

*Universidad de San Carlos de Guatemala*

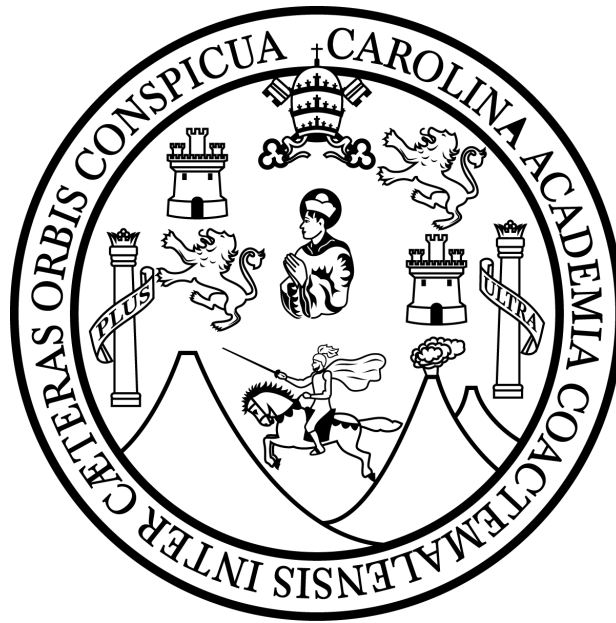
*Facultad de Ingeniería*

*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:*

Cindy Melissa Gatica Arriola

*Carné:*

201709692

# Primer Parcial

## Proyectos de Computación Aplicados a la Ingeniería Electrónica.\*

Cindy Melissa Gatica Arriola, 201709692<sup>1</sup>,\*\*

<sup>1</sup>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 opción 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 opción 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 Índice de Masa Corporal IMC de un usuario al ingresar su peso y altura.
- \* Implementar un sistema Login, en el cual, el usuario tiene únicamente tres intentos, los cuales son mostrados en pantalla.

### II. CÓDIGO EN OCTAVE

El código implementado en Octave es el siguiente

```
1 clc;
2 clear;
3 pkg load database;
4
5 %Variable para ir almacenando los resultados
6 global resultados conn;
7 resultados = struct();
8
9 try
```

```
10 % Conectar a la base de datos
11 conn = pq_connect(setdbopts('dbname', '
    primerParcial', 'host', 'localhost', 'port',
    '5432', 'user', 'postgres', 'password', '
    cindy123451'));
12 disp('Conexi n a la base de datos
    establecida.');
```

```
13 catch ME
14     error('No se pudo conectar a la base de
    datos: %s', ME.message);
15 end
16
17 function menu()
18     opcion = 0;
19     while opcion ~= 10
20         fprintf('\n\t\t Bienvenido al Men
    Principal!\n');
21         fprintf('1. Ingresar nombre usuario\n');
22         fprintf('2. Calcular factorial\n');
23         fprintf('3. Sistema login\n');
24         fprintf('4. Calcular distancia entre dos
    puntos\n');
25         fprintf('5. Calcular IMC\n');
26         fprintf('6. Sueldo base semanal\n');
27         fprintf('7. Visualizar archivo de texto\n');
28         fprintf('8. Historial de datos\n');
29         fprintf('9. Borrar datos\n');
30         fprintf('10. Salir\n');
31
32     try
33         opcion = input('Seleccione una opci n: ');
34         if isempty(opcion) || ~isnumeric(opcion) ||
            opcion < 1 || opcion > 10
35             error('Entrada inv lida. Por favor,
            ingrese un n mero entre 1 y 10.');
```

```
36         endif
37     catch ME
38         fprintf('Error: %s\n', ME.message);
39         continue;
40     end
41
42     switch opcion
43     case 1 %ingreso de nombre usuario
44         ingresoDatos();
45
46     case 2 %calcular factorial
47         calculo_factorial();
48
49     case 3 %Sistema login
50         sistemaLogin();
51
52     case 4 %calcular distancia entre dos
        puntos
53         calcular_distancia();
```

\* Proyectos de Computación Aplicados a la Ingeniería Electrónica.  
\*\* e-mail: 2787947930101@ingenieria.usac.edu.gt

```

53         case 5 %calcular imc
54             calcular_imc();
55         case 6 %sueldo base semanal
56             calcular_nomina();
57         case 7 %Visualizar datos archivo de
58             texto
59             registrar_operacion();
60         case 8 %Historial de datos
61             mostrar_historial();
62         case 9 %borrar datos
63             borrar_datos();
64             #mostrar_resultados();
65         case 10 %salir
66             fprintf(' Gracias por
67             visitarnos! Vuelva pronto.\n');
68             otherwise
69                 fprintf('Opci n no v lida.
70                 Intente nuevamente.\n');
71             end
72         end
73     end
74 end
75
76 function ingresoDatos()
77     global resultados;
78     try
79         resultados.nombre = input('Ingrese el
80         nombre del usuario: ', 's');
81         if isempty(resultados.nombre)
82             error('El nombre no puede estar
83             vac o.');
```

```

107         elseif(resultados.x >1)
108             resultados.factorial = 1;
109
110         for (i = 2:resultados.x)
111             resultados.factorial = resultados.
112             factorial * i;
113         end
114     end
115
116     % Mostrar el resultado
117     fprintf('El factorial de %d es %d\n',
118     resultados.x, resultados.factorial);
119
120 catch e
121     % Manejar el error si el usuario ingresa
122     un n mero negativo o no entero
123     fprintf('Ocurri un error. Error: %s\n',
124     e.message);
125 end
126
127 function sistemaLogin()
128     global resultados;
129     % Inicializar el n mero de intentos
130     resultados.intentos = 3;
131
132     while resultados.intentos > 0
133         resultados.tipo_programa2 = 'Login';
134         % Solicitar el nombre de usuario y la
135         contrase a
136         resultados.usuario = input('Ingrese el
137         nombre de usuario: ', 's');
138         resultados.contrasena = input('Ingrese la
139         contrase a: ', 's');
```

```

140         % Verificar las credenciales
141         if strcmp(resultados.usuario, 'admin') &&
142         strcmp(resultados.contrasena, '123')
143             disp(' Inicio de sesi n exitoso!');
144             break; % Salir del loop principal
145         else
146             % Mostrar el n mero de intentos
147             restantes
148             resultados.intentos = resultados.
149             intentos - 1;
150             fprintf('Credenciales incorrectas. Te
151             quedan %d intentos.\n', resultados.intentos)
152             ;
153
154             % Cerrar el programa si se agotan los
155             intentos
156             if resultados.intentos == 0
157                 disp('Has agotado todos los
158                 intentos. Cerrando el programa.');
```

```

159             return;
160         end
161     end
162 end
163
164 function calcular_distancia()
165     global resultados;
166     try
167         % Solicitar las coordenadas del primer punto
168         resultados.tipo_programa3 = 'Distancia';
169         fprintf('Ingrese las coordenadas del primer
170         punto (x1, y1):\n');
171         resultados.x1 = input('x1 = ');
172         resultados.y1 = input('y1 = ');

