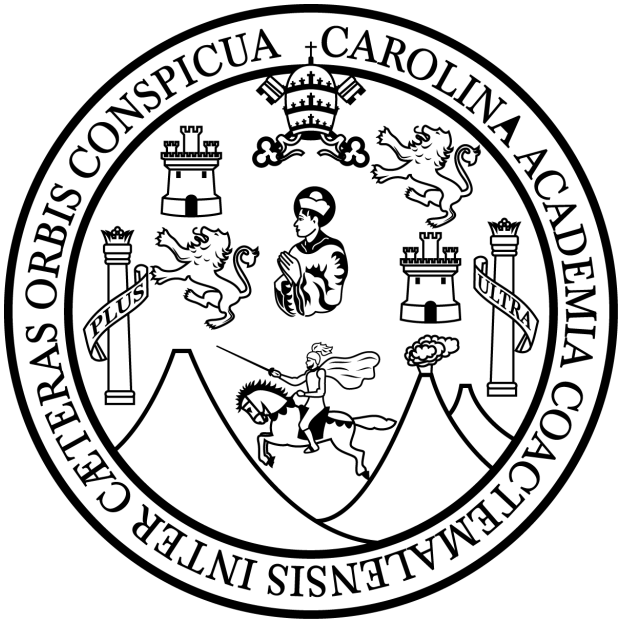
Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Mecánica Eléctrica
Proyectos de Computación Aplicados a la Ingeniería Electrónica.
Profesor: Ing. José Anibal Silva De Los Angeles.
Fecha de la práctica: 16/12/2024



Primer Parcial

Fecha: 16/12/2024

Estudiantes: *Carné:*

Cindy Melissa Gatica Arriola 201709692

Primer Parcial
Proyectos de Computación Aplicados a la Ingeniería Electrónica.[*]

Cindy Melissa Gatica Arriola, 201709692[1,**]

[1] *Facultad de Ingeniería, Universidad de San Carlos, Edificio T1, Ciudad Universitaria, Zona 12, Guatemala.*

Se desea desarrollar varios programas que permitan ingresar datos para su posterior análisis. Se presenta un menú en el cual, el usuario puede elegir entre *ingresar datos, calcular factorial, sistema login, calcular distancia entre dos puntos, calcular IMC, Calculo de sueldo base semanal, historial de datos, borrar datos o salir*. Cada opcion tiene integrada un proceso diferente. Los resultados fueron guardados en una base de datos PostgreSQL y un archivo de texto salida.txt. Finalmente, se le daba la opcion al usuario de borrar el historial o salir.

I. OBJETIVOS

A. Generales

- Desarrollar un programa que permita implementar varias funcionalidades, guardarla en una base de datos PostgreSQL y un archivo de texto, así como eliminarlos si el usuario lo solicita.

B. Específicos

* Implementar una función que permita calcular el factorial de un número.

* Construir una función que calcule el sueldo semanal de un trabajador.

* Codificar una función que permita calcular la distancia entre puntos cartesianos.

* Calcular el Indice de Masa Corporal IMC de un usuario al ingresar su peso y altura.

* Implementar un sistema Login, en el cual, el usuario tiene unicamente tres intentos, los cuales son mostrados en pantalla.

II. CÓDIGO EN OCTAVE

El código implementado en Octave es el siguiente

```
clc;
clear;
pkg load database;

%Variable para ir almacenando los resultados
global resultados conn;
resultados = struct();

try
    % Conectar a la base de datos
    conn = pq_connect(setdbopts('dbname', 'primerParcial', 'host', 'localhost', 'port', '5432', 'user', 'postgres', 'password', 'cindy123451'));
    disp('Conexi n a la base de datos establecida.');
catch ME
    error('No se pudo conectar a la base de datos: %s', ME.message);
end

function menu()
  opcion = 0;
  while opcion ~= 10
    fprintf('\n\t\t Bienvenido al Men Principal!\n');
    fprintf('1. Ingresar nombre usuario\n');
    fprintf('2. Calcular factorial\n');
    fprintf('3. Sistema login\n');
    fprintf('4. Calcular distancia entre dos puntos\n');
    fprintf('5. Calcular IMC\n');
    fprintf('6. Sueldo base semanal\n');
    fprintf('7. Visualizar archivo de texto\n');
    fprintf('8. Historial de datos\n');
    fprintf('9. Borrar datos\n');
    fprintf('10. Salir\n');

  try
    opcion = input('Seleccione una opci n: ');
    if isempty(opcion) || ~isnumeric(opcion) || opcion < 1 || opcion > 10
      error('Entrada inv lida. Por favor, ingrese un n mero entre 1 y 10.');
    endif
  catch ME
    fprintf('Error: %s\n', ME.message);
    continue;
  end

        switch opcion
            case 1  %ingreso de nombre usuario
              ingresoDatos();

            case 2 %calcular factorial
              calculo_factorial();

            case 3 %Sistema login
              sistemaLogin();
            case 4 %calcular distancia entre dos puntos
              calcular_distancia();
```

[*] Proyectos de Computación Aplicados a la Ingeniería Electrónica.

