



Bases de Datos



Facultad de
Ingeniería

TRABAJO INTEGRADOR FINAL

Análisis de Hipertensión Arterial



Matias Rios
Licenciatura en Bioinformática



Contenido

- Introducción
- Objetivos
- Metodología
- Diseño de la Base de Datos
- Diagrama Entidad - Relación (DER)
- Consultas SQL
- Optimización de consultas
- Implementación de la Interfaz Gráfica BioLAB
- Resultados y conclusiones

INTRODUCCIÓN



Base de datos BIOLAB

PROPÓSITO

Evaluar y consolidar conocimientos adquiridos en bases de datos mediante un caso práctico

PROBLEMA ABORDADO

- La hipertensión arterial como enfermedad crónica.
- Necesidad de análisis integral de datos clínicos, bioquímicos y de estilo de vida.

METODOLOGIA

Desarrollo de una base de datos funcional con integración de una interfaz gráfica para la visualización y análisis de información.

OBJETIVOS

Base de datos BIOLAB

OBJETIVO GENERAL

- Crear una base de datos bioinformática para analizar factores de riesgo de hipertensión arterial.
- Facilitar el análisis de factores de riesgo y la toma de decisiones en salud pública.





Objetivos específicos

1) Diseñar un modelo relacional que incluya datos clínicos, bioquímicos y antropométricos.

2) Implementar la base de datos en PostgreSQL.

3) Crear consultas SQL avanzadas para extraer información valiosa.

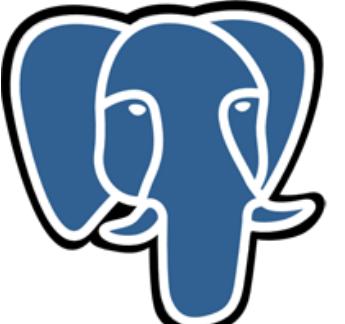
4) Optimizar el rendimiento de las consultas mediante árboles heurísticos.

5) Desarrollar una interfaz gráfica intuitiva para los usuarios finales.

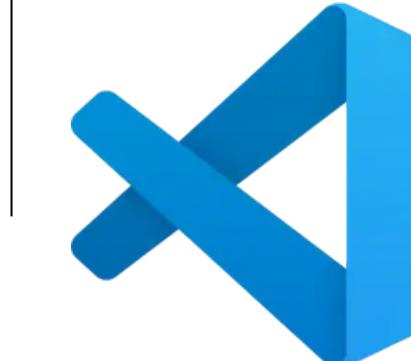
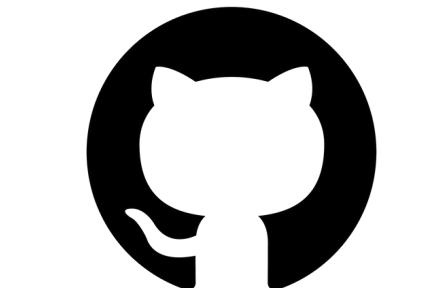
6) Comprobar funcionalidad de lo realizado

Herramientas utilizadas

kaggle

 PostgreSQL
 DBeaver



Metodología

1) Selección del Dataset:

- Fuente: Kaggle
 - Contenido: Datos demográficos, clínicos y bioquímicos de pacientes
-

2) Diseño DER

- Identificación de entidades y atributos y relaciones claves
-

3) Implementación

- Uso de PostgreSQL como sistema de gestión de bases de datos.
 - Carga de datos ficticios para ilustrar consultas.
-

4) Optimización heurística

- Aplicación de reglas heurísticas para mejorar el rendimiento.
-

5) Desarrollo de una interfaz gráfica

- Usando conocimientos previos adquiridos en Python, Matplotlib, QtDesigner, entre otros.