```

```

162 % Validar si las entradas son num ricas
163 if ~isnumeric(resultados.x1) || ~isnumeric(
164 resultados.y1) || isempty(resultados.x1) ||
165 isempty(resultados.y1)
166     error('Las coordenadas deben ser valores
167 num ricos.');
```

```

168 end
169 % Solicitar las coordenadas del segundo
170 punto
171 fprintf('\nIngrese las coordenadas del
172 segundo punto (x2, y2):\n');
173 resultados.x2 = input('x2 = ');
174 resultados.y2 = input('y2 = ');
175 % Validar si las entradas son num ricas
176 if ~isnumeric(resultados.x2) || ~isnumeric(
177 resultados.y2) || isempty(resultados.x2) ||
178 isempty(resultados.y2)
179     error('Las coordenadas deben ser valores
180 num ricos.');
```

```

181 end
182 % Calcular la distancia usando la fórmula
183 de Pitágoras
184 resultados.distancia = sqrt((resultados.x2 -
185 resultados.x1)^2 + (resultados.y2 -
186 resultados.y1)^2);
187 % Mostrar el resultado
188 fprintf('\nLa distancia más corta entre los
189 puntos (%d, %d) y (%d, %d) es: %.4f\n', ...
190 resultados.x1, resultados.y1,
191 resultados.x2, resultados.y2, resultados.
192 distancia);
193 catch ME
194     % Manejo de errores
195     fprintf('Error: %s\n', ME.message);
196 end_try_catch
197 end
198
199 function calcular_imc()
200 global resultados;
201 try
202     resultados.tipo_programa4 = 'IMC';
203     % Solicitar al usuario su sexo (hombre/mujer)
204     resultados.genero = input('Ingrese su genero
205 ("hombre" o "mujer"): ', 's');
206     resultados.genero = lower(strtrim(resultados.
207 genero)); % Convertir a minúsculas y
208 quitar espacios
209 % Validar el ingreso del sexo
210 if ~strcmp(resultados.genero, 'hombre') && ~
211 strcmp(resultados.genero, 'mujer')
212     error('Debe ingresar "hombre" o "mujer".');
213 ;
214 end
215 % Solicitar el peso
216 resultados.peso = input('Ingrese su peso en
217 kilogramos (kg): ');
218 if isempty(resultados.peso) || ~isnumeric(
219 resultados.peso) || resultados.peso <= 0
220     error('El peso debe ser un valor num rico
221 positivo.');
```

```

222 end
223
224 % Solicitar la altura
225 resultados.altura = input('Ingrese su altura
226 en metros (m): ');
227 if isempty(resultados.altura) || ~isnumeric(
228 resultados.altura) || resultados.altura <= 0
229     error('La altura debe ser un valor
230 num rico positivo.');
```

```

231 end
232 % Calcular el IMC
233 resultados.imc = resultados.peso / (
234 resultados.altura^2);
235 % Mostrar el resultado del IMC
236 fprintf('\nSu índice de Masa Corporal (IMC)
237 es: %.2f\n', resultados.imc);
238 % Determinar la conducta a seguir según el
239 IMC y sexo
240 if strcmp(resultados.genero, 'mujer')
241     if resultados.imc < 18.5
242         fprintf('Conducta a seguir: Bajo peso,
243 consulte a su médico.\n');
244     elseif resultados.imc >= 18.5 &&
245 resultados.imc < 24.9
246         fprintf('Conducta a seguir: Peso normal,
247 continúe con buenos hábitos.\n');
248     elseif resultados.imc >= 25 && resultados.
249 imc < 29.9
250         fprintf('Conducta a seguir: Sobrepeso,
251 considere una rutina de ejercicios.\n');
252     else
253         fprintf('Conducta a seguir: Obesidad,
254 consulte a su médico para un plan adecuado
255 .\n');
256     end
257 elseif strcmp(resultados.genero, 'hombre')
258     if resultados.imc < 18.5
259         fprintf('Conducta a seguir: Bajo peso,
260 consulte a su médico.\n');
261     elseif resultados.imc >= 18.5 &&
262 resultados.imc < 24.9
263         fprintf('Conducta a seguir: Peso normal,
264 continúe con buenos hábitos.\n');
265     elseif resultados.imc >= 25 && resultados.
266 imc < 29.9
267         fprintf('Conducta a seguir: Sobrepeso,
268 considere una rutina de ejercicios.\n');
269     else
270         fprintf('Conducta a seguir: Obesidad,
271 consulte a su médico para un plan adecuado
272 .\n');
273     end
274 end
275 catch ME
276     % Manejo de errores
277     fprintf('Error: %s\n', ME.message);
278 end_try_catch
279 end
280
281 function calcular_nomina()
282 global resultados;
283 % Constante para el precio por hora
284 PRECIO_HORA = 6;
285 try
286     % Solicitar horas de trabajo