[**] e-mail: 2787947930101@ingenieria.usac.edu.gt

```
            case 5 %calcular imc
              calcular_imc();
            case 6 %sueldo base semanal
              calcular_nomina();
            case 7 %Visualizar datos archivo de texto
                registrar_operacion();
            case 8 %Historial de datos
                mostrar_historial();
            case 9 %borrar datos
                borrar_datos();
                #mostrar_resultados();
            case 10 %salir
                fprintf('  Gracias   por visitarnos! Vuelva pronto.\n');
            otherwise
                fprintf('Opci n no v lida. Intente nuevamente.\n');
        end
    end
end


function ingresoDatos()
    global resultados;
    try
      resultados.nombre = input('Ingrese el nombre del usuario: ', 's');
      if isempty(resultados.nombre)
            error('El nombre no puede estar vac o.');
      end
        fprintf('Datos ingresados correctamente.\n');
    catch ME
        fprintf('Error: %s\n', ME.message);
    end
end

function calculo_factorial()
  global resultados;
  try
    while true
      % Pedir al usuario que ingrese un n mero
      resultados.tipo_programa1 = 'Factorial';
      resultados.x = input('Ingresa un n mero entero positivo: ');

      % Calcular el factorial usando la funci n calcularFactorial
      if( ~isnumeric(resultados.x) || resultados.x < 0 || mod(resultados.x,1) != 0)
          disp('Error: El numero ingresado debe ser entero positivo. Intenta de nuevo.'); % si no es entero positivo, se encicla de nuevo
      elseif(isempty(resultados.x))
        disp('Error: no ingresaste nada. Intenta de nuevo'); %si no ingresa nada. Vuelve al ciclo
      else
          break; %si ingresa un entero positivo sale del ciclo
      end
    end

    if (resultados.x == 0 || resultados.x == 1)
          resultados.factorial = 1;

      elseif(resultados.x >1)
          resultados.factorial = 1;

          for (i = 2:resultados.x)
              resultados.factorial = resultados.factorial * i;
          end
      end

      % Mostrar el resultado
      fprintf('El factorial de %d es %d\n', resultados.x, resultados.factorial);

  catch e
      % Manejar el error si el usuario ingresa un n mero negativo o no entero
      fprintf('Ocurri  un error. Error: %s\n', e.message);

  end
end

function sistemaLogin()
  global resultados;
% Inicializar el n mero de intentos
  resultados.intentos = 3;

  while resultados.intentos > 0
      resultados.tipo_programa2 = 'Login';
      % Solicitar el nombre de usuario y la contrase a
      resultados.usuario = input('Ingrese el nombre de usuario: ', 's');
      resultados.contrasena = input('Ingrese la contrase a: ', 's');

      % Verificar las credenciales
      if strcmp(resultados.usuario, 'admin') && strcmp(resultados.contrasena, '123')
          disp('  Inicio   de  sesi n exitoso!');
          break; % Salir del loop principal
      else
          % Mostrar el n mero de intentos restantes
          resultados.intentos = resultados.intentos - 1;
          fprintf('Credenciales incorrectas. Te quedan %d intentos.\n', resultados.intentos);

          % Cerrar el programa si se agotan los intentos
          if resultados.intentos == 0
              disp('Has agotado todos los intentos. Cerrando el programa.');
              return;
          end
      end
  end
end

function calcular_distancia()
  global resultados;
  try
    % Solicitar las coordenadas del primer punto
    resultados.tipo_programa3 = 'Distancia';
    fprintf('Ingrese las coordenadas del primer punto (x1, y1):\n');
    resultados.x1 = input('x1 = ');
    resultados.y1 = input('y1 = ');
```

```
    % Validar si las entradas son num ricas
    if ~isnumeric(resultados.x1) || ~isnumeric(resultados.y1) || isempty(resultados.x1) || isempty(resultados.y1)
      error('Las coordenadas deben ser valores num ricos.');
    end

    % Solicitar las coordenadas del segundo punto
    fprintf('\nIngrese las coordenadas del segundo punto (x2, y2):\n');
    resultados.x2 = input('x2 = ');
    resultados.y2 = input('y2 = ');

    % Validar si las entradas son num ricas
    if ~isnumeric(resultados.x2) || ~isnumeric(resultados.y2) || isempty(resultados.x2) || isempty(resultados.y2)
      error('Las coordenadas deben ser valores num ricos.');
    end

    % Calcular la distancia usando la f rmula de Pit goras
    resultados.distancia = sqrt((resultados.x2 - resultados.x1)^2 + (resultados.y2 - resultados.y1)^2);

    % Mostrar el resultado
    fprintf('\nLa distancia m s corta entre los puntos (%d, %d) y (%d, %d) es: %.4f\n', ...
            resultados.x1, resultados.y1, resultados.x2, resultados.y2, resultados.distancia);

  catch ME
    % Manejo de errores
    fprintf('Error: %s\n', ME.message);
  end_try_catch
end

function calcular_imc()
  global resultados;
  try
    resultados.tipo_programa4 = 'IMC';
    % Solicitar al usuario su sexo (hombre/mujer)
    resultados.genero = input('Ingrese su genero ("hombre" o "mujer"): ', 's');
    resultados.genero = lower(strtrim(resultados.genero)); % Convertir a min sculas y quitar espacios

    % Validar el ingreso del sexo
    if ~strcmp(resultados.genero, 'hombre') && ~strcmp(resultados.genero, 'mujer')
      error('Debe ingresar "hombre" o "mujer".');
    end

    % Solicitar el peso
    resultados.peso = input('Ingrese su peso en kilogramos (kg): ');
    if isempty(resultados.peso) || ~isnumeric(resultados.peso) || resultados.peso <= 0
      error('El peso debe ser un valor num rico positivo.');
    end

    % Solicitar la altura
    resultados.altura = input('Ingrese su altura en metros (m): ');
    if isempty(resultados.altura) || ~isnumeric(resultados.altura) || resultados.altura <= 0
      error('La altura debe ser un valor num rico positivo.');
    end

    % Calcular el IMC
    resultados.imc = resultados.peso / (resultados.altura^2);

    % Mostrar el resultado del IMC
    fprintf('\nSu  ndice  de Masa Corporal (IMC) es: %.2f\n', resultados.imc);

    % Determinar la conducta a seguir seg n el IMC y sexo
    if strcmp(resultados.genero, 'mujer')
      if resultados.imc < 18.5
        fprintf('Conducta a seguir: Bajo peso, consulte a su m dico.\n');
      elseif resultados.imc >= 18.5 && resultados.imc < 24.9
        fprintf('Conducta a seguir: Peso normal, contin e con buenos h bitos.\n');
      elseif resultados.imc >= 25 && resultados.imc < 29.9
        fprintf('Conducta a seguir: Sobrepeso, considere una rutina de ejercicios.\n');
      else
        fprintf('Conducta a seguir: Obesidad, consulte a su m dico para un plan adecuado.\n');
      end
    elseif strcmp(resultados.genero, 'hombre')
      if resultados.imc < 18.5
        fprintf('Conducta a seguir: Bajo peso, consulte a su m dico.\n');
      elseif resultados.imc >= 18.5 && resultados.imc < 24.9
        fprintf('Conducta a seguir: Peso normal, contin e con buenos h bitos.\n');
      elseif resultados.imc >= 25 && resultados.imc < 29.9
        fprintf('Conducta a seguir: Sobrepeso, considere una rutina de ejercicios.\n');
      else
        fprintf('Conducta a seguir: Obesidad, consulte a su m dico para un plan adecuado.\n');
      end
    end

  catch ME
    % Manejo de errores
    fprintf('Error: %s\n', ME.message);
  end_try_catch
end


function calcular_nomina()
  global resultados;
  % Constante para el precio por hora
  PRECIO_HORA = 6;

  try
    % Solicitar horas de trabajo
```

```
    resultados.tipo_programa5 = 'Sueldo semanal';
    resultados.horas_trabajo = input('Ingrese las horas de trabajo semanales: ');
    if isempty(resultados.horas_trabajo) || ~isnumeric(resultados.horas_trabajo) || resultados.horas_trabajo < 0
      error('Las horas de trabajo deben ser un valor num rico positivo.');
    end

    % Solicitar horas extras trabajadas
    resultados.horas_extras = input('Ingrese las horas extra semanales: ');
    if isempty(resultados.horas_extras) || ~isnumeric(resultados.horas_extras) || resultados.horas_extras < 0
      error('Las horas extra deben ser un valor num rico positivo.');
    end

    % Calcular precio por hora extra
    if resultados.horas_extras < 10
      resultados.precio_hora_extra = PRECIO_HORA * 1.5; % 50% mayor
    elseif resultados.horas_extras >= 10 && resultados.horas_extras <= 20
      resultados.precio_hora_extra = PRECIO_HORA * 1.4; % 40% mayor
    else
      resultados.precio_hora_extra = PRECIO_HORA * 1.2; % 20% mayor
    end

    % Calcular el sueldo base semanal
    resultados.sueldo_base = (resultados.horas_trabajo * PRECIO_HORA) + (resultados.horas_extras * resultados.precio_hora_extra);

    % Mostrar los resultados
    fprintf('\nResumen de n mina:\n');
    fprintf('Horas trabajadas: %d\n', resultados.horas_trabajo);
    fprintf('Horas extra: %d\n', resultados.horas_extras);
    fprintf('Precio por hora: %.2f\n', PRECIO_HORA);
    fprintf('Precio por hora extra: %.2f\n', resultados.precio_hora_extra);
    fprintf('Sueldo base semanal: %.2f\n', resultados.sueldo_base);

  catch ME
    % Manejo de errores
    fprintf('Error: %s\n', ME.message);
  end_try_catch
end

function registrar_operacion()
    #global resultados;
    global resultados conn;
    try

        % Generar el texto para el archivo
        fecha_actual = datestr(now, 'yyyy-mm-dd HH:MM:SS');
        registro_txt = sprintf([...
            '
    ----------------------------------------------\n', ...
            'Fecha: %s\n', ...
            'Usuario: %s\n', ...
            'Tipo de programa: %s\n', ...
            'Numero a calcular factorial: %s\n', ...
            'Resultado de factorial: %s\n', ...
            '
    ----------------------------------------------\n', ...
            'Tipo de programa: %s\n', ...
            'Intentos disponibles: %s\n', ...
            'Usuario: %s\n', ...
            'Password: %s\n', ...
            '
    ----------------------------------------------\n', ...
            'Tipo de programa: %s\n', ...
            'x1: %s\n', ...
            'y1: %s\n', ...
            'x2: %s\n', ...
            'y2: %s\n', ...
            'distancia entre coordenadas: %s\n', ...
            '
    ----------------------------------------------\n', ...
            'Tipo de programa: %s\n', ...
            'Genero: %s\n', ...
            'Peso: %s\n', ...
            'Altura: %s\n', ...
            'IMC: %s\n', ...
            '
    ----------------------------------------------\n', ...
            'Tipo de programa: %s\n', ...
            'Horas trabajadas: %s\n', ...
            'Horas extras trabajadas: %s\n', ...
            'Precio horas extras trabajadas: %s\n', ...
            'Sueldo por semana: %s\n', ...
            '
    ----------------------------------------------\n'], ...
            fecha_actual, resultados.nombre, ...
            resultados.tipo_programa1, num2str(resultados.x), num2str(resultados.factorial), ...
            resultados.tipo_programa2, num2str(resultados.intentos), resultados.usuario, resultados.contrasena, ...
            resultados.tipo_programa3, num2str(resultados.x1), num2str(resultados.y1), num2str(resultados.x2), num2str(resultados.y2), num2str(resultados.distancia), ...
            resultados.tipo_programa4, resultados.genero, num2str(resultados.peso), num2str(resultados.altura), num2str(resultados.imc), ...
            resultados.tipo_programa5, num2str(resultados.horas_trabajo), num2str(resultados.horas_extras), num2str(resultados.precio_hora_extra), num2str(resultados.sueldo_base));

        fprintf('Archivo salida.txt generado:\n');
        fprintf('%s\n', registro_txt);

        % Guardar en el archivo txt
```

```
        archivo = 'C:\Users\Melissa A\Documents\Cursos 2do semestre 2024\proyectos IE\PrimerParcial\salida.txt';
        fid = fopen(archivo, 'a');
        fprintf(fid, '%s', registro_txt);
        fclose(fid);
        disp('Operaci n guardada en el archivo de texto.');

        % Guardar en la base de datos factorial
        try
          query = "INSERT INTO factorial (nombre, numero, factorial) VALUES ($1, $2, $3)";
          params = {resultados.nombre, resultados.x, resultados.factorial};
          pq_exec_params(conn, query, params);
          disp('Datos guardados en la base de datos factorial.');
        catch ME
            fprintf('Error al guardar en la base de datos factorial: %s\n', ME.message);
        end

        % Guardar en la base de datos login
        try
          query = "INSERT INTO login (nombre, intentos, usuario, contrase a) VALUES ($1, $2, $3, $4)";
          params = {resultados.nombre, resultados.intentos, resultados.usuario, resultados.contrasena};
          pq_exec_params(conn, query, params);
          disp('Datos guardados en la base de datos login.');
        catch ME
            fprintf('Error al guardar en la base de datos login: %s\n', ME.message);
        end

        % Guardar en la base de datos distancia
        try
          query = "INSERT INTO distancia (nombre, x1, y1, x2, y2, distancia) VALUES ($1, $2, $3, $4, $5, $6)";
          params = {resultados.nombre, resultados.x1, resultados.y1, resultados.x2, resultados.y2, resultados.distancia};
          pq_exec_params(conn, query, params);
          disp('Datos guardados en la base de datos distancia.');
        catch ME
            fprintf('Error al guardar en la base de datos distancia: %s\n', ME.message);
        end

        % Guardar en la base de datos imc
        try
          query = "INSERT INTO imc (nombre, genero, peso, altura, imc) VALUES ($1, $2, $3, $4, $5)";
          params = {resultados.nombre, resultados.genero, resultados.peso, resultados.altura, resultados.imc};
          pq_exec_params(conn, query, params);
          disp('Datos guardados en la base de datos imc.');
        catch ME
            fprintf('Error al guardar en la base de datos imc: %s\n', ME.message);
        end

        % Guardar en la base de datos nominas
        try
          query = "INSERT INTO nominas (nombre, horas_trabajo, horas_extra, precio_hora_extra, sueldo_base) VALUES ($1, $2, $3, $4, $5)";
          params = {resultados.nombre, resultados.horas_trabajo, resultados.horas_extras, resultados.precio_hora_extra, resultados.sueldo_base};
          pq_exec_params(conn, query, params);
          disp('Datos guardados en la base de datos nominas.');
        catch ME
            fprintf('Error al guardar en la base de datos nominas: %s\n', ME.message);
        end

    catch ME
        fprintf('Error al registrar la operaci n: %s\n', ME.message);
    end
end

function mostrar_historial()
    global conn;

    % Obtener el historial desde la base de datos -factorial
    try
        historial_factorial = pq_exec_params(conn, "SELECT * FROM factorial;");
        disp('--------------------------------------------------');
        disp('Historial de operaciones desde la base de datos:');
        disp(historial_factorial);
        disp('--------------------------------------------------');
    catch ME
        fprintf('Error al obtener el historial factorial: %s\n', ME.message);
    end

 % Obtener el historial desde la base de datos -Login
    try
        historial_login = pq_exec_params(conn, "SELECT * FROM login;");
        disp('--------------------------------------------------');
        disp('Historial de operaciones desde la base de datos:');
        disp(historial_login);
        disp('--------------------------------------------------');
    catch ME
        fprintf('Error al obtener el historial login: %s\n', ME.message);
    end

 % Obtener el historial desde la base de datos -distancia
    try
```

```
        historial_distancia = pq_exec_params(conn, "SELECT * FROM distancia;");
        disp('----------');
        disp('Historial de operaciones desde la base de datos:');
        disp(historial_distancia);
        disp('----------');
    catch ME
        fprintf('Error al obtener el historial distancia: %s\n', ME.message);
    end

% Obtener el historial desde la base de datos - imc
    try
        historial_imc = pq_exec_params(conn, "SELECT * FROM imc;");
        disp('----------');
        disp('Historial de operaciones desde la base de datos:');
        disp(historial_imc);
        disp('----------');
    catch ME
        fprintf('Error al obtener el historial imc: %s\n', ME.message);
    end

% Obtener el historial desde la base de datos - nominas
    try
        historial_nominas = pq_exec_params(conn, "SELECT * FROM nominas;");
        disp('----------');
        disp('Historial de operaciones desde la base de datos:');
        disp(historial_nominas);
        disp('----------');
    catch ME
        fprintf('Error al obtener el historial nominas: %s\n', ME.message);
    end
end

function borrar_datos()
    global conn;
    try
        pq_exec_params(conn, "delete from factorial;");
        pq_exec_params(conn, "delete from login;");
        pq_exec_params(conn, "delete from distancia;");
        pq_exec_params(conn, "delete from imc;");
        pq_exec_params(conn, "delete from nominas;");
        fprintf('Todos los registros de la base de datos han sido eliminados.\n');
    catch ME
        fprintf('Error al borrar los datos de la base de datos: %s\n', ME.message);
    end
end

function mostrar_resultados()
    global resultados;
    disp('Valores actuales de resultados:');
    disp(resultados);
end

menu();
pq_close(conn);
```

