-- Bienvenido al Programa de Seguimiento de Presupuesto Personal
1. Ingresar nuevo gasto
2. Ver resumen de gastos acumulados
3. Ajustar presupuestos
4. Salir
Seleccione una opción: 1
Ingrese la descripción del gasto: Compra
Ingrese el monto del gasto: 100
Ingrese la fecha del gasto (DD-MM-YYYY): 21-08-2023
Gasto ingresado con éxito
Bienvenido al Programa de Seguimiento de Presupuesto Personal
1. Ingresar nuevo gasto
2. Ver resumen de gastos acumulados
3. Ajustar presupuestos
4. Salir
Seleccione una opción: 1
Ingrese la descripción del gasto: Compra
Ingrese el monto del gasto: 100
Ingrese la fecha del gasto (DD-MM-YYYY): 21-08-2023
Gasto ingresado con éxito
Bienvenido al Programa de Seguimiento de Presupuesto Personal
1. Ingresar nuevo gasto
2. Ver resumen de gastos acumulados
3. Ajustar presupuestos
4. Salir
Seleccione una opción: 2
Resumen de gastos acumulados:
Compra - 200.00 - 21-08-2023
Compra - 100.00 - 21-08-2023
Bienvenido al Programa de Seguimiento de Presupuesto Personal
1. Ingresar nuevo gasto
2. Ver resumen de gastos acumulados
3. Ajustar presupuestos
4. Salir
Seleccione una opción: 4
¡Hasta luego!
>>

```

Figura 11: Resultados en Octave.

```

1 --Creación de Tabla
2 --create table Presupersonal(
3     --descripcion varchar(80),
4     --monto decimal not null,
5     --fecha date not null
6 --);
7
8 --Ver datos en la tabla
9 Select*from Presupersonal

```

```

31     Descripcion = input('Ingrese la descripción del gasto: ');
32     Monto = input('Ingrese el monto del gasto: ');
33     Fecha = input('Ingrese la fecha del gasto (DD-MM-YYYY): ');
34
35     nuevoGasto.Descripcion = Descripcion;
36     nuevoGasto.Monto = Monto;
37     nuevoGasto.Fecha = Fecha;
38
39     historial(end + 1) = nuevoGasto;
40
41     fprintf('Gasto ingresado con éxito.\n');
42     case 2
43         if isempty(historial)
44             fprintf('No hay gastos en el historial.\n');
45         else
46             fprintf('Resumen de gastos acumulados:\n');
47             for i = 1:length(historial)
48                 fprintf('%s - %.2f - %s\n', historial(i).Descripcion, historial(i).Monto, historial(i).Fecha);
49             end
50         end
51     case 3
52         fprintf('Función de ajustar presupuestos no implementada en este ejemplo.\n');
53     case 4
54         fprintf('¡Hasta luego!\n');
55         return;
56     otherwise
57         fprintf('Opción no válida. Por favor, seleccione una opción válida.\n');
58     end
59 catch
60     fprintf('Ha ocurrido un error inesperado. Por favor, inténtalo nuevamente.\n');
61 end

```

Figura 9: Código en Octave.

Figura 12: Creación de base de datos en Postgres.

	descripcion	monto	fecha
	character varying (80)	numeric	date
1	Compra	100.000000	2023-08-31
2	Compra	250.000000	2023-08-31

Figura 13: Resultados en Postgres.

## III. PROGRAMA 3: GESTIÓN DE INVENTARIO.

```

36     nuevoGasto.Monto = Monto;
37     nuevoGasto.Fecha = Fecha;
38
39     historial(end + 1) = nuevoGasto;
40
41     fprintf('Gasto ingresado con éxito.\n');
42     case 2
43         if isempty(historial)
44             fprintf('No hay gastos en el historial.\n');
45         else
46             fprintf('Resumen de gastos acumulados:\n');
47             for i = 1:length(historial)
48                 fprintf('%s - %.2f - %s\n', historial(i).Descripcion, historial(i).Monto, historial(i).Fecha);
49             end
50         end
51     case 3
52         fprintf('Función de ajustar presupuestos no implementada en este ejemplo.\n');
53     case 4
54         fprintf('¡Hasta luego!\n');
55         return;
56     otherwise
57         fprintf('Opción no válida. Por favor, seleccione una opción válida.\n');
58     end
59 catch
60     fprintf('Ha ocurrido un error inesperado. Por favor, inténtalo nuevamente.\n');
61 end
62
63 conn = conectar_bd(Descripcion, Monto, Fecha);
64 end

```

Figura 10: Código en Octave.