```

```

259 resultados.tipo_programa5 = 'Sueldo semanal'
260 ;
261 resultados.horas_trabajo = input('Ingrese
las horas de trabajo semanales: ');
262 if isempty(resultados.horas_trabajo) || ~
isnumeric(resultados.horas_trabajo) ||
resultados.horas_trabajo < 0
263     error('Las horas de trabajo deben ser un
valor num rico positivo.');
```

```

264 end
265 % Solicitar horas extras trabajadas
266 resultados.horas_extras = input('Ingrese las
horas extra semanales: ');
267 if isempty(resultados.horas_extras) || ~
isnumeric(resultados.horas_extras) ||
resultados.horas_extras < 0
268     error('Las horas extra deben ser un valor
num rico positivo.');
```

```

269 end
270 % Calcular precio por hora extra
271 if resultados.horas_extras < 10
272     resultados.precio_hora_extra = PRECIO_HORA
* 1.5; % 50% mayor
273 else if resultados.horas_extras >= 10 &&
resultados.horas_extras <= 20
274     resultados.precio_hora_extra = PRECIO_HORA
* 1.4; % 40% mayor
275 else
276     resultados.precio_hora_extra = PRECIO_HORA
* 1.2; % 20% mayor
277 end
278 % Calcular el sueldo base semanal
279 resultados.sueldo_base = (resultados.
horas_trabajo * PRECIO_HORA) + (resultados.
horas_extras * resultados.precio_hora_extra)
280 ;
281 % Mostrar los resultados
282 fprintf('\nResumen de n mina:\n');
283 fprintf('Horas trabajadas: %d\n', resultados
.horas_trabajo);
284 fprintf('Horas extra: %d\n', resultados.
horas_extras);
285 fprintf('Precio por hora: %.2f\n',
PRECIO_HORA);
286 fprintf('Precio por hora extra: %.2f\n',
resultados.precio_hora_extra);
287 fprintf('Sueldo base semanal: %.2f\n',
resultados.sueldo_base);
288
289 catch ME
290     % Manejo de errores
291     fprintf('Error: %s\n', ME.message);
292 end_try_catch
293 end
294
295 function registrar_operacion()
296     #global resultados;
297     global resultados conn;
298     try
299         % Generar el texto para el archivo
300         fecha_actual = datestr(now, 'yyyy-mm-dd
HH:MM:SS');
301         registro_txt = sprintf(['...
302         ',
303         'Fecha: %s\n', ...
304         'Usuario: %s\n', ...
305         'Tipo de programa: %s\n', ...
306         'Numero a calcular factorial: %s\n',
307         ...
308         'Resultado de factorial: %s\n', ...
309         '
310         -----\n
311         n', ...
312         'Tipo de programa: %s\n', ...
313         'Intentos disponibles: %s\n', ...
314         'Usuario: %s\n', ...
315         'Password: %s\n', ...
316         '
317         -----\n
318         n', ...
319         'Tipo de programa: %s\n', ...
320         'x1: %s\n', ...
321         'y1: %s\n', ...
322         'x2: %s\n', ...
323         'y2: %s\n', ...
324         'distancia entre coordenadas: %s\n',
325         ...
326         '
327         -----\n
328         n', ...
329         'Tipo de programa: %s\n', ...
330         'Genero: %s\n', ...
331         'Peso: %s\n', ...
332         'Altura: %s\n', ...
333         'IMC: %s\n', ...
334         '
335         -----\n
336         n', ...
337         'Tipo de programa: %s\n', ...
338         'Horas trabajadas: %s\n', ...
339         'Horas extras trabajadas: %s\n', ...
340         'Precio horas extras trabajadas: %s\n',
341         ...
342         'Sueldo por semana: %s\n', ...
343         '
344         -----\n
345         n'], ...
346         fecha_actual, resultados.nombre, ...
347         resultados.tipo_programa1, num2str(
348         resultados.x), num2str(resultados.factorial)
349         , ...
350         resultados.tipo_programa2, num2str(
351         resultados.intentos), resultados.usuario,
352         resultados.contrasena, ...
353         resultados.tipo_programa3, num2str(
354         resultados.x1), num2str(resultados.y1),
355         num2str(resultados.x2), num2str(resultados.
356         y2), num2str(resultados.distancia), ...
357         resultados.tipo_programa4,
358         num2str(resultados.peso),
359         num2str(resultados.altura), num2str(
360         resultados.imc), ...
361         resultados.tipo_programa5, num2str(
362         resultados.horas_trabajo), num2str(
363         resultados.horas_extras), num2str(resultados.
364         precio_hora_extra), num2str(resultados.
365         sueldo_base));
366
367         fprintf('Archivo salida.txt generado:\n'
368         );
369         fprintf('%s\n', registro_txt);
370
371         % Guardar en el archivo txt

```



```

435     historial_distancia = pq_exec_params(
436         conn, "SELECT * FROM distancia;");
437     disp('
-----
');
438     disp('Historial de operaciones desde la
base de datos:');
439     disp(historial_distancia);
440     disp('
-----
');
441     catch ME
442         fprintf('Error al obtener el historial
distancia: %s\n', ME.message);
443     end
444 % Obtener el historial desde la base de datos -
imc
445 try
446     historial_imc = pq_exec_params(conn, "
SELECT * FROM imc;");
447     disp('
-----
');
448     disp('Historial de operaciones desde la
base de datos:');
449     disp(historial_imc);
450     disp('
-----
');
451     catch ME
452         fprintf('Error al obtener el historial
imc: %s\n', ME.message);
453     end
454 % Obtener el historial desde la base de datos -
nominas
455 try
456     historial_nominas = pq_exec_params(conn,
"SELECT * FROM nominas;");
457     disp('
-----
');
458     disp('Historial de operaciones desde la
base de datos:');
459     disp(historial_nominas);
460     disp('
-----
');
461     catch ME
462         fprintf('Error al obtener el historial
nominas: %s\n', ME.message);
463     end
464 end
465
466 function borrar_datos()
467     global conn;
468     try
469         pq_exec_params(conn, "delete from
factorial;");
470         pq_exec_params(conn, "delete from login;
");
471         pq_exec_params(conn, "delete from
distancia;");
472         pq_exec_params(conn, "delete from imc;");
473         ;
474         pq_exec_params(conn, "delete from
nominas;");
475         fprintf('Todos los registros de la base
de datos han sido eliminados.\n');
476     catch ME
477         fprintf('Error al borrar los datos de la
base de datos: %s\n', ME.message);
478     end
479 end
480
481 function mostrar_resultados()
482     global resultados;
483     disp('Valores actuales de resultados:');
484     disp(resultados);
485 end
486
487 menu();
488 pq_close(conn);
489
490