1 - Selección del Dataset



1 - Selección de Dataset

kaggle

Create

Home

Competitions

Datasets

Models

Code

Discussions

Learn

More

View Active Events

Search

AXEL FREDERICK FÉLIX JIMÉNEZ AND 1 COLLABORATOR · UPDATED 10 MONTHS AGO

47 New Notebook Download

Hypertension Arterial Mexico Data Set

Hypertension_Arterial_Mexico.csv



Data Card Code (4) Discussion (5) Suggestions (0)

About Dataset

Este conjunto de datos fue creado utilizando tres conjuntos de datos provenientes de la página de gobierno llamada Encuesta Nacional de Salud y Nutrición (ENSANUT)<https://ensanut.insp.mx/encuestas/ensanutcontinua2022/descargas.php>. Los conjuntos de datos utilizados son los siguientes:

- [ensaaント2022_entrega_w.csv](#) (Cuestionario de antropometría y tensión arterial)
- [Determinaciones_bioquímicas_cronicas_deficiencias_9feb23.csv](#) (Determinaciones para enfermedades crónicas y deficiencias)
- [ensafisica2022_adultos_entrega_w.csv](#) (Actividad física - Adolescentes y adultos)

Este conjunto de datos incluye información biométrica y medidas asociadas con la salud de pacientes en México, proporcionando un análisis detallado de variables como sexo, edad, concentración de hemoglobina, temperatura ambiente, valores de ácido úrico, albúmina, colesterol (HDL, LDL, total), creatinina, resultado de glucosa, insulina, proteína C reactiva, triglicéridos, glucosa promedio, hemoglobina glucosilada, ferritina, folato, homocisteína, transferrina, vitamina B12, vitamina D, peso, estatura, medida de cintura, entre otros.

Usability 10.00

License Attribution 4.0 International (CC ...)