```

1 pkg load database
2
3 function conn = conectar_bd()
4     dbname = "REPASOIP";
5     host = "localhost";
6     port = 5432;
7     user = "postgres";
8     password = "centenario";
9
10    conn = pg_connect(testdbopts('dbname', 'REPASOIP', 'host', 'localhost', 'port', '5432', 'user', 'postgres', 'password', 'centenario'));
11    query = sprintf("UPDATE Inventario SET cantidad = %d, precio = %f WHERE Identificacion = '%s'", cantidad, precio, identificacion);
12    pg_exec(conn, query);
13 end
14
15 function agregar_producto(conn, nombre, identificacion, cantidad, precio)
16     query = sprintf("INSERT INTO Inventario (nombre, identificacion, cantidad, precio) VALUES ('%s','%s',%d,%f)", nombre, identificacion, cantidad, precio);
17    pg_exec(conn, query);
18 end
19
20 function eliminar_producto(conn, identificacion)
21     query = sprintf("DELETE FROM Inventario WHERE Identificacion = '%s'", identificacion);
22    pg_exec(conn, query);
23 end
24
25 function gestion_inventario()
26     conn = conectar_bd();
27
28     while true
29         fprintf('Bienvenido al Programa de Gestión de Inventario\n');
30         fprintf('1. Agregar nuevo producto\n');
31         fprintf('2. Actualizar información de producto\n');
32         fprintf('3. Eliminar producto\n');
33         fprintf('4. Salir\n');

```

Figura 14: Código en Octave.



```

32 opcion = input("Seleccione una opción: ");
33
34 try
35     switch opcion
36     case 1
37         nombre = input("Ingrese el nombre del producto: ", 's');
38         identificacion = input("Ingrese la identificación del producto: ", 's');
39         cantidad = input("Ingrese la cantidad del producto: ");
40         precio = input("Ingrese el precio del producto: ");
41         agregar_producto(conn, nombre, identificacion, cantidad, precio);
42         fprintf('Producto agregado exitosamente.\n');
43     case 2
44         identificacion = input("Ingrese la identificación del producto a actualizar: ", 's');
45         cantidad = input("Ingrese la nueva cantidad del producto: ");
46         precio = input("Ingrese el nuevo precio del producto: ");
47         actualizar_producto(conn, identificacion, cantidad, precio);
48         fprintf('Información del producto actualizada exitosamente.\n');
49     case 3
50         identificacion = input("Ingrese la identificación del producto a eliminar: ", 's');
51         eliminar_producto(conn, identificacion);
52         fprintf('Producto eliminado exitosamente.\n');
53     case 4
54         pq_close(conn);
55         fprintf('¡Hasta luego!\n');
56         return;
57     end
58 catch
59     fprintf('Ha ocurrido un error inesperado. Por favor, inténtelo nuevamente.\n');
60 end
61
62 gestion_inventario();

```

Figura 15: Código en Octave.

```

43 fprintf('Producto agregado exitosamente.\n');
44
45 case 2
46     identificacion = input("Ingrese la identificación del producto a actualizar: ", 's');
47     cantidad = input("Ingrese la nueva cantidad del producto: ");
48     precio = input("Ingrese el nuevo precio del producto: ");
49     actualizar_producto(conn, identificacion, cantidad, precio);
50     fprintf('Información del producto actualizada exitosamente.\n');
51 case 3
52     identificacion = input("Ingrese la identificación del producto a eliminar: ", 's');
53     eliminar_producto(conn, identificacion);
54     fprintf('Producto eliminado exitosamente.\n');
55 case 4
56     pq_close(conn);
57     fprintf('¡Hasta luego!\n');
58     return;
59 otherwise
60     fprintf('Opción no válida. Por favor, seleccione una opción válida.\n');
61 end
62 catch
63     fprintf('Ha ocurrido un error inesperado. Por favor, inténtelo nuevamente.\n');
64 end
65
66 gestion_inventario();

```

Figura 16: Código en Octave.