```

Listing 1: Programa primer parcial

### III. CÓDIGO IMPLEMENTADO EN PYTHON

```

1 import psycopg2
2 from psycopg2 import sql
3 import math
4 import datetime
5 import getpass
6
7 # Variables globales para almacenar los datos
del cliente
8 resultados = {}
9
10 # Men principal
11 def menu():
12     opcion = 0
13     while opcion != 10:
14         print("\n\t\t Bienvenido al Men
Principal!")
15         print("1. Ingresar nombre usuario")
16         print("2. Calcular factorial")
17         print("3. Sistema login")
18         print("4. Calcular distancia entre dos
puntos")
19         print("5. Calcular IMC")
20         print("6. Sueldo base semanal")
21         print("7. Visualizar archivo de texto")
22         print("8. Historial de datos")
23         print("9. Borrar datos")
24         print("10. Salir")
25         try:
26             opcion = int(input("Seleccione una
opción: "))
27             if opcion == 1:
28                 ingreso_datos()
29             elif opcion == 2:
30                 calcular_factorial()
31             elif opcion == 3:
32                 sistema_login()
33             elif opcion == 4:
34                 calcular_distancia()
35             elif opcion == 5:
36                 calcular_imc()
37             elif opcion == 6:
38                 calcular_nomina()
39             elif opcion == 7:
40                 visualizar_archivo()
41             elif opcion == 8:
42                 #imprimir_resultados()

```



```

43     mostrar_historial()
44     elif opcion == 9:
45         borrar_datos()
46     elif opcion == 10:
47         print(" Gracias por visitarnos!
48         Vuelva pronto.")
49     else:
50         print("Opci n no v lida.
51         Intente nuevamente.")
52     except ValueError:
53         print("Entrada inv lida. Por favor,
54         ingrese un n mero entre 1 y 9.")
55
56 # Funci n para ingresar nombre
57 def ingreso_datos():
58     global resultados
59     try:
60         nombre = input("Ingrese el nombre del
61         usuario: ").strip()
62         if not nombre:
63             raise ValueError("El nombre no puede
64             estar vac o.")
65         print(f"Datos ingresados correctamente:
66         {nombre}")
67
68         resultados["nombre"] = nombre #guarde
69         nombre en variable global resultados
70         #print(f"Nombre guardado: {resultados['
71         nombre'}]")
72     except Exception as e:
73         print(f"Error: {e}")
74
75 # Calcular factorial
76 def calcular_factorial():
77     global resultados
78     try:
79         while True:
80             x = int(input("Ingresa un n mero
81             entero positivo: "))
82             resultados["x"] = x
83
84             if x < 0:
85                 print("Error: El n mero debe
86                 ser positivo.")
87             else:
88                 break
89
90             factorial = math.factorial(x)
91             print(f"El factorial de {x} es {
92             factorial}")
93             #almacenar datos en la variable global
94             resultados["factorial"] = factorial
95     except ValueError:
96         print("Error: Debe ingresar un n mero
97         entero.")
98
99 # Sistema login
100 def sistema_login():
101     global resultados
102     intentos = 3
103     try:
104         while intentos > 0:
105             usuario = input("Ingrese el nombre
106             de usuario: ")
107             resultados["usuario"] = usuario
108             contrasena = getpass.getpass("
109             Introduce tu contrase a: ")
110             resultados["Contrasena"] =
111             contrasena
112             if usuario == "admin" and contrasena
113             == "123":

```

```

114         print(" Inicio de sesi n
115         exitoso!")
116     return
117     else:
118         intentos -= 1
119         print(f"Credenciales incorrectas
120         . Intentos restantes: {intentos}")
121         if intentos == 0:
122             print("Has agotado todos los
123             intentos. Cerrando el programa.")
124             return
125         resultados["intentos"] = intentos
126     except Exception as e:
127         print(f"Error: {e}")
128
129 # Calcular distancia
130 def calcular_distancia():
131     global resultados
132     try:
133         x1 = float(input("x1 = "))
134         y1 = float(input("y1 = "))
135         x2 = float(input("x2 = "))
136         y2 = float(input("y2 = "))
137         resultados["x1"] = x1
138         resultados["y1"] = y1
139         resultados["x2"] = x2
140         resultados["y2"] = y2
141         distancia = math.sqrt((x2 - x1)**2 + (y2
142         - y1)**2)
143         resultados["distancia"] = distancia
144         print(f"La distancia entre ({x1}, {y1})
145         y ({x2}, {y2}) es: {distancia:.4f}")
146     except ValueError:
147         print("Error: Las coordenadas deben ser
148         valores num ricos.")
149
150 # Calcular IMC
151 def calcular_imc():
152     global resultados
153     try:
154         genero = input("Ingrese su g nero ('
155         hombre' o 'mujer'): ").strip().lower()
156         resultados["genero"] = genero
157         if genero not in ["hombre", "mujer"]:
158             raise ValueError("Debe ingresar '
159             hombre' o 'mujer'.")
160         peso = float(input("Ingrese su peso (kg)
161         : "))
162         altura = float(input("Ingrese su altura
163         (m): "))
164         resultados["peso"] = peso
165         resultados["altura"] = altura
166         if peso <= 0 or altura <= 0:
167             raise ValueError("Peso y altura
168             deben ser positivos.")
169         imc = peso / (altura ** 2)
170         resultados["imc"] = imc
171         print(f"S u IMC es: {imc:.2f}")
172         if imc < 18.5:
173             print("Bajo peso, consulte a su
174             m dico.")
175         elif imc < 24.9:
176             print("Peso normal, contin e con
177             buenos h bits.")
178         elif imc < 29.9:
179             print("Sobrepeso, considere una
180             rutina de ejercicios.")
181         else:
182             print("Obesidad, consulte a su
183             m dico para un plan adecuado.")