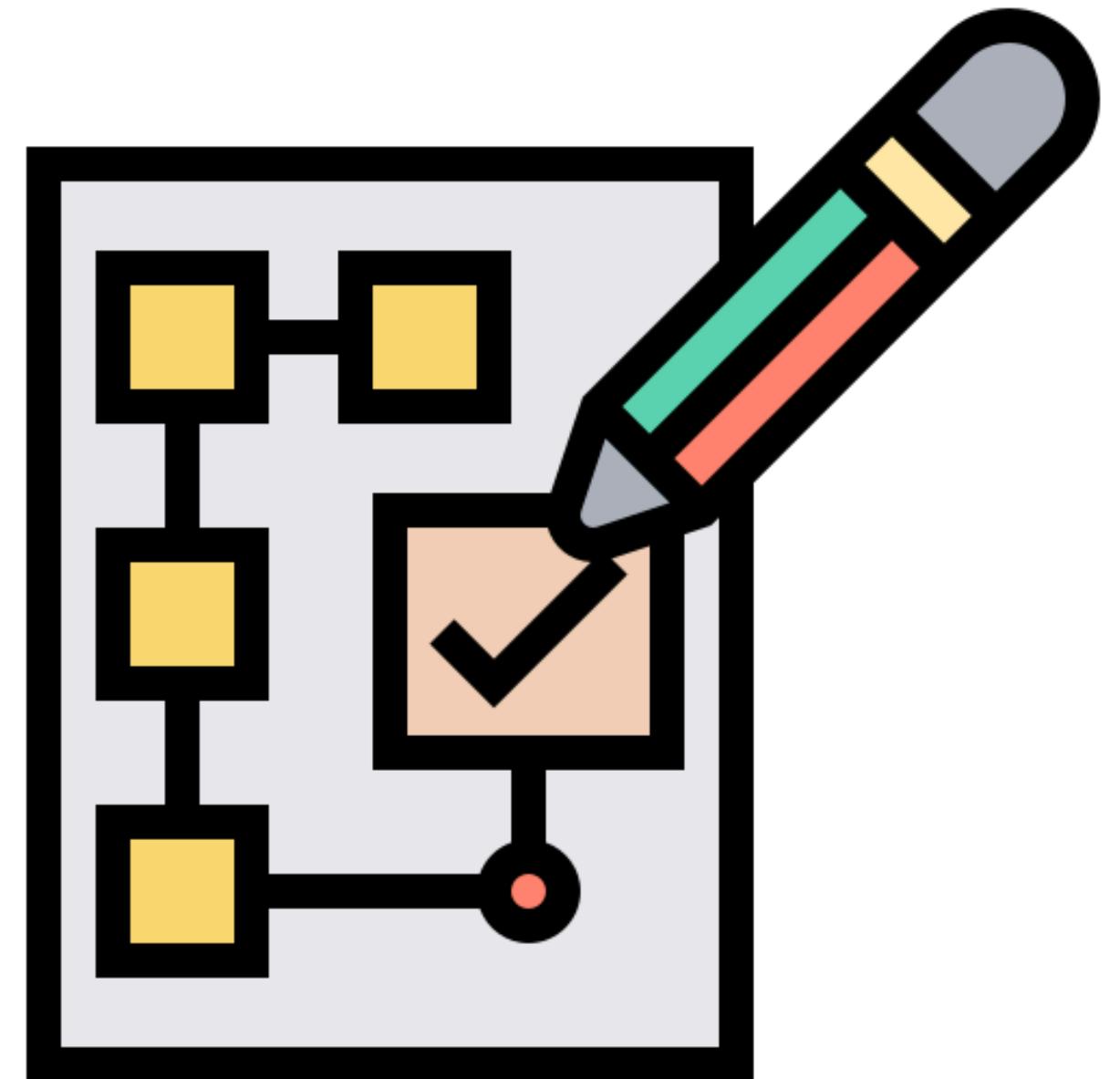
Expected update frequency Annually

Tags

Health Health Conditions
Government Healthcare

kaggle

2 - Diseño del DER



Entidades

- 👉 Paciente
- 👉 Examen_clinico
- 👉 Prueba_sangre
- 👉 Perfil_lipidico
- 👉 Marcadores_nutricionales
- 👉 Marcadores_inflamatorios
- 👉 Mediciones_fisicas

2 - Diseño del DER

- Identificación de entidades y relaciones

Relaciones

- Paciente ↔ Examen_clinico (1:N)
- Examen_clinico ↔ Prueba_sangre (1:N)
- Examen_clinico ↔ Perfil_lipidico (1:1)
- Examen_clinico ↔ Marcadores_nutricionales (1:1)
- Examen_clinico ↔ Marcadores_inflamatorios (1:1)
- Paciente ↔ Mediciones_fisicas (1:1)

Relaciones en detalle...



Paciente ↔ Examen_clinico:

- Un paciente puede tener varios exámenes clínicos (1:N)
- Llave foránea: Examen_clinico.folio_id referencia a Paciente.folio_id

Examen_clinico ↔ Prueba_sangre:

- Un examen clínico puede estar asociado con una o más pruebas de sangre (1:N).
- Llave foránea: Prueba_sangre.examen_id referencia Examen_clinico.examen_id.

Examen_clinico ↔ Perfil_lipidico

- Un examen clínico puede estar asociado con un perfil lipídico (1:1).
- Llave foránea: Perfil_lipidico.examen_id referencia a Examen_clinico.examen_id.

Examen_clinico ↔ Marcadores_nutricionales

- Un examen clínico puede estar asociado con un conjunto de marcadores nutricionales (1:1).
Llave foránea:
Marcadores_nutricionales.examen_id referencia a Examen_clinico.examen_id.

Examen_clinico ↔ Marcadores_inflamatorios

- Un examen clínico puede estar asociado con marcadores inflamatorios (1:1).
- Llave foránea:
Marcadores_inflamatorios.examen_id referencia a Examen_clinico.examen_id.

Examen_clinico ↔ Marcadores_inflamatorios

- Un paciente puede tener mediciones físicas específicas (1:1).
- Llave foránea: Mediciones_fisicas.folio_id referencia a Paciente.folio_id.