```

Ingrese el precio del producto: 900.00
Producto agregado exitosamente.
Bienvenido al Programa de Gestión de Inventario
1. Agregar nuevo producto
2. Actualizar información de producto
3. Eliminar producto
4. Salir
Seleccione una opción: 1
Ingrese el nombre del producto: Audifonos SkullCandy
Ingrese la identificación del producto: 989
Ingrese la cantidad del producto: 300
Ingrese el precio del producto: 359.00
Producto agregado exitosamente.
Bienvenido al Programa de Gestión de Inventario
1. Agregar nuevo producto
2. Actualizar información de producto
3. Eliminar producto
4. Salir
Seleccione una opción: 2
Ingrese la identificación del producto a actualizar: 989
Ingrese la nueva cantidad del producto: 400
Ingrese el nuevo precio del producto: 300.00
Información del producto actualizada exitosamente.
Bienvenido al Programa de Gestión de Inventario
1. Agregar nuevo producto
2. Actualizar información de producto
3. Eliminar producto
4. Salir
Seleccione una opción: 3
Ingrese la identificación del producto a eliminar: 899.99
Producto eliminado exitosamente.
Bienvenido al Programa de Gestión de Inventario
1. Agregar nuevo producto
2. Actualizar información de producto
3. Eliminar producto
4. Salir
Seleccione una opción: 4
¡Hasta luego!
>>

```

Figura 17: Resultados en Octave.

```

1 --Tabla de gestión de inventarios
2
3 --Create table Inventario(
4     --id serial Primary Key,
5     --nombre varchar(255),
6     --identificacion varchar(80),
7     --cantidad integer not null,
8     --precio numeric(10,2) not null
9 --)
10
11 Select * from Inventario;
12

```

Figura 18: Creación de base de datos en Postgres.

	id [PK] integer	nombre character varying (255)	identificacion character varying (80)	cantidad integer	precio numeric (10,2)
1	1	Television LG	626	35	5000.00
2	2	Tenis Nike	300	900	900.00
3	3	Audifonos SkullCandy	989	400	300.00
Total rows: 3 of 3		Query complete 00:00:00.327			

Figura 19: Resultados en Postgres.

#### IV. PROGRAMA 4: SEGUIMIENTO DE PEDIDOS.

```

1 pkg load database
2
3 conn = pg_connect(setdbopts('dbname', 'REPOSOLP', 'host', 'localhost', 'port', '5432', 'user', 'postgres', 'password', 'centenario'));
4
5 while true
6     % Mostrar menú
7     disp('Programa de Seguimiento de Pedidos:');
8     disp('1. Agregar nuevo pedido');
9     disp('2. Actualizar estado de pedido');
10    disp('3. Eliminar pedido');
11    disp('4. Ver historial de pedidos');
12    disp('5. Borrar historial de pedidos');
13    disp('6. Buscar en historial de pedidos');
14    disp('0. Salir');
15
16    % Solicitar opción al usuario
17    opcion = input("Seleccione una opción: ");
18
19    switch opcion
20    case 1
21        % Agregar nuevo pedido
22        cliente = input("Ingrese el nombre del cliente: ", 's');
23        producto = input("Ingrese el nombre del producto: ", 's');
24        estado = input("Ingrese el estado del pedido: ", 's');
25
26        % Crear y ejecutar la consulta de inserción
27        insert_query = sprintf("INSERT INTO SPedidos (cliente, producto, estado) VALUES ('%s', '%s', '%s')", cliente, producto, estado);
28        pq_exec_params(conn, insert_query);
29
30        disp("Pedido registrado exitosamente.");
31    case 2
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64

```

Figura 20: Código en Octave.