Listing 1: Programa primer parcial

III. CÓDIGO IMPLEMENTADO EN PYTHON

```
import psycopg2
from psycopg2 import sql
import math
import datetime
import getpass

# Variables globales para almacenar los datos del cliente
resultados = {}

# Men  principal
def menu():
    opcion = 0
    while opcion != 10:
        print("\n\t\t Bienvenido al Men Principal!")
        print("1. Ingresar nombre usuario")
        print("2. Calcular factorial")
        print("3. Sistema login")
        print("4. Calcular distancia entre dos puntos")
        print("5. Calcular IMC")
        print("6. Sueldo base semanal")
        print("7. Visualizar archivo de texto")
        print("8. Historial de datos")
        print("9. Borrar datos")
        print("10. Salir")
        try:
            opcion = int(input("Seleccione una opci n: "))
            if opcion == 1:
                ingreso_datos()
            elif opcion == 2:
                calcular_factorial()
            elif opcion == 3:
                sistema_login()
            elif opcion == 4:
                calcular_distancia()
            elif opcion == 5:
                calcular_imc()
            elif opcion == 6:
                calcular_nomina()
            elif opcion == 7:
                visualizar_archivo()
            elif opcion == 8:
                #imprimir_resultados()
```

```
                mostrar_historial()
            elif opcion == 9:
                borrar_datos()
            elif opcion == 10:
                print(" Gracias  por visitarnos! Vuelva pronto.")
            else:
                print("Opci n no v lida. Intente nuevamente.")
        except ValueError:
            print("Entrada inv lida. Por favor, ingrese un n mero entre 1 y 9.")

# Funci n para ingresar nombre
def ingreso_datos():
    global resultados
    try:
        nombre = input("Ingrese el nombre del usuario: ").strip()
        if not nombre:
            raise ValueError("El nombre no puede estar vac o.")
        print(f"Datos ingresados correctamente: {nombre}")

        resultados["nombre"] = nombre #guarde nombre en variable global resultados
        #print(f"Nombre guardado: {resultados['nombre']}")
    except Exception as e:
        print(f"Error: {e}")

# Calcular factorial
def calcular_factorial():
    global resultados
    try:
        while True:
            x = int(input("Ingresa un n mero entero positivo: "))
            resultados["x"] = x

            if x < 0:
                print("Error: El n mero debe ser positivo.")
            else:
                break
        factorial = math.factorial(x)
        print(f"El factorial de {x} es {factorial}")
        #almacenar datos en la variable global
        resultados["factorial"] = factorial
    except ValueError:
        print("Error: Debe ingresar un n mero entero.")

# Sistema login
def sistema_login():
    global resultados
    intentos = 3
    try:
        while intentos > 0:
            usuario = input("Ingrese el nombre de usuario: ")
            resultados["usuario"] = usuario
            contrasena = getpass.getpass("Introduce tu contrase a: ")
            resultados["Contrasena"] = contrasena
            if usuario == "admin" and contrasena == "123":
                print(" Inicio  de sesi n exitoso!")
                return
            else:
                intentos -= 1
                print(f"Credenciales incorrectas. Intentos restantes: {intentos}")
                if intentos == 0:
                    print("Has agotado todos los intentos. Cerrando el programa.")
                    return
            resultados["intentos"] = intentos
    except Exception as e:
        print(f"Error: {e}")

# Calcular distancia
def calcular_distancia():
    global resultados
    try:
        x1 = float(input("x1 = "))
        y1 = float(input("y1 = "))
        x2 = float(input("x2 = "))
        y2 = float(input("y2 = "))
        resultados["x1"] = x1
        resultados["y1"] = y1
        resultados["x2"] = x2
        resultados["y2"] = y2
        distancia = math.sqrt((x2 - x1)**2 + (y2 - y1)**2)
        resultados["distancia"] = distancia
        print(f"La distancia entre ({x1}, {y1}) y ({x2}, {y2}) es: {distancia:.4f}")
    except ValueError:
        print("Error: Las coordenadas deben ser valores num ricos.")

# Calcular IMC
def calcular_imc():
    global resultados
    try:
        genero = input("Ingrese su g nero ('hombre' o 'mujer'): ").strip().lower()
        resultados["genero"] = genero
        if genero not in ["hombre", "mujer"]:
            raise ValueError("Debe ingresar 'hombre' o 'mujer'.")
        peso = float(input("Ingrese su peso (kg): "))
        altura = float(input("Ingrese su altura (m): "))
        resultados["peso"] = peso
        resultados["altura"] = altura
        if peso <= 0 or altura <= 0:
            raise ValueError("Peso y altura deben ser positivos.")
        imc = peso / (altura ** 2)
        resultados["imc"] = imc
        print(f"Su IMC es: {imc:.2f}")
        if imc < 18.5:
            print("Bajo peso, consulte a su m dico.")
        elif imc < 24.9:
            print("Peso normal, contin e con buenos h bitos.")
        elif imc < 29.9:
            print("Sobrepeso, considere una rutina de ejercicios.")
        else:
            print("Obesidad, consulte a su m dico para un plan adecuado.")
```

```
    except ValueError as e:
        print(f"Error: {e}")

# Calcular n mina
def calcular_nomina():
    global resultados
    PRECIO_HORA = 6
    try:
        horas_trabajo = int(input("Ingrese horas trabajadas: "))
        horas_extras = int(input("Ingrese horas extras: "))
        resultados["horas_trabajo"] = horas_trabajo
        resultados["horas_extras"] = horas_extras
        if horas_extras < 10:
            precio_extra = PRECIO_HORA * 1.5
        elif horas_extras <= 20:
            precio_extra = PRECIO_HORA * 1.4
        else:
            precio_extra = PRECIO_HORA * 1.2
        sueldo = (horas_trabajo * PRECIO_HORA) + (horas_extras * precio_extra)
        resultados["precio_extra"] = precio_extra
        resultados["sueldo"] = sueldo
        print(f"Sueldo base semanal: {sueldo:.2f}")
    except ValueError:
        print("Error: Ingrese valores num ricos.")

# Visualizar archivos txt
def visualizar_archivo():
    global resultados
    try:
 # Acceder a los valores de resultados correctamente
        salida = (
            "
    ----------------------------------------------\n"
            f"Nombre: {resultados.get('nombre', 'N/A')}\n"
            f"Tipo de operacion: Funcion factorial\n"
            f"x: {resultados.get('x', 'N/A')}\n"
            f"Factorial: {resultados.get('factorial', 'N/A')}\n"
            f"
    ----------------------------------------------\n"
            f"Tipo de operacion: Login\n"
            f"Intentos: {resultados.get('intentos', 'N/A')} \n"
            f"Usuario: {resultados.get('usuario', 'N/A')} \n"
            f"Contrase a: {resultados.get('contrasena', 'N/A')} \n"
            f"
    ----------------------------------------------\n"
            f"Tipo de operacion: Distancia\n"
            f"x1: {resultados.get('x1', 0):.2f}\n"
            f"y1: {resultados.get('y1', 0):.2f}\n"
            f"x2: {resultados.get('x2', 0):.2f}\n"
            f"y2: {resultados.get('y2', 0):.2f}\n"
            f"Distancia: {resultados.get('distancia', 0):.2f}\n"
            f"
    ----------------------------------------------\n"
            f"Tipo de operacion: IMC\n"
            f"G nero: {resultados.get('genero', 'N/A')}\n"
            f"Peso: {resultados.get('peso', 0):.2f}\n"
            f"Altura: {resultados.get('altura', 0):.2f}\n"
            f"IMC: {resultados.get('imc', 0):.2f}\n"
            f"
    ----------------------------------------------\n"
            f"Tipo de operacion: Sueldo Semanal\n"
            f"Horas trabajadas: {resultados.get('horas_trabajo', 0):.2f}\n"
            f"Horas extras: {resultados.get('horas_extras', 0):.2f}\n"
            f"Precio hora extra: {resultados.get('precio_extra', 0):.2f}\n"
            f"Sueldo semanal: {resultados.get('sueldo', 0):.2f}\n"
            "
    ----------------------------------------------\n"
        )

        print("Archivo txt generado:")
        print(salida)

        with open("c:/Users/Melissa A/Documents/Cursos 2do semestre 2024/proyectos IE/PrimerParcial/salidap.txt", "a") as f:
            f.write(salida)
        print("Archivo guardado en 'salidap.txt'.")
    except Exception as e:
        print(f"Error al generar la factura: {e}")

#Conectar a la base de datos
def basePosgresql():
    try:
        conn = psycopg2.connect(
            dbname="primerParcial",
            user="postgres",
            password="cindy123451",
            host="localhost",
            port="5432"
        )
        return conn
    except Exception as e:
        print("Error al conectar a la base de datos:", e)
        return None

#Insertar datos en base
def insertar_en_base_datos():
    global resultados
    Nombre = resultados["nombre"]
    Numero = resultados["x"]
    Factorial = resultados["factorial"]