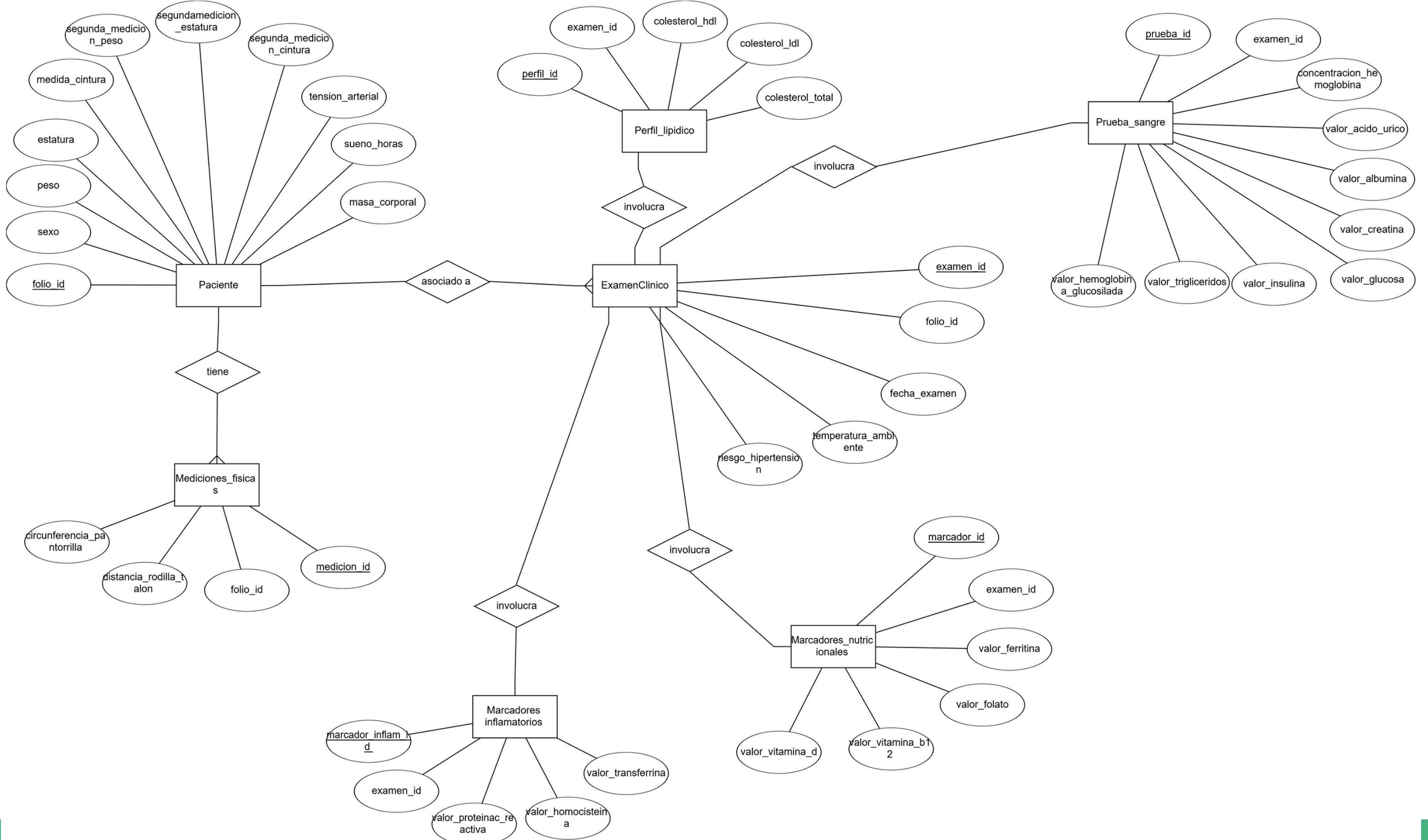


Diagrama Entidad - Relación DER

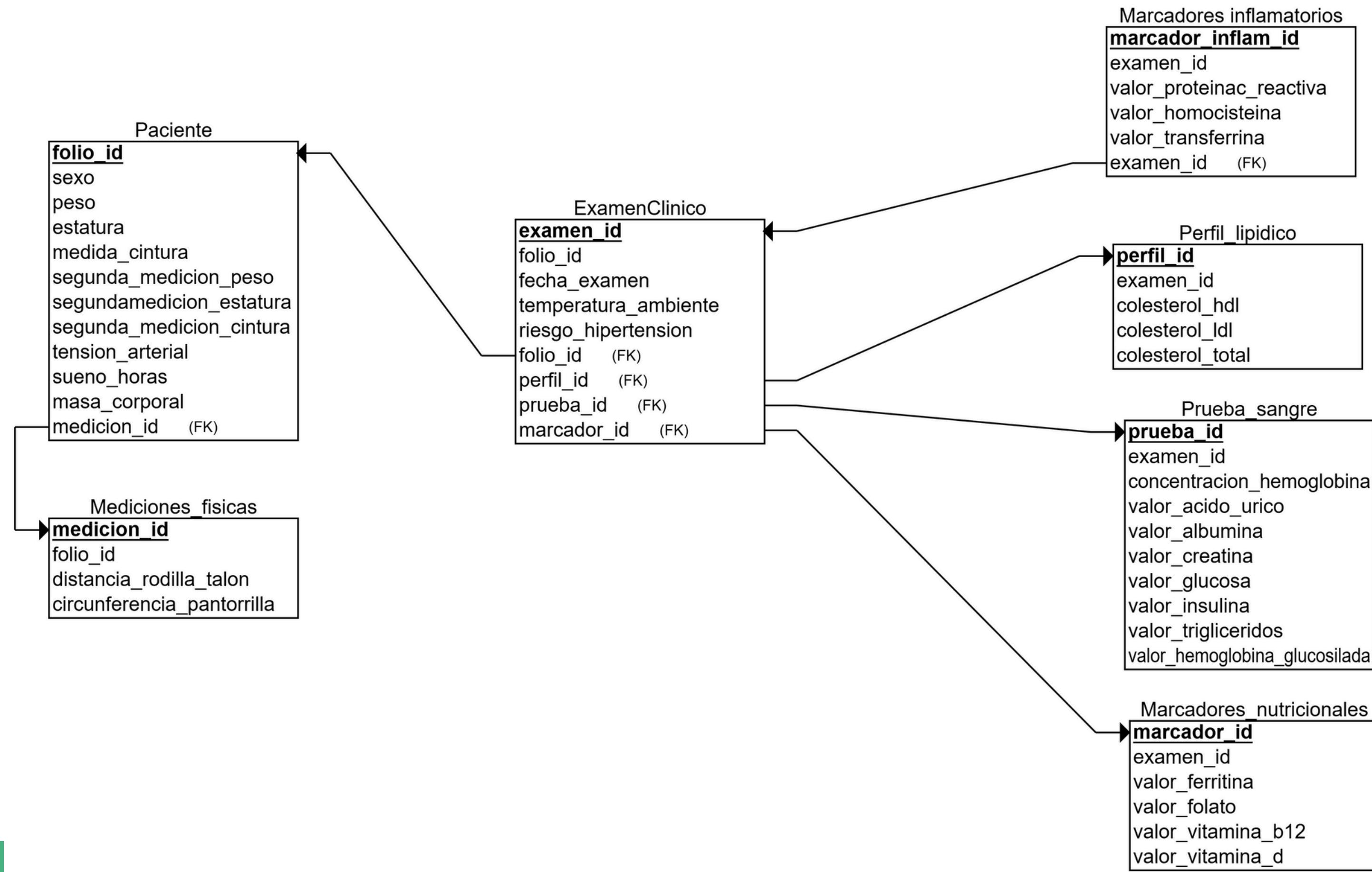


Diagrama de Tablas

3- Implementación



Herramientas Utilizadas



Sistema de gestión de
bases de datos
relacional



Herramienta de
administración que facilitó
la creación, consulta y
diseño de tablas.

3- Implementación

Manos a la obra...

Para la creación de datos ficticios
en base al esquema realizado:



Pasos en la implementación:

3- Implementación

Manos a la obra...