```



```

152     except ValueError as e:
153         print(f"Error: {e}")
154
155 # Calcular n mina
156 def calcular_nomina():
157     global resultados
158     PRECIO_HORA = 6
159     try:
160         horas_trabajo = int(input("Ingrese horas
161         trabajadas: "))
162         horas_extras = int(input("Ingrese horas
163         extras: "))
164         resultados["horas_trabajo"] =
165         horas_trabajo
166         resultados["horas_extras"] =
167         horas_extras
168         if horas_extras < 10:
169             precio_extra = PRECIO_HORA * 1.5
170         elif horas_extras <= 20:
171             precio_extra = PRECIO_HORA * 1.4
172         else:
173             precio_extra = PRECIO_HORA * 1.2
174         sueldo = (horas_trabajo * PRECIO_HORA) +
175         (horas_extras * precio_extra)
176         resultados["precio_extra"] =
177         precio_extra
178         resultados["sueldo"] = sueldo
179         print(f"Sueldo base semanal: {sueldo:.2f}
180         ")
181     except ValueError:
182         print("Error: Ingrese valores num ricos
183         .")
184
185 # Visualizar archivos txt
186 def visualizar_archivo():
187     global resultados
188     try:
189         # Acceder a los valores de resultados
190         correctamente
191         salida = (
192             "
193             -----\n
194             f"Nombre: {resultados.get('nombre',
195             'N/A')}\n"
196             f"Tipo de operacion: Funcion
197             factorial\n"
198             f"x: {resultados.get('x', 'N/A')}\n"
199             f"Factorial: {resultados.get('
200             factorial', 'N/A')}\n"
201             f"
202             -----\n
203             f"Tipo de operacion: Login\n"
204             f"Intentos: {resultados.get('
205             intentos', 'N/A')}\n"
206             f"Usuario: {resultados.get('usuario
207             ', 'N/A')}\n"
208             f"Contrase a: {resultados.get('
209             contrasena', 'N/A')}\n"
210             f"
211             -----\n
212             f"Tipo de operacion: Distancia\n"
213             f"x1: {resultados.get('x1', 0):.2f}\n
214             "
215             f"y1: {resultados.get('y1', 0):.2f}\n
216             "
217             f"x2: {resultados.get('x2', 0):.2f}\n
218             "
219             f"y2: {resultados.get('y2', 0):.2f}\n
220             "
221             f"Distancia: {resultados.get('
222             distancia', 0):.2f}\n"
223             f"
224             -----\n
225             f"Tipo de operacion: IMC\n"
226             f"G nero: {resultados.get('genero',
227             'N/A')}\n"
228             f"Peso: {resultados.get('peso', 0)
229             :.2f}\n"
230             f"Altura: {resultados.get('altura',
231             0):.2f}\n"
232             f"IMC: {resultados.get('imc', 0):.2f
233             }\n"
234             f"
235             -----\n
236             f"Tipo de operacion: Sueldo Semanal
237             \n"
238             f"Horas trabajadas: {resultados.get
239             ('horas_trabajo', 0):.2f}\n"
240             f"Horas extras: {resultados.get('
241             horas_extras', 0):.2f}\n"
242             f"Precio hora extra: {resultados.get
243             ('precio_extra', 0):.2f}\n"
244             f"Sueldo semanal: {resultados.get('
245             sueldo', 0):.2f}\n"
246             "
247             -----\n
248             "
249             )
250         print("Archivo txt generado:")
251         print(salida)
252
253         with open("c:/Users/Melissa A/Documents/
254         Cursos 2do semestre 2024/proyectos IE/
255         PrimerParcial/salidap.txt", "a") as f:
256             f.write(salida)
257             print("Archivo guardado en 'salidap.txt
258             '.")
259     except Exception as e:
260         print(f"Error al generar la factura: {e}
261         ")
262
263 #Conectar a la base de datos
264 def basePosgresql():
265     try:
266         conn = psycopg2.connect(
267             dbname="primerParcial",
268             user="postgres",
269             password="cindy123451",
270             host="localhost",
271             port="5432"
272         )
273         return conn
274     except Exception as e:
275         print("Error al conectar a la base de
276         datos:", e)
277         return None
278
279 #Insertar datos en base
280 def insertar_en_base_datos():
281     global resultados
282     Nombre = resultados["nombre"]
283     Numero = resultados["x"]
284     Factorial = resultados["factorial"]

```

```

246 """Inserta datos en las tablas."""
247 try:
248     conn = basePosgresql()
249     if conn:
250         cursor = conn.cursor()
251         # Query para insertar datos
252         query = 'INSERT INTO factorial("
nombre", "numero", "factorial") VALUES (%s,
%s, %s)'
253         cursor.execute(query, (Nombre,
Numero, Factorial))
254         conn.commit()
255         conn.close()
256         print("Datos insertados en la base
de datos correctamente.")
257     except Exception as e:
258         print(f"Error al insertar en la base de
datos: {e}")
259
260 # Mostrar historial
261 def mostrar_historial():
262     try:
263         conn = basePosgresql()
264         if conn:
265             cursor = conn.cursor()
266             cursor.execute('SELECT * FROM
factorial;')
267             registros = cursor.fetchall()
268             print("\n--- Historial en la base de
datos ---")
269             for registro in registros:
270                 print(registro)
271             conn.close()
272     except Exception as e:
273         print(f"Error: {e}")
274
275 # Borrar datos
276 def borrar_datos():
277     try:
278         if os.path.exists("c:/Users/Melissa A/
Documents/Cursos 2do semestre 2024/proyectos
IE/PrimerParcial/salidap.txt"):
279             os.remove("c:/Users/Melissa A/
Documents/Cursos 2do semestre 2024/proyectos
IE/PrimerParcial/salidap.txt")
280             print("Archivo 'salidap.txt'
eliminado.")
281         else:
282             print("El archivo 'salidap.txt' no
existe.")
283
284         conn = basePosgresql()
285         if conn:
286             cursor = conn.cursor()
287             cursor.execute('DELETE FROM
factorial;')
288             conn.commit()
289             conn.close()
290             print("Datos eliminados de la base
de datos.")
291     except Exception as e:
292         print(f"Error al borrar datos: {e}")
293
294
295 #####
296 def imprimir_resultados():
297     global resultados
298     if not resultados:
299         print("No hay datos almacenados en
resultados.")