```

```
    """Inserta datos en las tablas."""
    try:
        conn = basePosgresql()
        if conn:
            cursor = conn.cursor()
            # Query para insertar datos
            query = 'INSERT INTO factorial("nombre", "numero", "factorial") VALUES (%s, %s, %s)'
            cursor.execute(query, (Nombre, Numero, Factorial))
            conn.commit()
            conn.close()
            print("Datos insertados en la base de datos correctamente.")
    except Exception as e:
        print(f"Error al insertar en la base de datos: {e}")

# Mostrar historial
def mostrar_historial():
    try:
        conn = basePosgresql()
        if conn:
            cursor = conn.cursor()
            cursor.execute('SELECT * FROM factorial;')
            registros = cursor.fetchall()
            print("\n--- Historial en la base de datos ---")
            for registro in registros:
                print(registro)
            conn.close()
    except Exception as e:
        print(f"Error: {e}")

# Borrar datos
def borrar_datos():
    try:
        if os.path.exists("c:/Users/Melissa A/Documents/Cursos 2do semestre 2024/proyectos IE/PrimerParcial/salidap.txt"):
            os.remove("c:/Users/Melissa A/Documents/Cursos 2do semestre 2024/proyectos IE/PrimerParcial/salidap.txt")
            print("Archivo 'salidap.txt' eliminado.")
        else:
            print("El archivo 'salidap.txt' no existe.")

        conn = basePosgresql()
        if conn:
            cursor = conn.cursor()
            cursor.execute('DELETE FROM factorial;')
            conn.commit()
            conn.close()
            print("Datos eliminados de la base de datos.")
    except Exception as e:
        print(f"Error al borrar datos: {e}")


############################
def imprimir_resultados():
    global resultados
    if not resultados:
        print("No hay datos almacenados en resultados.")
        return
    print("\nValores almacenados en resultados:")
    for clave, valor in resultados.items():
        print(f"{clave}: {valor}")
############################

# Ejecutar el men
if __name__ == "__main__":
    menu()
```

Listing 2: primer parcial

IV. RESULTADOS EN EL EDITOR DE TEXTO

A. Octave

```
clc;
clear;
pkg load database;

%Variable para ir almacenando los resultados
global resultados conn;
resultados = struct();

try
    % Conectar a la base de datos
    conn = pq_connect(setdbopts('dbname', 'primerParcial', 'host', 'localhost'
    disp('Conexión a la base de datos establecida.');
catch ME
    error('No se pudo conectar a la base de datos: %s', ME.message);
end

function menu()
  opcion = 0;
  while opcion ~= 10
    fprintf('\n\t\t¡Bienvenido al Menú Principal!\n');
    fprintf('1. Ingresar nombre usuario\n');
    fprintf('2. Calcular factorial\n');
    fprintf('3. Sistema login\n');
    fprintf('4. Calcular distancia entre dos puntos\n');
    fprintf('5. Calcular IMC\n');
    fprintf('6. Sueldo base semanal\n');
    fprintf('7. Visualizar archivo de texto\n');
    fprintf('8. Historial de datos\n');
    fprintf('9. Borrar datos\n');
    fprintf('10. Salir\n');

  try
    opcion = input('Seleccione una opción: ');
    if isempty(opcion) || ~isnumeric(opcion) || opcion < 1 || opcion > 10
      error('Entrada inválida. Por favor, ingrese un número entre 1 y 10.');
    endif
  catch ME
    fprintf('Error: %s\n', ME.message);
    continue;
  end
```

Figura 1: Editor de texto Octave

Fuente: Elaboración propia utilizando Octave, 2024

```
        switch opcion
            case 1  %ingreso de nombre usuario
              ingresoDatos();

            case 2 %calcular factorial
              calculo_factorial();

            case 3 %Sistema login
              sistemaLogin();
            case 4 %calcular distancia entre dos puntos
              calcular_distancia();
            case 5 %calcular imc
              calcular_imc();
            case 6 %sueldo base semanal
              calcular_nomina();
            case 7 %Visualizar datos archivo de texto
                registrar_operacion();
            case 8 %Historial de datos
                mostrar_historial();
            case 9 %borrar datos
                borrar_datos();
                #mostrar_resultados();
            case 10 %salir
                fprintf('¡Gracias por visitarnos! Vuelva pronto.\n');
            otherwise
                fprintf('Opción no válida. Intente nuevamente.\n');
        end
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function ingresoDatos()
    global resultados;
    try
      resultados.nombre = input('Ingrese el nombre del usuario: ', 's');
      if isempty(resultados.nombre)
            error('El nombre no puede estar vacío.');
      end
```

Figura 2: Editor de texto Octave

Fuente: Elaboración propia utilizando Octave, 2024

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function sistemaLogin()
  global resultados;
  Inicializar el número de intentos
  resultados.intentos = 3;

  while resultados.intentos > 0
      resultados.tipo_programa2 = 'Login';
      % Solicitar el nombre de usuario y la contraseña
      resultados.usuario = input('Ingrese el nombre de usuario: ', 's');
      resultados.contrasena = input('Ingrese la contraseña: ', 's');

      % Verificar las credenciales
      if strcmp(resultados.usuario, 'admin') && strcmp(resultados.contrasena, '123')
          disp('¡Inicio de sesión exitoso!');
          break; % Salir del loop principal
      else
          % Mostrar el número de intentos restantes
          resultados.intentos = resultados.intentos - 1;
          fprintf('Credenciales incorrectas. Te quedan %d intentos.\n', resultados.intentos);

          % Cerrar el programa si se agotan los intentos
          if resultados.intentos == 0
              disp('Has agotado todos los intentos. Cerrando el programa.');
              return;
          end
      end
  end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function calcular_distancia()
  global resultados;
  try
    % Solicitar las coordenadas del primer punto
    resultados.tipo_programa3 = 'Distancia';
    fprintf('Ingrese las coordenadas del primer punto (x1, y1):\n');
    resultados.x1 = input('x1 = ');
    resultados.y1 = input('y1 = ');
```

Figura 3: Editor de texto Octave

Fuente: Elaboración propia utilizando Octave, 2024

```
global resultados;
try
  % Solicitar las coordenadas del primer punto
  resultados.tipo_programa3 = 'Distancia';
  fprintf('Ingrese las coordenadas del primer punto (x1, y1):\n');
  resultados.x1 = input('x1 = ');
  resultados.y1 = input('y1 = ');

  % Validar si las entradas son numéricas
  if ~isnumeric(resultados.x1) || ~isnumeric(resultados.y1) || isempty(resultados.x1) || isempty(resultados.y1)
    error('Las coordenadas deben ser valores numéricos.');
  end

  % Solicitar las coordenadas del segundo punto
  fprintf('\nIngrese las coordenadas del segundo punto (x2, y2):\n');
  resultados.x2 = input('x2 = ');
  resultados.y2 = input('y2 = ');

  % Validar si las entradas son numéricas
  if ~isnumeric(resultados.x2) || ~isnumeric(resultados.y2) || isempty(resultados.x2) || isempty(resultados.y2)
    error('Las coordenadas deben ser valores numéricos.');
  end

  % Calcular la distancia usando la fórmula de Pitágoras
  resultados.distancia = sqrt((resultados.x2 - resultados.x1)^2 + (resultados.y2 - resultados.y1)^2);

  % Mostrar el resultado
  fprintf('\nLa distancia más corta entre los puntos (%d, %d) y (%d, %d) es: %.4f\n', ...
          resultados.x1, resultados.y1, resultados.x2, resultados.y2, resultados.distancia);

catch ME
  % Manejo de errores
  fprintf('Error: %s\n', ME.message);
end try_catch
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function calcular_imc()
  global resultados;
  try
```

Figura 4: Editor de texto Octave

Fuente: Elaboración propia utilizando Octave, 2024

B. Python

```
import psycopg2
from psycopg2 import sql
import math
import datetime

# Conexión a la base de datos
def conectar_bd():
    try:
        conn = psycopg2.connect(
            dbname="primerParcial",
            user="postgres",
            password="cindy123451",
            host="localhost",
            port="5432"
        )
        print("Conexión exitosa a la base de datos.")
        return conn
    except Exception as e:
        print(f"Error al conectar a la base de datos: {e}")
        return None

# Menú principal
def menu():
    opcion = 0
    while opcion != 9:
        print("\n\t\t¡Bienvenido al Menú Principal!")
        print("1. Ingresar nombre usuario")
        print("2. Calcular factorial")
        print("3. Sistema login")
        print("4. Calcular distancia entre dos puntos")
        print("5. Calcular IMC")
        print("6. Sueldo base semanal")
        print("7. Historial de datos")
        print("8. Borrar datos")
        print("9. Salir")
        try:
```

Figura 5: Editor de texto Python

Fuente: Elaboración propia utilizando python, 2024

```python
def menu():
        print("8. Borrar datos")
        print("9. Salir")
        try:
            opcion = int(input("Seleccione una opción: "))
            if opcion == 1:
                ingreso_datos()
            elif opcion == 2:
                calcular_factorial()
            elif opcion == 3:
                sistema_login()
            elif opcion == 4:
                calcular_distancia()
            elif opcion == 5:
                calcular_imc()
            elif opcion == 6:
                calcular_nomina()
            elif opcion == 7:
                mostrar_historial()
            elif opcion == 8:
                borrar_datos()
            elif opcion == 9:
                print("¡Gracias por visitarnos! Vuelva pronto.")
            else:
                print("Opción no válida. Intente nuevamente.")
        except ValueError:
            print("Entrada inválida. Por favor, ingrese un número entre 1 y 9.")

# Función para ingresar nombre
def ingreso_datos():
    try:
        nombre = input("Ingrese el nombre del usuario: ").strip()
        if not nombre:
            raise ValueError("El nombre no puede estar vacío.")
        print(f"Datos ingresados correctamente: {nombre}")
```

Figura 6: Editor de texto Python
Fuente: Elaboración propia utilizando python, 2024

```python
# Calcular factorial
def calcular_factorial():
    try:
        while True:
            x = int(input("Ingresa un número entero positivo: "))
            if x < 0:
                print("Error: El número debe ser positivo.")
            else:
                break
        factorial = math.factorial(x)
        print(f"El factorial de {x} es {factorial}")
    except ValueError:
        print("Error: Debe ingresar un número entero.")