- Diseño del esquema relacional:
 - Basado en el DER
 - Definición de entidades y atributos

- Creación de las tablas en SQL:
 - Tablas principales como Paciente, Examen_Clinico, Pruebas_Sangre, Perfil_Lipidico, etc.
 - Uso de sentencias **CREATE TABLE** para definir la estructura de cada tabla.

Ejemplo de creación

```
-- Tabla: Paciente
CREATE TABLE Paciente (
    folio_id VARCHAR(15) PRIMARY KEY, -- Identificador único del paciente
    sexo VARCHAR(10) NOT NULL CHECK (sexo IN ('Masculino', 'Femenino')), -- Restricción para sexo
    edad INTEGER NOT NULL CHECK (edad >= 0), -- Restricción: edad no negativa
    peso NUMERIC(5,2) NOT NULL CHECK (peso >= 0), -- Restricción: peso no negativo
    estatura NUMERIC(5,2) NOT NULL CHECK (estatura >= 0), -- Restricción: estatura no negativa
    medida_cintura NUMERIC(5,2) NOT NULL CHECK (medida_cintura >= 0), -- Restricción: medida de cintura no negativa
    segundamedicion_peso NUMERIC(5,2) CHECK (segundamedicion_peso >= 0), -- Restricción opcional
    segundamedicion_estatura NUMERIC(5,2) CHECK (segundamedicion_estatura >= 0), -- Restricción opcional
    segundamedicion_cintura NUMERIC(5,2) CHECK (segundamedicion_cintura >= 0), -- Restricción opcional
    tension_arterial VARCHAR(15), -- Medición de la presión arterial
    sueno_horas NUMERIC(3,1) CHECK (sueno_horas >= 0), -- Restricción: horas de sueño no negativas
    masa_corporal NUMERIC(5,2), -- Índice de masa corporal
    actividad_total NUMERIC(5,2) CHECK (actividad_total >= 0) -- Restricción: actividad física no negativa
);
```

Pasos en la implementación:

- Inserción de datos
 - Uso de datos ficticios para simular la información clínica y bioquímica.
 - Sentencias **INSERT INTO** para alimentar las tablas.