```

33 % Actualizar estado de pedido
34 id_pedido = input("Ingrese el ID del pedido a actualizar: ");
35 nuevo_estado = input("Ingrese el nuevo estado del pedido: ", 's');
36
37 % Crear y ejecutar la consulta de actualización
38 update_query = sprintf("UPDATE SPedidos SET estado = '%s' WHERE id = %d;", nuevo_estado, id_pedido);
39 pq_exec_params(conn, update_query);
40
41 disp("Estado del pedido actualizado exitosamente.");
42
43 case 3
44     % Eliminar pedido
45     id_pedido = input("Ingrese el ID del pedido a eliminar: ");
46
47     % Crear y ejecutar la consulta de eliminación
48     delete_query = sprintf("DELETE FROM SPedidos WHERE id = %d;", id_pedido);
49     pq_exec_params(conn, delete_query);
50
51     disp("Pedido eliminado exitosamente.");
52
53 case 4
54     % Ver historial de pedidos
55     select_all_query = "SELECT * FROM SPedidos;";
56     result = pq_exec_params(conn, select_all_query);
57
58     disp("Historial de pedidos:");
59     disp(result.data);
60
61 case 5
62     % Borrar historial de pedidos
63     disp("¡Atención! Esta acción borrará todos los registros de pedidos.");
64     confirmation = input("¿Está seguro? (S/N): ", 's');

```

Figura 21: Código en Octave.

```

53 case 4
54     % Ver historial de pedidos
55     select_all_query = "SELECT * FROM SPedidos;";
56     result = pq_exec_params(conn, select_all_query);
57     disp("Historial de pedidos:");
58     disp(result.data);
59
60 case 5
61     % Borrar historial de pedidos
62     disp("¡Atención! Esta acción borrará todos los registros de pedidos.");
63     confirmation = input("¿Está seguro? (S/N): ", 's');
64
65     if strcmp(confirmation, 'S')
66         delete_query = "DELETE FROM SPedidos;";
67         pq_exec_params(conn, delete_query);
68         disp("Historial de pedidos borrado exitosamente.");
69     end
70
71 case 6
72     % Buscar en historial de pedidos (puedes implementar esta opción según tus necesidades)
73
74 case 0
75     % Salir del programa
76     pq_close(conn);
77     disp("¡Hasta luego!");
78     return;
79
80 otherwise
81     disp("Opción no válida. Por favor, seleccione una opción válida.");
82 end
83 end
84 end

```

Figura 22: Código en Octave.

```

4. Ver historial de pedidos
5. Borrar historial de pedidos
6. Buscar en historial de pedidos
0. Salir
Seleccione una opción: 1
Ingrese el nombre del cliente: Juan Franco
Ingrese el nombre del producto: Computadora portatil HP
Ingrese el estado del pedido: En ruta
Pedido registrado exitosamente.
Programa de Seguimiento de Pedidos:
1. Agregar nuevo pedido
2. Actualizar estado de pedido
3. Eliminar pedido
4. Ver historial de pedidos
5. Borrar historial de pedidos
6. Buscar en historial de pedidos
0. Salir
Seleccione una opción: 2
Ingrese el ID del pedido a actualizar: 1
Ingrese el nuevo estado del pedido: Entregado
Estado del pedido actualizado exitosamente.
Programa de Seguimiento de Pedidos:
1. Agregar nuevo pedido
2. Actualizar estado de pedido
3. Eliminar pedido
4. Ver historial de pedidos
5. Borrar historial de pedidos
6. Buscar en historial de pedidos
0. Salir
Seleccione una opción: 4
Historial de pedidos:
[
    [1,1] = 2
    [2,1] = 1
    [1,2] = Juan Franco
    [2,2] = Edgar Franco
    [1,3] = Computadora portatil HP
    [2,3] = Audifonos inalámbricos SkullCandy
    [1,4] = En ruta
    [2,4] = Entregado
]

```

Figura 23: Resultados en Octave.

```

1 --Tabla de seguimiento de pedidos.
2
3 --CREATE TABLE SPedidos (
4     --id SERIAL PRIMARY KEY,
5     --cliente varchar(100) not null,
6     --producto varchar(100) not null,
7     --estado varchar(100) not null
8 --);
9
10
11
12 Select * from SPedidos;

```

Figura 24: Creación de base de datos en Postgres.

Data Output Messages Notifications				
	id [PK] integer	cliente character varying (100)	producto character varying (100)	estado character varying (100)
1		2 Juan Franco	Computadora portatil HP	En ruta
2		1 Edgar Franco	Audifonos inalámbricos SkullCandy	Entregado

Figura 25: Resultados en Postgres.

## V. PROGRAMA 5: MONITOREO DE VENTAS.