```

```

300         return
301         print("\nValores almacenados en resultados:"
)
302         for clave, valor in resultados.items():
303             print(f"{clave}: {valor}")
304 #####
305
306 # Ejecutar el men
307 if __name__ == "__main__":
308     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 = pg_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.');
```

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();
case 10 %salir
    #mostrar_resultados();
    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.');
```

Figura 2: 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.');
```

Figura 4: Editor de texto Octave

Fuente: Elaboración propia utilizando Octave, 2024

## B. Python

```

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.');
```

Figura 3: Editor de texto Octave

Fuente: Elaboración propia utilizando Octave, 2024

```

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\tBienvenido 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

```
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

```
# 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

```
# 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

```
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



```
def calcular_nomina():
    print(" 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

```
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

```
                ¡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

	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

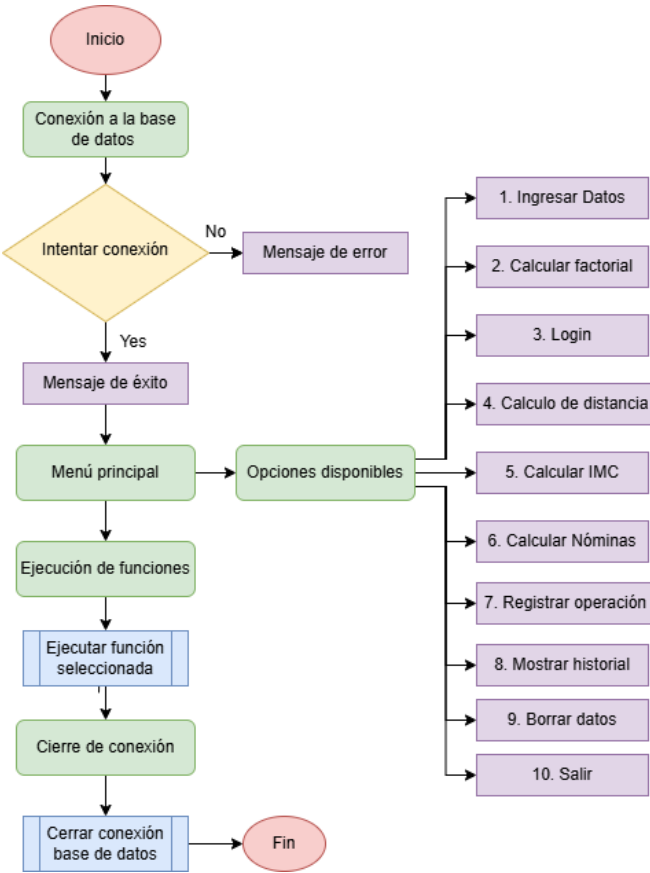


Figura 30: Diagrama de flujo para la logica general  
Fuente: Elaboración propia utilizando drawio, 2024

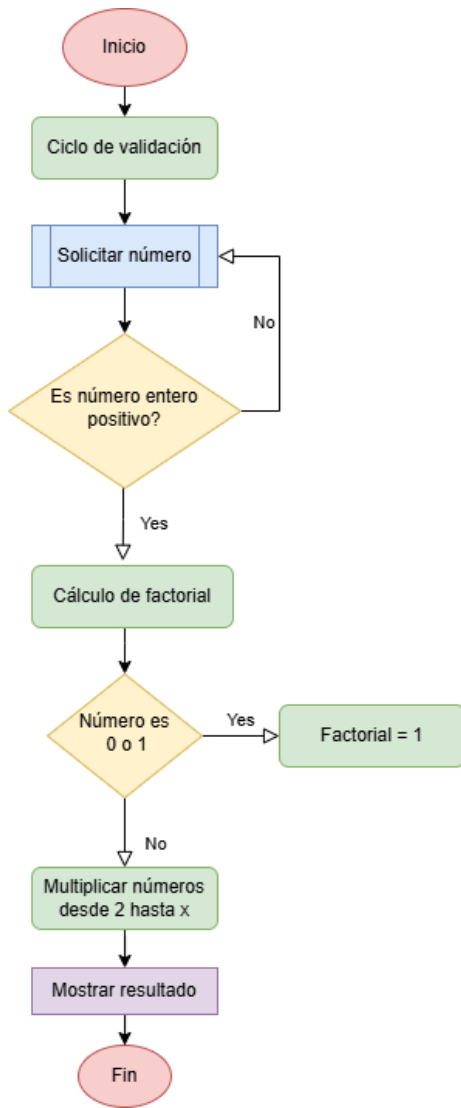


Figura 31: Diagrama de flujo para la función factorial  
Fuente: Elaboración propia utilizando drawio, 2024