3- Implementación

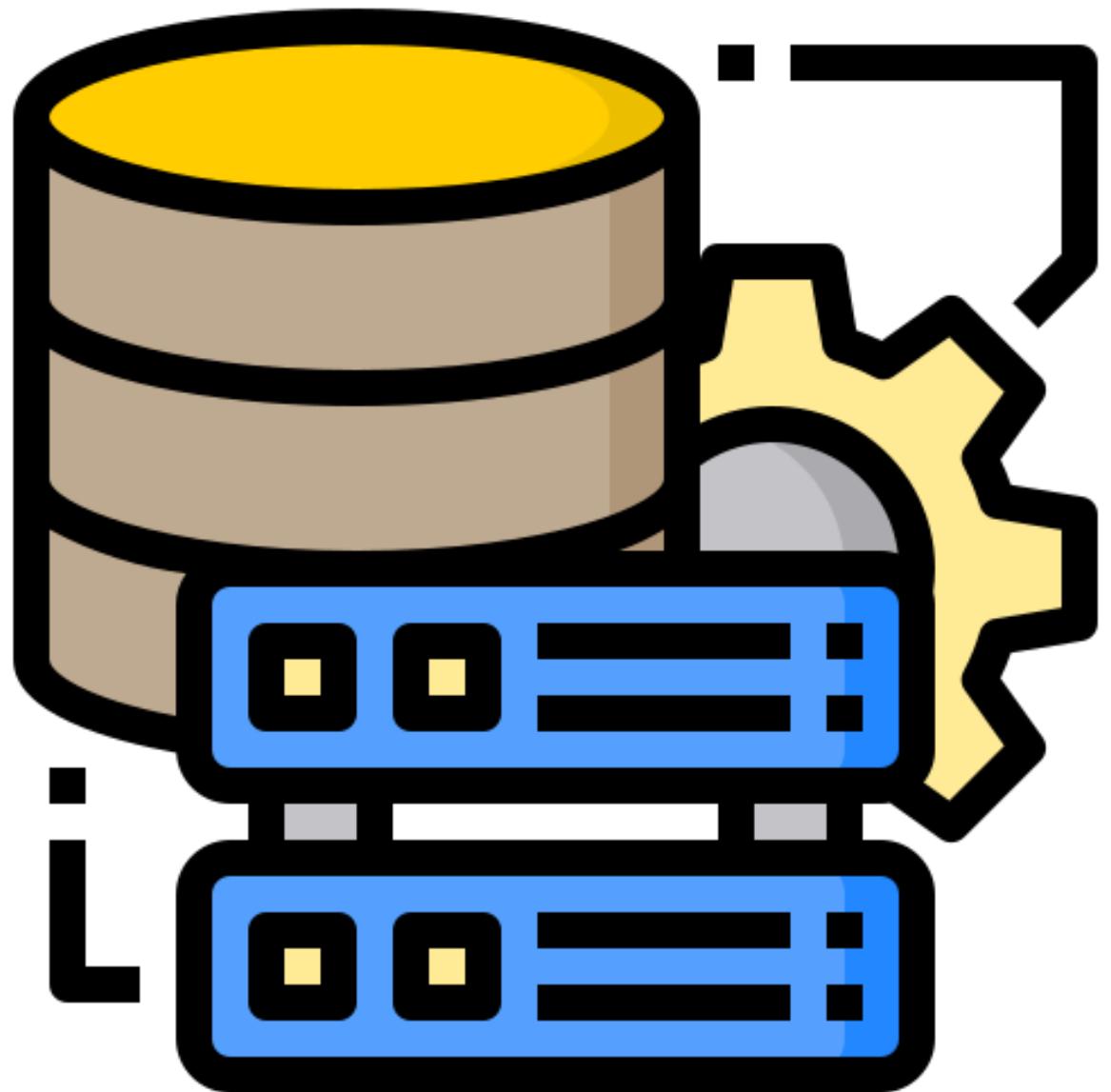
Manos a la obra...

```
-- Inserta registros en la tabla Paciente  
INSERT INTO Paciente (folio_id, sexo, edad, peso, estatura,  
medida_cintura, segundamedicion_peso, segundamedicion_estatura,  
segundamedicion_cintura, tension_arterial, sueno_horas,  
masa_corporal, actividad_total) VALUES  
(‘1000’, ‘F’, 80, 62.03, 1.75, 91.99, 60.39, 1.75, 91.55, ‘132/75’, 6.3,  
20.25, 159.92)
```

.....

```
-- Inserta registros en la tabla Examen_clinico  
INSERT INTO Examen_clinico (examen_id, folio_id, fecha_examen,  
temperatura_ambiente, riesgo_hipertension) VALUES  
(1, ‘1000’, ‘2021-07-20’, 27.1, True),
```

4 - Consultas y optimización heuristica



4 - Consultas y optimización heuristica

CONSULTA 1

Obtener los pacientes cuyo colesterol total no supera los 190 mg/dL y cuyo colesterol HDL es mayor a 50 mg/dL, junto con la información de la fecha del examen y el riesgo de hipertensión.

```
SELECT P.folio_id, E.fecha_examen, PL.colesterol_total,  
PL.colesterol_hdl, E.riesgo_hipertension  
  
FROM Paciente AS P  
  
JOIN Examen_clinico AS E ON P.folio_id = E.folio_id  
  
JOIN Perfil_lipidico AS PL ON E.examen_id = PL.examen_id  
  
WHERE PL.colesterol_total < 190 AND PL.colesterol_hdl > 50;
```