# Sistema login
def sistema_login():
    intentos = 3
    while intentos > 0:
        usuario = input("Ingrese el nombre de usuario: ")
        contrasena = input("Ingrese la contraseña: ")
        if usuario == "admin" and contrasena == "123":
            print("¡Inicio de sesión exitoso!")
            return
        else:
            intentos -= 1
            print(f"Credenciales incorrectas. Intentos restantes: {intentos}")
            if intentos == 0:
                print("Has agotado todos los intentos. Cerrando el programa.")
                return

# Calcular distancia
def calcular_distancia():
    try:
        x1 = float(input("x1 = "))
```

Figura 7: Editor de texto Python
Fuente: Elaboración propia utilizando python, 2024

```python
# Calcular IMC
def calcular_imc():
    try:
        genero = input("Ingrese su género ('hombre' o 'mujer'): ").strip().lower()
        if genero not in ["hombre", "mujer"]:
            raise ValueError("Debe ingresar 'hombre' o 'mujer'.")
        peso = float(input("Ingrese su peso (kg): "))
        altura = float(input("Ingrese su altura (m): "))
        if peso <= 0 or altura <= 0:
            raise ValueError("Peso y altura deben ser positivos.")
        imc = peso / (altura ** 2)
        print(f"Su IMC es: {imc:.2f}")
        if imc < 18.5:
            print("Bajo peso, consulte a su médico.")
        elif imc < 24.9:
            print("Peso normal, continúe con buenos hábitos.")
        elif imc < 29.9:
            print("Sobrepeso, considere una rutina de ejercicios.")
        else:
            print("Obesidad, consulte a su médico para un plan adecuado.")
    except ValueError as e:
        print(f"Error: {e}")

# Calcular nómina
def calcular_nomina():
    PRECIO_HORA = 6
    try:
        horas_trabajo = int(input("Ingrese horas trabajadas: "))
        horas_extras = int(input("Ingrese horas extras: "))
        if horas_extras < 10:
            precio_extra = PRECIO_HORA * 1.5
        elif horas_extras <= 20:
            precio_extra = PRECIO_HORA * 1.4
        else:
```

Figura 8: Editor de texto Python
Fuente: Elaboración propia utilizando python, 2024

```python
def calcular_nomina():
        print(f"Sueldo base semanal: {sueldo:.2f}")
    except ValueError:
        print("Error: Ingrese valores numéricos.")

# Mostrar historial
def mostrar_historial():
    conn = conectar_bd()
    if conn:
        try:
            with conn.cursor() as cursor:
                cursor.execute("SELECT * FROM parcial;")
                resultados = cursor.fetchall()
                for fila in resultados:
                    print(fila)
        except Exception as e:
            print(f"Error al obtener el historial: {e}")
        finally:
            conn.close()

# Borrar datos
def borrar_datos():
    conn = conectar_bd()
    if conn:
        try:
            with conn.cursor() as cursor:
                cursor.execute("DELETE FROM parcial;")
                conn.commit()
                print("Historial borrado exitosamente.")
        except Exception as e:
            print(f"Error al borrar los datos: {e}")
        finally:
            conn.close()

# Ejecutar el menú
if __name__ == "__main__":
```

Figura 9: Editor de texto Python
Fuente: Elaboración propia utilizando python, 2024

```python
def calcular_nomina():
        print(f"Sueldo base semanal: {sueldo:.2f}")
    except ValueError:
        print("Error: Ingrese valores numéricos.")

# Mostrar historial
def mostrar_historial():
    conn = conectar_bd()
    if conn:
        try:
            with conn.cursor() as cursor:
                cursor.execute("SELECT * FROM parcial;")
                resultados = cursor.fetchall()
                for fila in resultados:
                    print(fila)
        except Exception as e:
            print(f"Error al obtener el historial: {e}")
        finally:
            conn.close()

# Borrar datos
def borrar_datos():
    conn = conectar_bd()
    if conn:
        try:
            with conn.cursor() as cursor:
                cursor.execute("DELETE FROM parcial;")
                conn.commit()
                print("Historial borrado exitosamente.")
        except Exception as e:
            print(f"Error al borrar los datos: {e}")
        finally:
            conn.close()

# Ejecutar el menú
if __name__ == "__main__":
```

Figura 10: Editor de texto Python

Fuente: Elaboración propia utilizando Opython, 2024

V. RESULTADOS EN TERMINAL

A. Octave

```
Conexión a la base de datos establecida.

                ¡Bienvenido al Menú Principal!
1. Ingresar nombre usuario
2. Calcular factorial
3. Sistema login
4. Calcular distancia entre dos puntos
5. Calcular IMC
6. Sueldo base semanal
7. Visualizar archivo de texto
8. Historial de datos
9. Borrar datos
10. Salir
Seleccione una opción: 1
Ingrese el nombre del usuario: cindy
Datos ingresados correctamente.

                ¡Bienvenido al Menú Principal!
1. Ingresar nombre usuario
2. Calcular factorial
3. Sistema login
4. Calcular distancia entre dos puntos
5. Calcular IMC
6. Sueldo base semanal
7. Visualizar archivo de texto
8. Historial de datos
9. Borrar datos
10. Salir
Seleccione una opción: 2
Ingresa un número entero positivo: 6
El factorial de 6 es 720

                ¡Bienvenido al Menú Principal!
1. Ingresar nombre usuario
2. Calcular factorial
3. Sistema login
4. Calcular distancia entre dos puntos
5. Calcular IMC
6. Sueldo base semanal
7. Visualizar archivo de texto
8. Historial de datos
9. Borrar datos
10. Salir
Seleccione una opción: 3
```

Figura 11: Editor de texto Octave

Fuente: Elaboración propia utilizando Octave, 2024

```
10. Salir
Seleccione una opción: 3
Ingrese el nombre de usuario: admin
Ingrese la contraseña: 123
¡Inicio de sesión exitoso!

              ¡Bienvenido al Menú Principal!
1. Ingresar nombre usuario
2. Calcular factorial
3. Sistema login
4. Calcular distancia entre dos puntos
5. Calcular IMC
6. Sueldo base semanal
7. Visualizar archivo de texto
8. Historial de datos
9. Borrar datos
10. Salir
Seleccione una opción: 4
Ingrese las coordenadas del primer punto (x1, y1):
x1 = 1
y1 = 2

Ingrese las coordenadas del segundo punto (x2, y2):
x2 = 3
y2 = 4

La distancia más corta entre los puntos (1, 2) y (3, 4) es: 2.8284

              ¡Bienvenido al Menú Principal!
1. Ingresar nombre usuario
2. Calcular factorial
3. Sistema login
4. Calcular distancia entre dos puntos
5. Calcular IMC
6. Sueldo base semanal
7. Visualizar archivo de texto
8. Historial de datos
9. Borrar datos
10. Salir
Seleccione una opción: 5
Ingrese su genero ("hombre" o "mujer"): mujer
Ingrese su peso en kilogramos (kg): 56
Ingrese su altura en metros (m): 1.70
```

Figura 12: Editor de texto Octave

Fuente: Elaboración propia utilizando Octave, 2024

```
Su Índice de Masa Corporal (IMC) es: 19.38
Conducta a seguir: Peso normal, continúe con buenos hábitos.

              ¡Bienvenido al Menú Principal!
1. Ingresar nombre usuario
2. Calcular factorial
3. Sistema login
4. Calcular distancia entre dos puntos
5. Calcular IMC
6. Sueldo base semanal
7. Visualizar archivo de texto
8. Historial de datos
9. Borrar datos
10. Salir
Seleccione una opción: 6
Ingrese las horas de trabajo semanales: 67
Ingrese las horas extra semanales: 5

Resumen de nómina:
Horas trabajadas: 67
Horas extra: 5
Precio por hora: 6.00
Precio por hora extra: 9.00
Sueldo base semanal: 447.00

              ¡Bienvenido al Menú Principal!
1. Ingresar nombre usuario
2. Calcular factorial
3. Sistema login
4. Calcular distancia entre dos puntos
5. Calcular IMC
6. Sueldo base semanal
7. Visualizar archivo de texto
8. Historial de datos
9. Borrar datos
10. Salir
Seleccione una opción: 7
Archivo salida.txt generado:
------------------------------------------------
Fecha: 2024-12-17 13:15:37
Usuario: cindy
Tipo de programa: Factorial
```

Figura 13: Editor de texto Octave

Fuente: Elaboración propia utilizando Octave, 2024

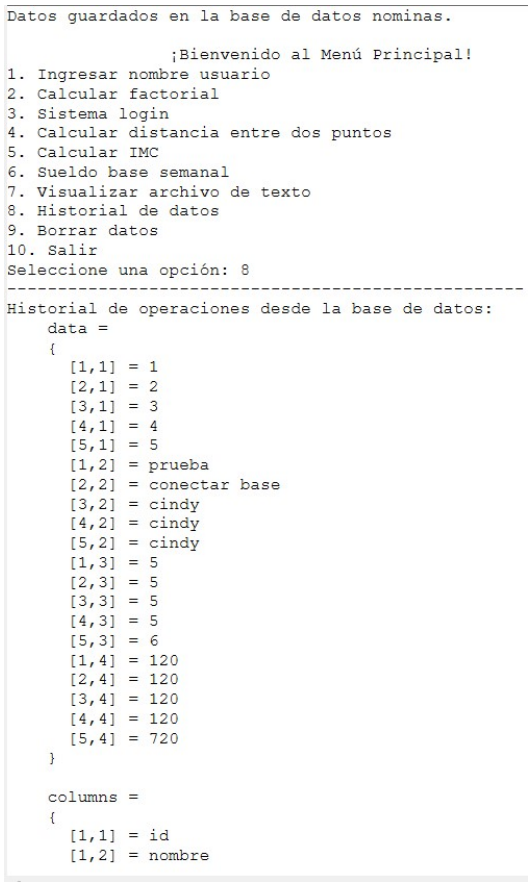
```
Resultado de factorial: 720
------------------------------------------------
Tipo de programa: Login
Intentos disponibles: 3
Usuario: admin
Password: 123
------------------------------------------------
Tipo de programa: Distancia
x1: 1
y1: 2
x2: 3
y2: 4
distancia entre coordenadas: 2.8284
------------------------------------------------
Tipo de programa: IMC
Genero: mujer
Peso: 56
Altura: 1.7
IMC: 19.3772
------------------------------------------------
Tipo de programa: Sueldo semanal
Horas trabajadas: 67
Horas extras trabajadas: 5
Precio horas extras trabajadas: 9
Sueldo por semana: 447
------------------------------------------------

Operación guardada en el archivo de texto.
Datos guardados en la base de datos factorial.
Datos guardados en la base de datos login.
Datos guardados en la base de datos distancia.
Datos guardados en la base de datos imc.
Datos guardados en la base de datos nominas.