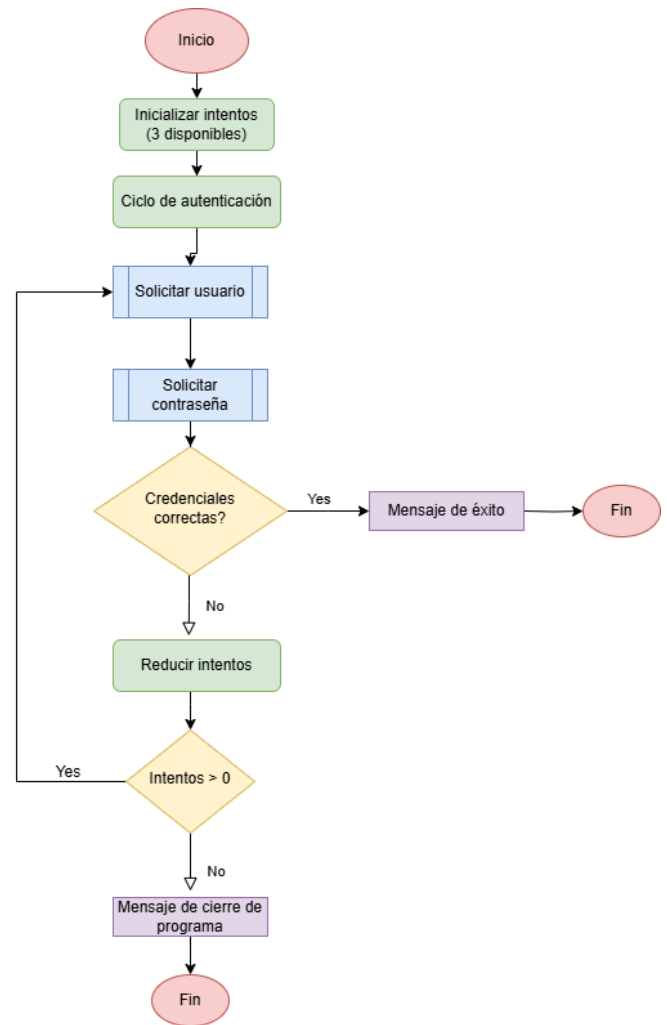


Figura 32: Diagrama de flujo para la función login  
Fuente: Elaboración propia utilizando drawio, 2024