```

1 % Cargar la librería 'octave-pgsql'
2 pkg load database;
3
4 conn = pq_connect(setdbopts('dbname','REPASO1P','host','localhost','port','5432','user','postgres','password','centenario'));
5
6 while true
7     disp("1. Agregar venta:");
8     disp("2. Generar informe de ventas:");
9     disp("3. Modificar inventario:");
10    disp("4. Salir");
11
12    % Solicitar una opción al usuario.
13    opcion = input("Seleccione una opción: ");
14
15    switch opcion
16    case 1 %Agregar venta
17        producto = input("Ingrese el nombre del producto vendido: ", 's');
18        while ~isempty(strfind(producto, ' '))
19            disp("Error: El nombre del producto no puede contener comillas simples ('), por favor, ingrese nuevamente.");
20            producto = input("Ingrese el nombre del producto vendido: ", 's');
21        endwhile
22
23        cantidad = input("Ingrese la cantidad vendida: ");
24        insert_query = sprintf("INSERT INTO Monitoreo (producto, cantidad) VALUES ('%s',%d)",producto,cantidad);
25        pq_exec_params(conn, insert_query);
26        disp("Venta registrada exitosamente.");
27
28    case 2
29        select_all_query = "Select * From Monitoreo;";
30        result = pq_exec_params(conn, select_all_query);
31        disp("Informe de ventas:");
32        disp(result.data);
33    end
34 end

```

Figura 26: Código en Octave.

```

28 case 2
29     select_all_query = "Select * From Monitoreo;";
30     result = pq_exec_params(conn, select_all_query);
31     disp("Informe de ventas:");
32     disp(result.data);
33
34 case 3
35     id_modificar = input("Ingrese el ID del producto a modificar: ");
36
37     while ~isnumeric(id_modificar) || id_modificar <= 0 || mod(id_modificar, 1) ~= 0
38         disp("Error: Ingrese un ID valido (numero entero positivo).");
39         id_modificar = input("Ingrese el ID del producto a modificar: ");
40     endwhile
41
42     cantidad_nueva = input("Ingrese la nueva cantidad en inventario: ");
43
44     update_query = sprintf("UPDATE Monitoreo SET cantidad = %d WHERE id = %d;",cantidad_nueva, id_modificar);
45     pq_exec_params(conn, update_query);
46     disp("Inventario modificado exitosamente.");
47
48 case 4
49     pq_close(conn);
50     disp("Saliendo del programa...");
51     return;
52
53 otherwise
54     disp("Opción no valida. Seleccione otra opcion.");
55 end
56 end
57 end

```

Figura 27: Código en Octave.

```

>> P1IN25
1. Agregar venta
2. Generar informe de ventas
3. Analizar datos de ventas
4. Salir
Seleccione una opción: 1
Ingrese el nombre del producto vendido: calcewines
Ingrese la cantidad vendida: 400
Venta registrada exitosamente.
1. Agregar venta
2. Generar informe de ventas
3. Analizar datos de ventas
4. Salir
Seleccione una opción: 2
Informe de ventas:
[
    [1,1] = 1
    [2,1] = 2
    [3,1] = 3
    [1,2] = 75559183674210
    [2,2] = 75559183174614
    [3,2] = 755591849180282
    [1,3] = camisa
    [2,3] = pantalón
    [3,3] = calcewines
    [1,4] = 200
    [2,4] = 500
    [3,4] = 400
]
1. Agregar venta
2. Generar informe de ventas
3. Analizar datos de ventas
4. Salir
Seleccione una opción: 3
Análisis de datos de ventas:
error: 'rotatedLabels' undefined near line 66, column 5
error: called from
P1IN25 at line 66 column 5
>>

```

Figura 28: Resultados en Octave.

```

1  --Tabla de monitoreo de ventas
2
3  --CREATE TABLE Monitoreo (
4    --producto varchar(1000),
5    --cantidad varchar(1000)
6  --);
7
8  Select * from Monitoreo

```

Figura 29: Creación de base de datos en Postgres.

	id_venta [PK] integer	historial timestamp without time zone	producto character varying (255)	cantidad integer
1	1	2023-12-15 21:43:56.97421	camisa	200
2	2	2023-12-15 21:44:11.734814	pantalón	500
3	3	2023-12-15 21:45:49.188252	calcetines	400

Figura 30: Resultados en Postgres.

## VII. PROGRAMA 6: BUSCADOR DE CANCIONES.