              ¡Bienvenido al Menú Principal!
1. Ingresar nombre usuario
2. Calcular factorial
3. Sistema login
4. Calcular distancia entre dos puntos
5. Calcular IMC
6. Sueldo base semanal
7. Visualizar archivo de texto
8. Historial de datos
9. Borrar datos
```

Figura 14: Editor de texto Octave

Fuente: Elaboración propia utilizando Octave, 2024



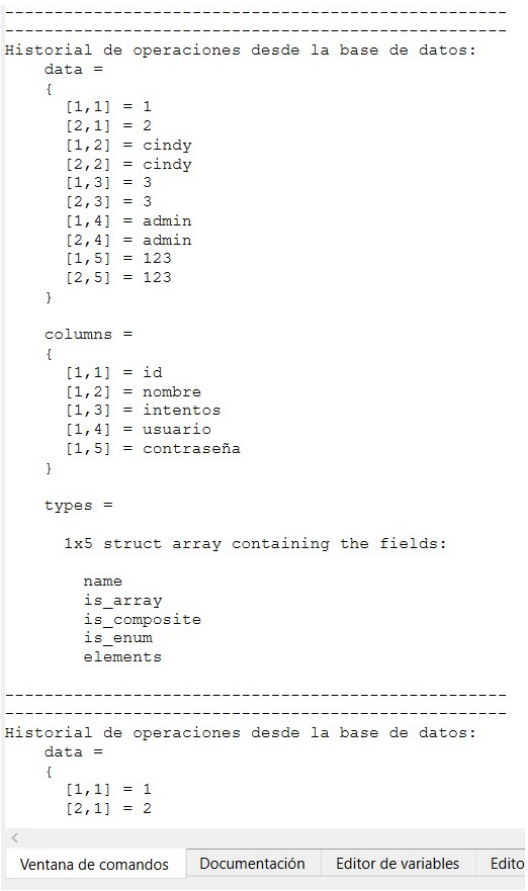
```
Datos guardados en la base de datos nominas.

               ¡Bienvenido al Menú Principal!
1. Ingresar nombre usuario
2. Calcular factorial
3. Sistema login
4. Calcular distancia entre dos puntos
5. Calcular IMC
6. Sueldo base semanal
7. Visualizar archivo de texto
8. Historial de datos
9. Borrar datos
10. Salir
Seleccione una opción: 8
------------------------------------------------------
Historial de operaciones desde la base de datos:
    data =
    {
      [1,1] = 1
      [2,1] = 2
      [3,1] = 3
      [4,1] = 4
      [5,1] = 5
      [1,2] = prueba
      [2,2] = conectar base
      [3,2] = cindy
      [4,2] = cindy
      [5,2] = cindy
      [1,3] = 5
      [2,3] = 5
      [3,3] = 5
      [4,3] = 5
      [5,3] = 6
      [1,4] = 120
      [2,4] = 120
      [3,4] = 120
      [4,4] = 120
      [5,4] = 720
    }

    columns =
    {
      [1,1] = id
      [1,2] = nombre
```

Figura 15: Editor de texto Octave

Fuente: Elaboración propia utilizando Octave, 2024



```
------------------------------------------------------
------------------------------------------------------
Historial de operaciones desde la base de datos:
    data =
    {
      [1,1] = 1
      [2,1] = 2
      [1,2] = cindy
      [2,2] = cindy
      [1,3] = 3
      [2,3] = 3
      [1,4] = admin
      [2,4] = admin
      [1,5] = 123
      [2,5] = 123
    }

    columns =
    {
      [1,1] = id
      [1,2] = nombre
      [1,3] = intentos
      [1,4] = usuario
      [1,5] = contraseña
    }

    types =

      1x5 struct array containing the fields:

        name
        is_array
        is_composite
        is_enum
        elements

------------------------------------------------------
------------------------------------------------------
Historial de operaciones desde la base de datos:
    data =
    {
      [1,1] = 1
      [2,1] = 2
```

Ventana de comandos | Documentación | Editor de variables | Edito

Figura 16: Editor de texto Octave

Fuente: Elaboración propia utilizando Octave, 2024

```
    [1,4] = y1
    [1,5] = x2
    [1,6] = y2
    [1,7] = distancia
  }

  types =

    1x7 struct array containing the fields:

      name
      is_array
      is_composite
      is_enum
      elements

-----------------------------------------------------
-----------------------------------------------------
Historial de operaciones desde la base de datos:
  data =
  {
    [1,1] = 1
    [2,1] = 2
    [1,2] = cindy
    [2,2] = cindy
    [1,3] = mujer
    [2,3] = mujer
    [1,4] = 45
    [2,4] = 56
    [1,5] = 1.6000
    [2,5] = 1.7000
    [1,6] = 17.578
    [2,6] = 19.377
  }

  columns =
  {
    [1,1] = id
    [1,2] = nombre
    [1,3] = genero
    [1,4] = peso
    [1,5] = altura
    [1,6] = imc
  }
```

Figura 17: Editor de texto Octave

Fuente: Elaboración propia utilizando Octave, 2024

```
Historial de operaciones desde la base de datos:
  data =
  {
    [1,1] = 1
    [1,2] = cindy
    [1,3] = 67
    [1,4] = 5
    [1,5] = 9
    [1,6] = 447
  }

  columns =
  {
    [1,1] = id
    [1,2] = nombre
    [1,3] = horas_trabajo
    [1,4] = horas_extra
    [1,5] = precio_hora_extra
    [1,6] = sueldo_base
  }

  types =

    1x6 struct array containing the fields:

      name
      is_array
      is_composite
      is_enum
      elements

-----------------------------------------------------

              ¡Bienvenido al Menú Principal!
1. Ingresar nombre usuario
2. Calcular factorial
3. Sistema login
4. Calcular distancia entre dos puntos
5. Calcular IMC
6. Sueldo base semanal
7. Visualizar archivo de texto
8. Historial de datos
9. Borrar datos
10. Salir
```

Figura 18: Editor de texto Octave

Fuente: Elaboración propia utilizando Octave, 2024

B. Python

```
PS C:\Users\Melissa A> & "C:/Users/Melissa A/AppData/Local/Program

              ¡Bienvenido al Menú Principal!
1. Ingresar nombre usuario
2. Calcular factorial
3. Sistema login
4. Calcular distancia entre dos puntos
5. Calcular IMC
6. Sueldo base semanal
7. Historial de datos
8. Borrar datos
9. Salir
Seleccione una opción: 1
Ingrese el nombre del usuario: Cindy
Datos ingresados correctamente: Cindy

              ¡Bienvenido al Menú Principal!
1. Ingresar nombre usuario
2. Calcular factorial
3. Sistema login
4. Calcular distancia entre dos puntos
5. Calcular IMC
6. Sueldo base semanal
7. Historial de datos
8. Borrar datos
9. Salir
Seleccione una opción: 2
Ingresa un número entero positivo: 5
```

Figura 19: Editor de texto Python

Fuente: Elaboración propia utilizando python, 2024

```
El factorial de 5 es 120

                ¡Bienvenido al Menú Principal!
1. Ingresar nombre usuario
2. Calcular factorial
3. Sistema login
4. Calcular distancia entre dos puntos
5. Calcular IMC
6. Sueldo base semanal
7. Historial de datos
8. Borrar datos
9. Salir
Seleccione una opción: 3
Ingrese el nombre de usuario: Kan
Ingrese la contraseña: 456
Credenciales incorrectas. Intentos restantes: 2
Ingrese el nombre de usuario: Admin
Ingrese la contraseña: 123
Credenciales incorrectas. Intentos restantes: 1
Ingrese el nombre de usuario: admin
Ingrese la contraseña: 123
¡Inicio de sesión exitoso!

                ¡Bienvenido al Menú Principal!
1. Ingresar nombre usuario
2. Calcular factorial
3. Sistema login
4. Calcular distancia entre dos puntos
```

Figura 20: Editor de texto Python

Fuente: Elaboración propia utilizando python, 2024

```
                ¡Bienvenido al Menú Principal!
1. Ingresar nombre usuario
2. Calcular factorial
3. Sistema login
4. Calcular distancia entre dos puntos
5. Calcular IMC
6. Sueldo base semanal
7. Historial de datos
8. Borrar datos
9. Salir
Seleccione una opción: 4
x1 = 1
y1 = 2
x2 = 3
y2 = 4
La distancia entre (1.0, 2.0) y (3.0, 4.0) es: 2.8284

                ¡Bienvenido al Menú Principal!
1. Ingresar nombre usuario
2. Calcular factorial
3. Sistema login
4. Calcular distancia entre dos puntos
5. Calcular IMC
6. Sueldo base semanal
7. Historial de datos
8. Borrar datos
```

Figura 21: Editor de texto Python

Fuente: Elaboración propia utilizando python, 2024

```
4. Calcular distancia entre dos puntos
5. Calcular IMC
6. Sueldo base semanal
7. Historial de datos
8. Borrar datos
9. Salir
Seleccione una opción: 5
Ingrese su género ('hombre' o 'mujer'): mujer
Ingrese su peso (kg): 45
Ingrese su altura (m): 1.60
Su IMC es: 17.58
Bajo peso, consulte a su médico.