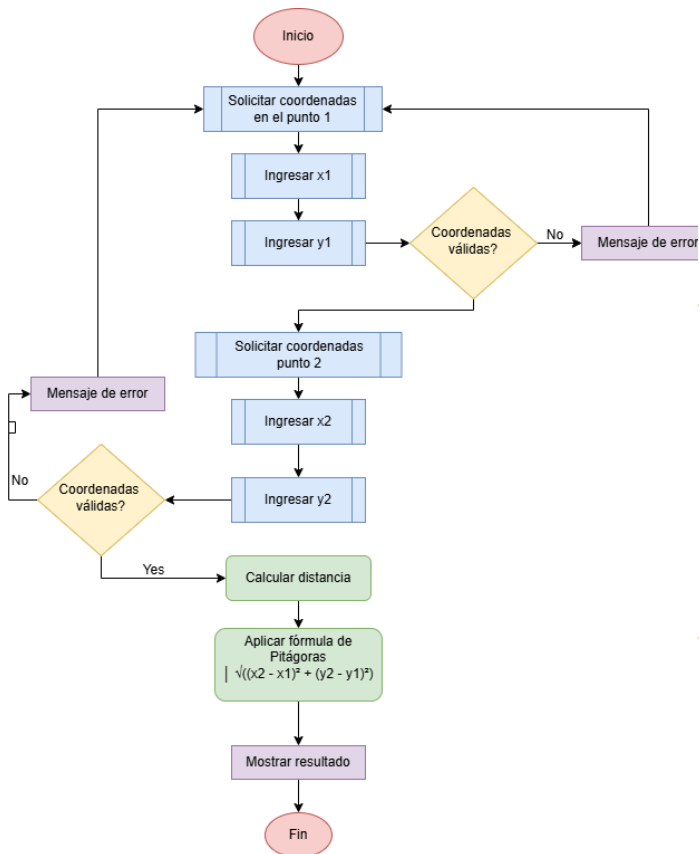


Figura 33: Diagrama de flujo para la función distancia

Fuente: Elaboración propia utilizando drawio, 2024

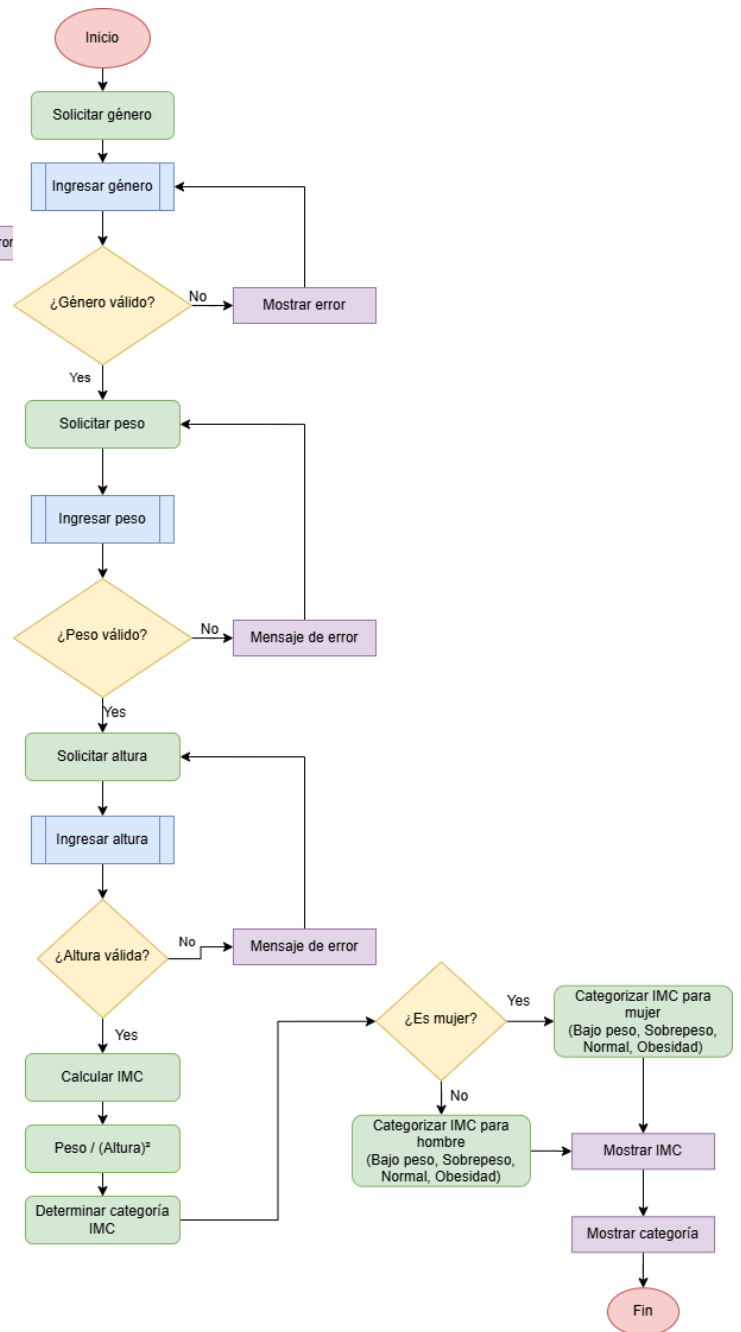


Figura 34: Diagrama de flujo para la función IMC

Fuente: Elaboración propia utilizando drawio, 2024

## VIII. QUERYS

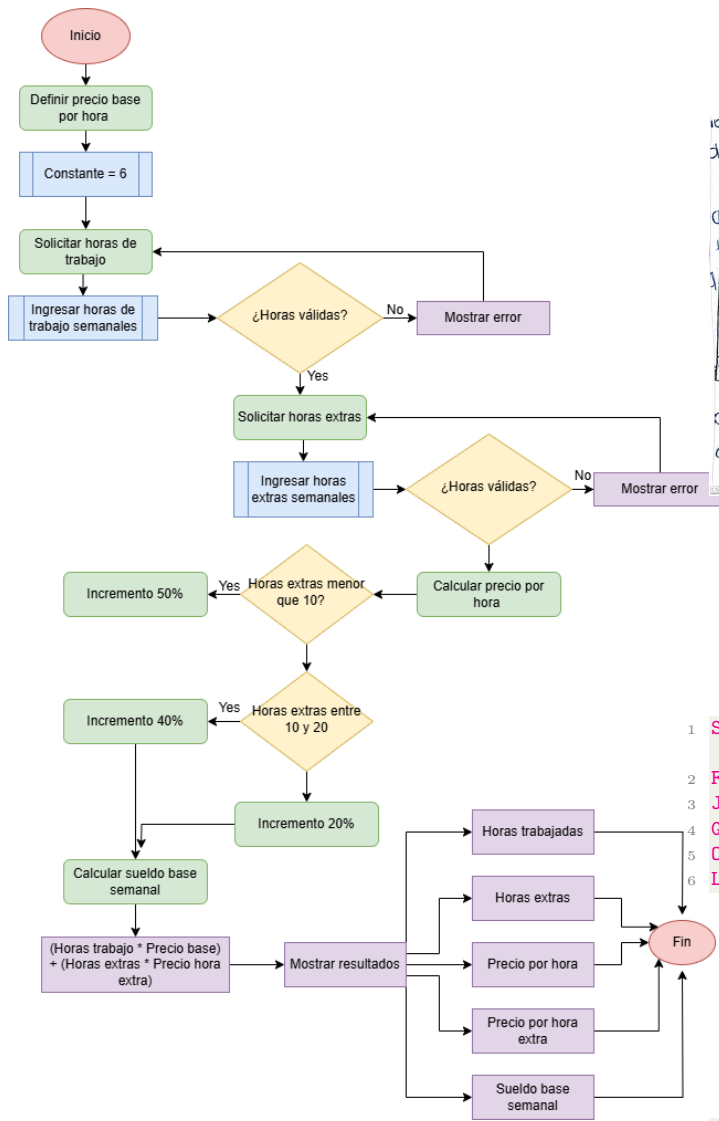


Figura 35: Diagrama de flujo para la función Nominas

Fuente: Elaboración propia utilizando drawio, 2024

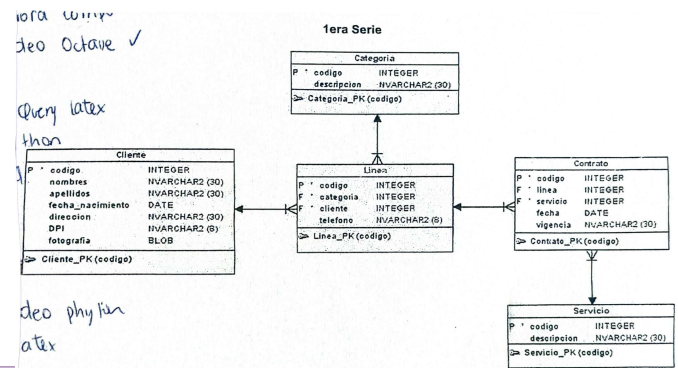


Figura 36: Tabla dada por el catedrático.

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

```

1 SELECT s.descripcion, COUNT(c.servicio) AS
   total_contratos
2 FROM pcontrato c
3 JOIN pservicio s ON c.servicio = s.codigo
4 GROUP BY s.descripcion
5 ORDER BY total_contratos DESC
6 LIMIT 1;

```

Listing 3: servicio mas contratado

## B. 2. Clientes con más servicios

```

1 SELECT cl.nombres, cl.apellidos, COUNT(c.
   servicio) AS total_servicios
2 FROM pcontrato c
3 JOIN pcliente cl ON c.linea = cl.codigo
4 GROUP BY cl.nombres, cl.apellidos
5 ORDER BY total_servicios DESC;

```

Listing 4: clientes con mas contratado

## C. 3. ¿Qué categorías existen?

```

1 SELECT codigo, descripcion
2 FROM pcategoria;
3 SELECT cat.descripcion, COUNT(*) AS
   total_relacionados
4 FROM pcategoria cat
5 JOIN plinea l ON cat.codigo = l.categoria --
   Ajusta la tabla 'Linea' si aplica
6 GROUP BY cat.descripcion
7 ORDER BY total_relacionados DESC;

```

Listing 5: categorias existentes

#### D. 4.¿Cuántas líneas hay?

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

Listing 6: categorias existentes

#### E. 4.¿Cantidad total de contratos?

```
1 SELECT COUNT(*) AS total_contratos
2 FROM pcontrato;
3 --SELECT *
4 ---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>

---