```

1  pkg load database
2
3  % Establecer la conexión a la base de datos
4  conn = pg_connect(isetdbopts('dbname', 'REPASOLP', 'host', 'localhost', 'port', '5432', 'user', 'postgres', 'password', 'centenario'));
5
6  cancion = ''; % Definir la variable cancion fuera del bucle while
7
8  while true
9    % Mostrar menú
10   disp('1. Desplegar el listado de canciones. ');
11   disp('2. Buscar por artista. ');
12   disp('3. Buscar por cancion. ');
13   disp('4. Salir. ');
14
15   % Solicitar opción al usuario
16   opcion = input('Seleccione una opción: ');
17
18   switch opcion
19     case 1
20       % Desplegar el listado de canciones
21       select_all_query = "SELECT * FROM LisCanciones";
22       result = pg_exec_params(conn, select_all_query);
23
24       disp('Listado de canciones: ');
25       disp(result.data);
26
27     case 2
28       % Buscar por artista
29       artista = input('Ingrese el nombre del artista: ');
30       select_query = sprintf('SELECT * FROM LisCanciones WHERE artista = %s', artista);

```

Figura 32: Código en Octave.

## VI. DIAGRAMA DE FLUJO: ANÁLISIS DE DATOS DE SENSORES.

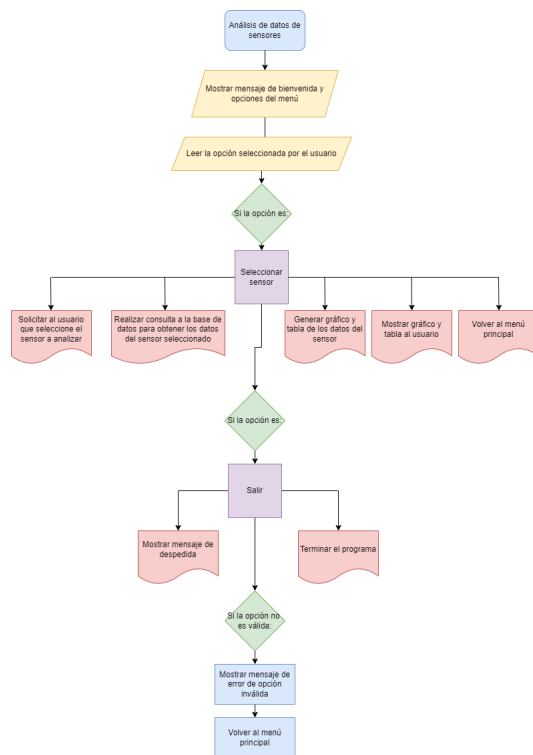


Figura 31: Diagrama de flujo.

```

31   % Ejecutar la consulta SQL
32   result = pg_exec_params(conn, select_query);
33
34   % Verificar si se encontraron resultados
35   if isempty(result.data)
36     disp('No se encontraron canciones para el artista especificado. ');
37   else
38     % Mostrar las canciones del artista
39     disp('Canciones del artista: ');
40     disp(result.data);
41   end
42
43   case 3
44     % Buscar por cancion
45     cancion = input('Ingrese el nombre de la cancion: ');
46     select_query = sprintf('SELECT * FROM LisCanciones WHERE cancion = %s', cancion);
47
48     % Ejecutar la consulta SQL
49     result = pg_exec_params(conn, select_query);
50
51     % Verificar si se encontraron resultados
52     if isempty(result.data)
53       disp('No se encontraron canciones para el artista especificado. ');
54     else
55       % Mostrar las canciones del artista
56       disp('Canciones del artista: ');
57       disp(result.data);
58     end
59
60   end

```

Figura 33: Código en Octave.