                ¡Bienvenido al Menú Principal!
1. Ingresar nombre usuario
2. Calcular factorial
3. Sistema login
4. Calcular distancia entre dos puntos
5. Calcular IMC
6. Sueldo base semanal
7. Historial de datos
8. Borrar datos
9. Salir
Seleccione una opción: 6
Ingrese horas trabajadas: 85
Ingrese horas extras:
Error: Ingrese valores numéricos.
```

Figura 22: Editor de texto Python

Fuente: Elaboración propia utilizando python, 2024

VI. RESULTADO BASE DE DATOS Y TXT

	id [PK] integer	nombre character varying (20)	numero integer	factorial integer
1	1	prueba	5	120
2	2	conectar base	5	120
3	3	cindy	5	120
4	4	cindy	5	120
5	5	cindy	6	720

Figura 23: PostgreSQL - tabla factorial

Fuente: Elaboración propia utilizando Octave, 2024

	id [PK] integer	nombre character varying (20)	intentos integer	usuario character varying (20)	contraseña character varying (20)
1	1	cindy	3	admin	123
2	2	cindy	3	admin	123

Figura 24: PostgreSQL - tabla login

Fuente: Elaboración propia utilizando Octave, 2024

	id [PK] integer	nombre character varying (20)	x1 real	y1 real	x2 real	y2 real	distancia real
1	1	cindy	1	2	3	4	2.828427
2	2	cindy	1	2	3	4	2.828427

Figura 25: PostgreSQL - tabla distancia

Fuente: Elaboración propia utilizando Octave, 2024

	id [PK] integer	nombre character varying (20)	genero character varying (20)	peso real	altura real	imc real
1	1	cindy	mujer	45	1.6	17.578125
2	2	cindy	mujer	56	1.7	19.377163

Figura 26: PostgreSQL - tabla imc
Fuente: Elaboración propia utilizando Octave, 2024

	id [PK] integer	nombre character varying (20)	horas_trabajo integer	horas_extra integer	precio_hora_extra real	sueldo_base real
1	1	cindy	67	5	9	447

Figura 27: PostgreSQL - tabla nominas
Fuente: Elaboración propia utilizando Octave, 2024

```
------------------------------------------------
Tipo de programa: Sueldo semanal
Horas trabajadas: 56
Horas extras trabajadas: 3
Precio horas extras trabajadas: 9
Sueldo por semana: 363
------------------------------------------------
------------------------------------------------
Fecha: 2024-12-16 23:15:37
Usuario: cindy
Tipo de programa: Factorial
Numero a calcular factorial: 6
Resultado de factorial: 720
------------------------------------------------
Tipo de programa: Login
Intentos disponibles: 3
Usuario: admin
Password: 123
------------------------------------------------
Tipo de programa: Distancia
x1: 1
y1: 2
x2: 3
y2: 4
distancia entre coordenadas: 2.8284
------------------------------------------------
Tipo de programa: IMC
Genero: mujer
Peso: 56
Altura: 1.7
IMC: 19.3772
------------------------------------------------
Tipo de programa: Sueldo semanal
Horas trabajadas: 67
Horas extras trabajadas: 5
Precio horas extras trabajadas: 9
Sueldo por semana: 447
------------------------------------------------
```

Figura 28: salida.txt
Fuente: Elaboración propia utilizando Octave, 2024