```

43   case 3
44     % Buscar por cancion
45     cancion = input('Ingrese el nombre de la cancion: ');
46     select_query = sprintf('SELECT * FROM LisCanciones WHERE cancion = %s', cancion);
47
48     % Ejecutar la consulta SQL
49     result = pg_exec_params(conn, select_query);
50
51     % Verificar si se encontraron resultados
52     if isempty(result.data)
53       disp('No se encontraron canciones para el artista especificado. ');
54     else
55       % Mostrar las canciones del artista
56       disp('Canciones del artista: ');
57       disp(result.data);
58     end
59
60   end
61
62   case 4
63     % Salir del programa
64     pg_close(conn);
65     disp('¡Basta luego!');
66     return;
67
68   otherwise
69     disp('Opción no válida. Por favor, seleccione una opción válida. ');
70
71   end
72
73 end

```

Figura 34: Código en Octave.

```

Ventana de comandos
>> LISTADOCCM
1. Desplegar el listado de canciones.
2. Buscar por artista.
3. Buscar por canción
4. Salir
Seleccione una opción: 1
Listado de canciones:
{
  [1,1] = Babasonicos
  [2,1] = Los Caligaris
  [3,1] = Soda Stereo
  [4,1] = Guns N Roses
  [5,1] = Alux Nahual
  [1,2] = Risa
  [2,2] = Kilometros
  [3,2] = Profugos
  [4,2] = Dont Cry
  [5,2] = Conquista
}
1. Desplegar el listado de canciones.
2. Buscar por artista.
3. Buscar por canción
4. Salir
Seleccione una opción: 2
Ingrese el nombre del artista: Babasonicos
Canciones del artista:
{
  [1,1] = Babasonicos
  [1,2] = Risa
}
1. Desplegar el listado de canciones.
2. Buscar por artista.
3. Buscar por canción
4. Salir
Seleccione una opción: 3
Ingrese el nombre de la canción: Risa
Canciones del artista:
{
  [1,1] = Babasonicos
  [1,2] = Risa
}

```

Figura 35: Resultados en Octave.

## VIII. SERIE IV.

### A. Grabadora de audio.

```

1 %Comprobación de ejecución de Octave
2
3 if (exist('OCTAVE_VERSION', 'builtin') ==0)
4   pkg load signal;
5 end
6
7 %Menú principal
8
9 opcion = 0;
10 while opcion ~=5
11   %opcion = input('Seleccione una opcion:\n 1.Grabar audio\n 2.Reproducir audio\n 3.Graficar audio')
12   %Menu de opciones
13   disp('Seleccione una opcion: ')
14   disp('1. Grabar')
15   disp('2. Reproducir')
16   disp('3. Graficar')
17   disp('4. Graficar densidad')
18   disp('5. Salir')
19   opcion = input('Ingrese su elección: ');
20   switch opcion
21     case 1 %Grabación de audio
22       try
23         duracion = input('Ingrese la duración de la grabación en segundos: ');
24         disp('Comenzando la grabación....');
25         recObj = audiorecorder;
26         recordblocking(recObj, duracion);
27         disp('Grabación finalizada. ');
28         data = getaudiodata(recObj);
29         audiowrite('audio.wav', data, recObj.SampleRate);
30         disp('Archivo de audio grabado correctamente. ');
31       catch

```

Figura 37: Código en Octave.

```

32       disp('Error al grabar el audio. ');
33     end_try_catch
34
35     case 2 %Reproducción de audio
36       try
37         [data, fs] = audioread('audio.wav');
38         sound(data, fs);
39       catch
40         disp('Error al reproducir el audio. ');
41       end_try_catch
42
43     case 3 %Gráfico de audio
44       try
45         [data, fs] = audioread('audio.wav');
46         tiempo = linspace(0, length(data)/fs, length(data));
47         plot(tiempo, data);
48         xlabel('Tiempo (s)');
49         ylabel('Amplitud');
50         title('Audio');
51       catch
52         disp('Error al graficar el audio. ');
53       end_try_catch
54
55     case 4 %Graficando el espectro de frecuencia
56       try
57         disp('Graficando espectro de frecuencia....');
58         [audio, Fs] = audioread('audio.wav'); %Lee la señal desde el archivo .wav
59         N = length(audio); %Numero de muestras de la señal
60         f = linspace(0, Fs/2, N/2+1); %Vector de frecuencias
61         ventana = hann(N); %Ventana de Hann para reducir el efecto de las discontinuidades al calcul
62         Sxx = pwelch(audio, ventana, 0, N, Fs); %Densidad espectral de potencia

```

Figura 38: Código en Octave.

Query	Query History
1	--Ver datos ingresados en la tabla
2	
3	Select*from LisCanciones
4	
5	
6	

Data Output	Messages	Notifications
	artista character varying (1000)	cancion character varying (1000)
1	Babasonicos	Risa
2	Los Caligaris	Kilometros
3	Soda Stereo	Profugos

Figura 36: Resultados en Postgres.