```
------------------------------------------------
Tipo de programa: Sueldo semanal
Horas trabajadas: 54
Horas extras trabajadas: 4
Precio horas extras trabajadas: 9
Sueldo por semana: 360
------------------------------------------------
------------------------------------------------
Fecha: 2024-12-16 20:00:27
Usuario: cindy
Tipo de programa: Factorial
Numero a calcular factorial: 5
Resultado de factorial: 120
------------------------------------------------
Tipo de programa: Login
Intentos disponibles: 3
Usuario: admin
Password: 123
------------------------------------------------
Tipo de programa: Distancia
x1: 1
y1: 2
x2: 3
y2: 4
distancia entre coordenadas: 2.8284
------------------------------------------------
Tipo de programa: IMC
Genero: mujer
Peso: 45
Altura: 1.6
IMC: 17.5781
------------------------------------------------
Tipo de programa: Sueldo semanal
Horas trabajadas: 56
Horas extras trabajadas: 3
Precio horas extras trabajadas: 9
Sueldo por semana: 363
------------------------------------------------
------------------------------------------------
```

Figura 29: salida.txt
Fuente: Elaboración propia utilizando Octave, 2024

VII. DIAGRAMAS DE FLUJO

Se muestran los diagramas con la lógica aplicada

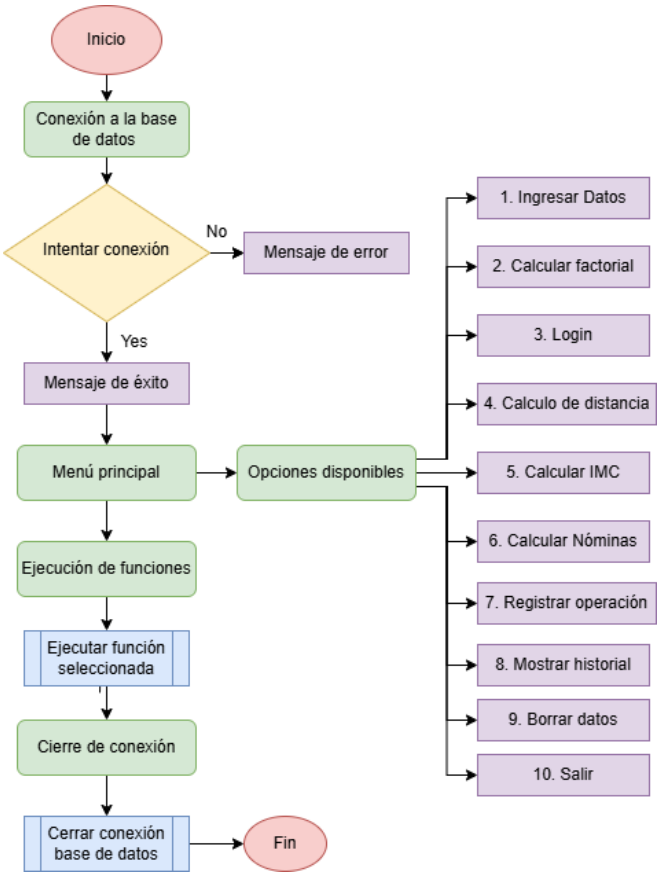


Figura 30: Diagrama de flujo para la logica general
Fuente: Elaboración propia utilizando drawio, 2024

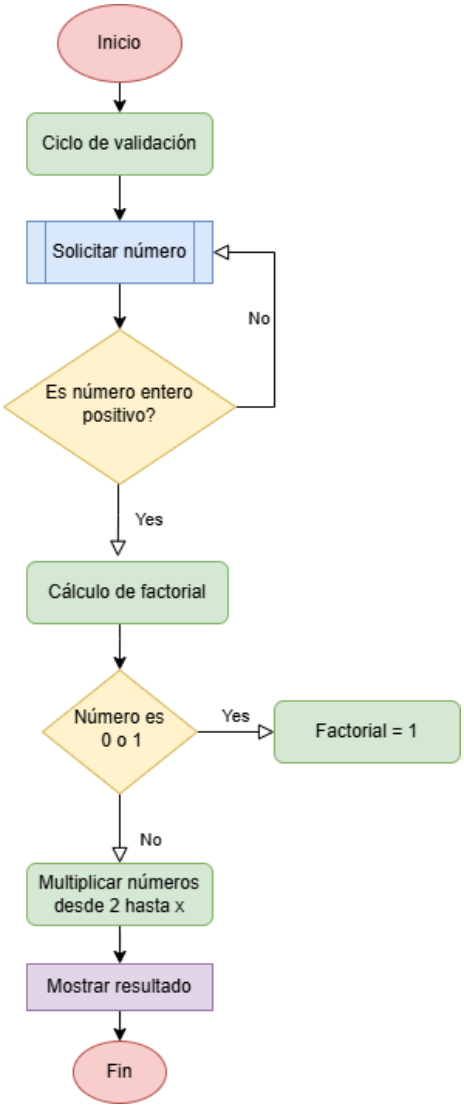


Figura 31: Diagrama de flujo para la función factorial

Fuente: Elaboración propia utilizando drawio, 2024

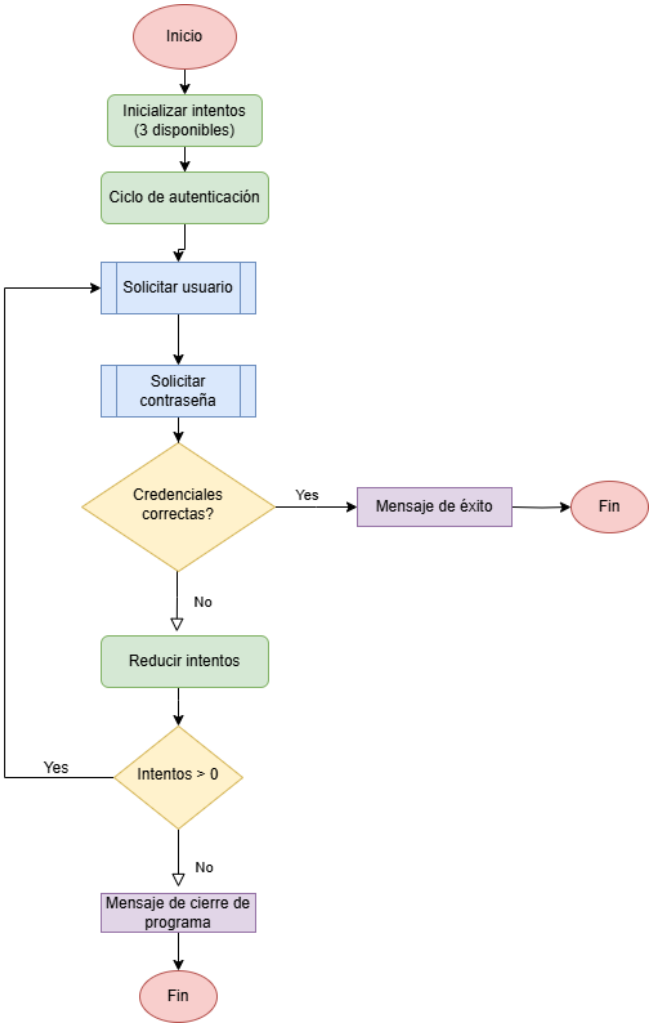


Figura 32: Diagrama de flujo para la función login

Fuente: Elaboración propia utilizando drawio, 2024

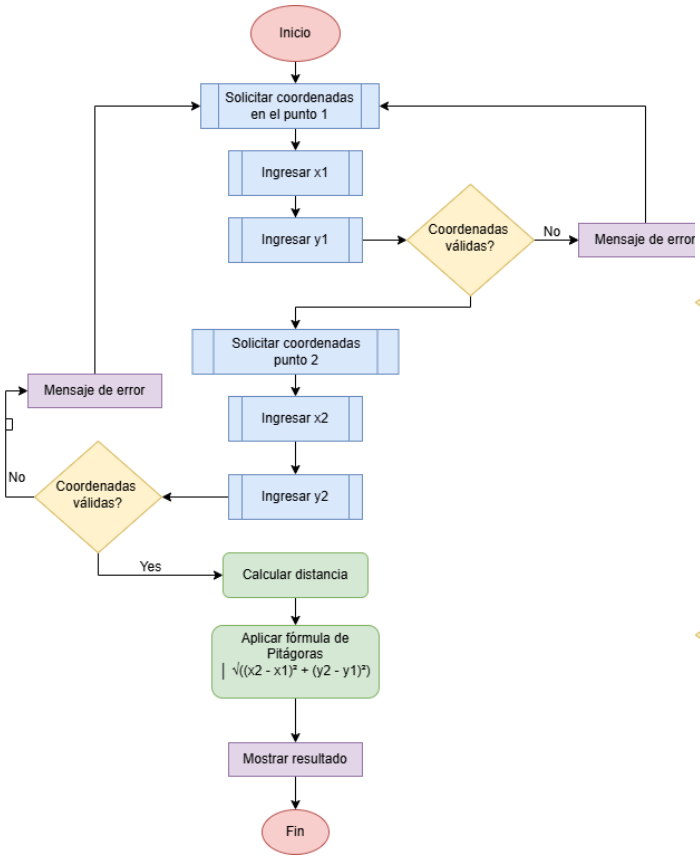


Figura 33: Diagrama de flujo para la función distancia

Fuente: Elaboración propia utilizando drawio, 2024

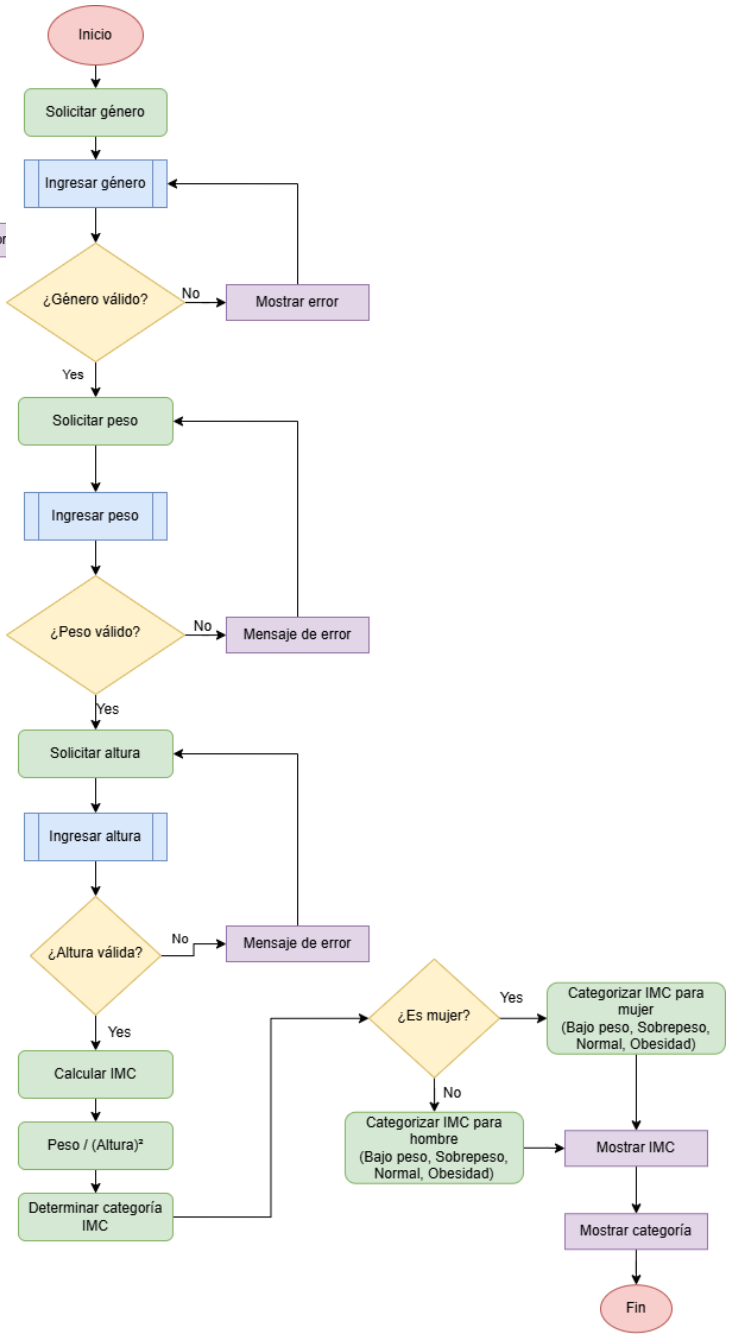


Figura 34: Diagrama de flujo para la función IMC

Fuente: Elaboración propia utilizando drawio, 2024

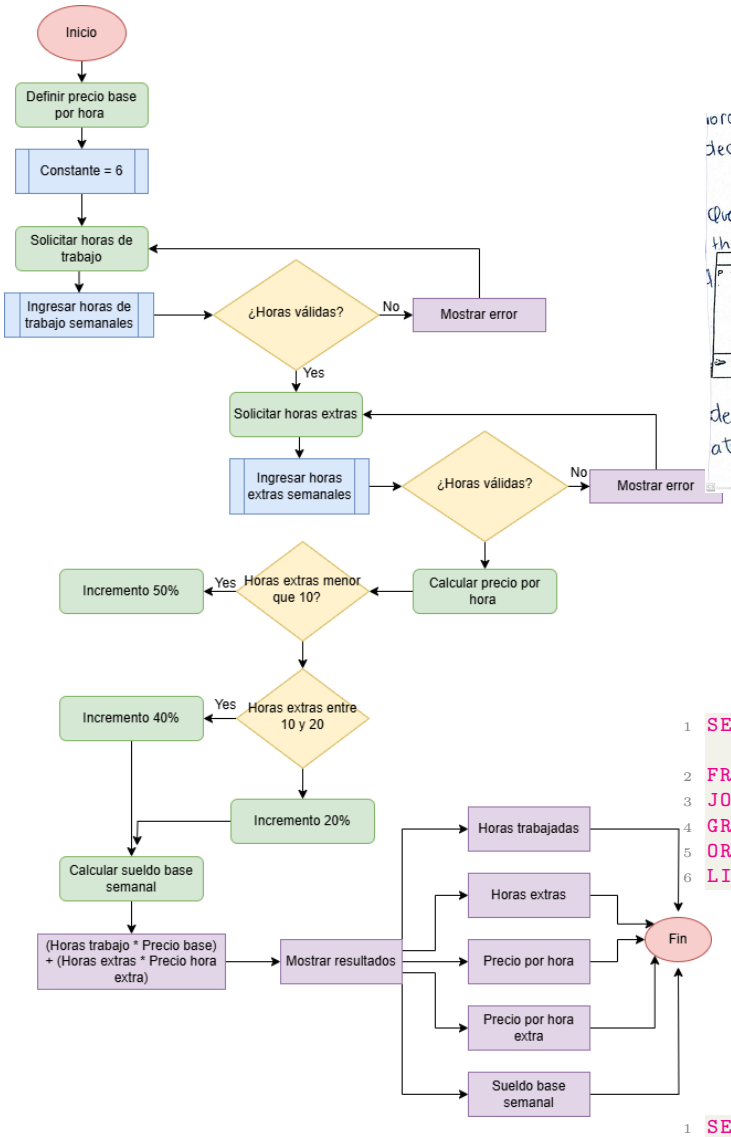


Figura 35: Diagrama de flujo para la función Nominas

Fuente: Elaboración propia utilizando drawio, 2024

VIII. QUERYS

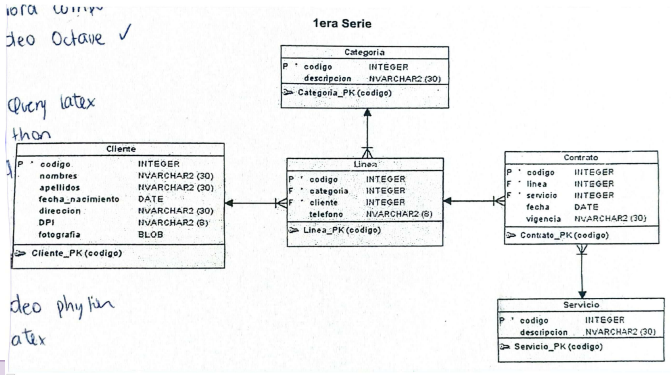


Figura 36: Tabla dada por el catedrático.

A. 1.¿Cuál es el servicio que más contratan los clientes?

```sql
SELECT s.descripcion, COUNT(c.servicio) AS
    total_contratos
FROM pcontrato c
JOIN pservicio s ON c.servicio = s.codigo
GROUP BY s.descripcion
ORDER BY total_contratos DESC
LIMIT 1;
```

Listing 3: servicio mas contratado

B. 2. Clientes con más servicios

```sql
SELECT cl.nombres, cl.apellidos, COUNT(c.
    servicio) AS total_servicios
FROM pcontrato c
JOIN pcliente cl ON c.linea = cl.codigo
GROUP BY cl.nombres, cl.apellidos
ORDER BY total_servicios DESC;
```

Listing 4: clientes con mas contratado

C. 3. ¿Qué categorías existen?

```sql
SELECT codigo, descripcion
FROM pcategoria;
SELECT cat.descripcion, COUNT(*) AS
    total_relacionados
FROM pcategoria cat
JOIN plinea l ON cat.codigo = l.categoria --
    Ajusta la tabla 'Linea' si aplica
GROUP BY cat.descripcion
ORDER BY total_relacionados DESC;
```

Listing 5: categorias existentes

D. 4.¿Cuántas líneas hay?

```
SELECT COUNT(*) AS total_lineas
FROM plinea;
SELECT codigo, categoria, cliente, telefono
FROM plinea;
```

Listing 6: categorias existentes

E. 4.¿Cantidad total de contratos?

```
SELECT COUNT(*) AS total_contratos
FROM pcontrato;
--SELECT *
---FROM pcontrato;
```

Listing 7: categorias existentes

IX. LINK DE GITHUB

Se ha subido un documento a GitHub en el repositorio privado Lenguajes aplicados IE. Presionar aqui o bien seguir el siguiente link https://github.com/CINDYGAT/Proyectos-aplicados-a-IE/tree/7c9d08fab40e3d1adedfd10ed0acdcdede5cbfe0/PrimerParcial

X. LINK DE DRIVE

Se ha subido un video mostrando el funcionamiento. Presionar aqui o bien seguir el siguiente link https://drive.google.com/drive/folders/1NyaIZKpWNw1X7yyEbiod02s5qMY1zYts?usp=sharing