```

49       ylabel('Amplitud');
50       title('Audio');
51     catch
52       disp('Error al graficar el audio. ');
53     end_try_catch
54
55     case 4 %Graficando el espectro de frecuencia
56       try
57         disp('Graficando espectro de frecuencia....');
58         [audio, Fs] = audioread('audio.wav'); %Lee la señal desde el archivo .wav
59         N = length(audio); %Numero de muestras de la señal
60         f = linspace(0, Fs/2, N/2+1); %Vector de frecuencias
61         ventana = hann(N); %Ventana de Hann para reducir el efecto de las discontinuidades al calcul
62         Sxx = pwelch(audio, ventana, 0, N, Fs); %Densidad espectral de potencia
63         plot(f, 10*log10(Sxx(1:N/2+1))); %Grafica del espectro de frecuencia en dB
64         xlabel('Frecuencia (Hz)');
65         ylabel('Densidad espectral de potencia (dB/Hz)');
66         title('Espectro de frecuencia de la señal grabada');
67       catch
68         disp('Error al graficar el audio. ');
69       end_try_catch
70
71     case 5 %Salir
72       disp('Saliendo del programa....');
73       break
74     otherwise
75       disp('Opción no válida....');
76     end
77 end

```

Figura 39: Código en Octave.

```

>> GRABADORA
Seleccione una opción:
1. Grabar
2. Reproducir
3. Graficar
4. Graficar densidad
5. Salir
Ingrese su elección: 1
Ingrese la duración de la grabación en segundos: 10
Comenzando la grabación....
Grabación finalizada.
Archivo de audio grabado correctamente.
Seleccione una opción:
1. Grabar
2. Reproducir
3. Graficar
4. Graficar densidad
5. Salir
Ingrese su elección: 2
Seleccione una opción:
1. Grabar
2. Reproducir
3. Graficar
4. Graficar densidad
5. Salir
Ingrese su elección: 3
Seleccione una opción:
1. Grabar
2. Reproducir
3. Graficar
4. Graficar densidad
5. Salir
Ingrese su elección: 4
Graficando espectro de frecuencia.....
Seleccione una opción:
1. Grabar
2. Reproducir
3. Graficar
4. Graficar densidad
5. Salir

```

Figura 40: Resultados en Octave.

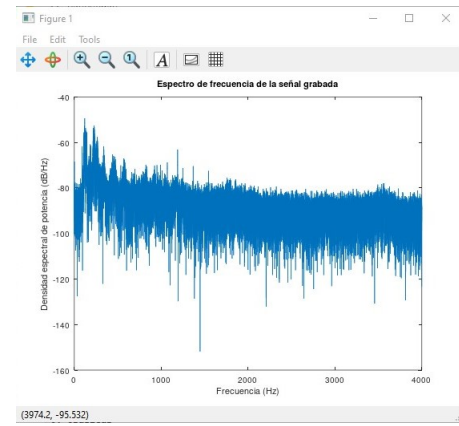


Figura 42: Espectro de frecuencia.

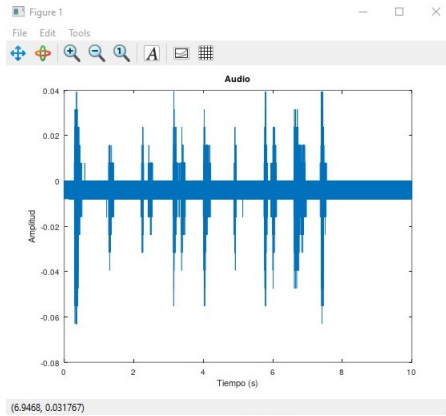


Figura 41: Gráfica de audio.

## IX. LINK DE GITHUB.

<https://github.com/EdgarFranco8/TAREA-PREPARATORIA-1/tree/main/TAREA%20PREPARATORIA%20PRIMER%20PARCIAL>