4 - Consultas y optimización heuristica

CONSULTA 1

Obtener los pacientes cuyo colesterol total no supera los 190 mg/dL y cuyo colesterol HDL es mayor a 50 mg/dL, junto con la información de la fecha del examen y el riesgo de hipertensión.

```
SELECT P.folio_id, E.fecha_examen, PL.colesterol_total,  
PL.colesterol_hdl, E.riesgo_hipertension  
  
FROM Paciente AS P  
  
JOIN Examen_clinico AS E ON P.folio_id = E.folio_id  
  
JOIN Perfil_lipidico AS PL ON E.examen_id = PL.examen_id  
  
WHERE PL.colesterol_total < 190 AND PL.colesterol_hdl > 50;
```

	folio_id character varying (15) 	fecha_examen date 	colesterol_total numeric (5,2) 	colesterol_hdl numeric (5,2) 	riesgo_hipertension boolean 
1	1000	2021-07-20	159.29	55.38	true
2	1002	2023-09-19	170.60	57.27	false
3	1005	2021-06-05	172.36	54.54	false
4	1014	2023-11-27	176.98	59.86	false
5	1017	2022-06-10	158.51	56.87	false
6	1018	2022-03-14	152.55	59.86	false
7	1020	2022-07-15	161.95	59.34	true
8	1021	2021-02-06	182.57	53.56	true
9	1022	2020-07-21	163.35	53.34	false
10	1024	2023-07-17	156.26	54.93	false
11	1026	2020-12-04	180.11	51.09	true
12	1029	2022-12-01	171.47	52.94	false

4 - Consultas y optimización heuristica

CONSULTA 2

Listar los pacientes que tienen niveles de glucosa en sangre superiores a 100 mg/dL y niveles de insulina mayores a 20 µU/mL.

```
SELECT P.folio_id, P.edad, PS.valor_glucosa, PS.valor_insulina,  
E.riesgo_hipertension  
FROM Paciente AS P  
JOIN Examen_clinico AS E ON P.folio_id = E.folio_id  
JOIN Prueba_sangre AS PS ON E.examen_id = PS.examen_id  
WHERE PS.valor_glucosa > 100 AND PS.valor_insulina > 20;
```

4 - Consultas y optimización heuristica

CONSULTA 2

Listar los pacientes que tienen niveles de glucosa en sangre superiores a 100 mg/dL y niveles de insulina mayores a 20 µU/mL.

```
SELECT P.folio_id, P.edad, PS.valor_glucosa, PS.valor_insulina,  
E.riesgo_hipertension  
FROM Paciente AS P  
JOIN Examen_clinico AS E ON P.folio_id = E.folio_id  
JOIN Prueba_sangre AS PS ON E.examen_id = PS.examen_id  
WHERE PS.valor_glucosa > 100 AND PS.valor_insulina > 20;
```

	folio_id character varying (15)	edad integer	valor_glucosa numeric (5,2)	valor_insulina numeric (5,2)	riesgo_hipertension boolean
1	1053	54	101.03	24.76	true
2	1075	75	103.28	24.62	false
3	1076	31	104.15	21.59	true
4	1086	26	103.32	20.23	false
5	1087	72	108.59	21.13	true
6	1099	44	100.76	24.83	true
7	1113	64	103.99	22.05	false
8	1134	40	105.81	23.26	true
9	1139	41	104.29	21.80	true
10	1172	51	109.49	20.77	false
11	1194	20	101.94	22.58	false
12	1264	49	105.60	23.08	true
13	1285	61	103.99	23.42	true
14	1319	49	101.23	21.61	false

CONSULTA 3

Usando vistas: obtener los valores promedio de ferritina y folato en pacientes mayores de 50 años

- Crear una vista para los pacientes mayores de 50 años

```
CREATE VIEW Pacientes_Mayores50 AS  
SELECT folio_id, edad  
FROM Paciente  
WHERE edad > 50;
```

- Crear una vista que combine los datos de marcadores nutricionales y exámenes clínicos

```
CREATE VIEW Marcadores_Completos AS  
SELECT mn.valor_ferritina, mn.valor_folato, ec.folio_id  
FROM Marcadores_nutricionales mn  
JOIN Examen_clinico ec ON mn.examen_id = ec.examen_id;
```

- Calcular los promedios utilizando las vistas

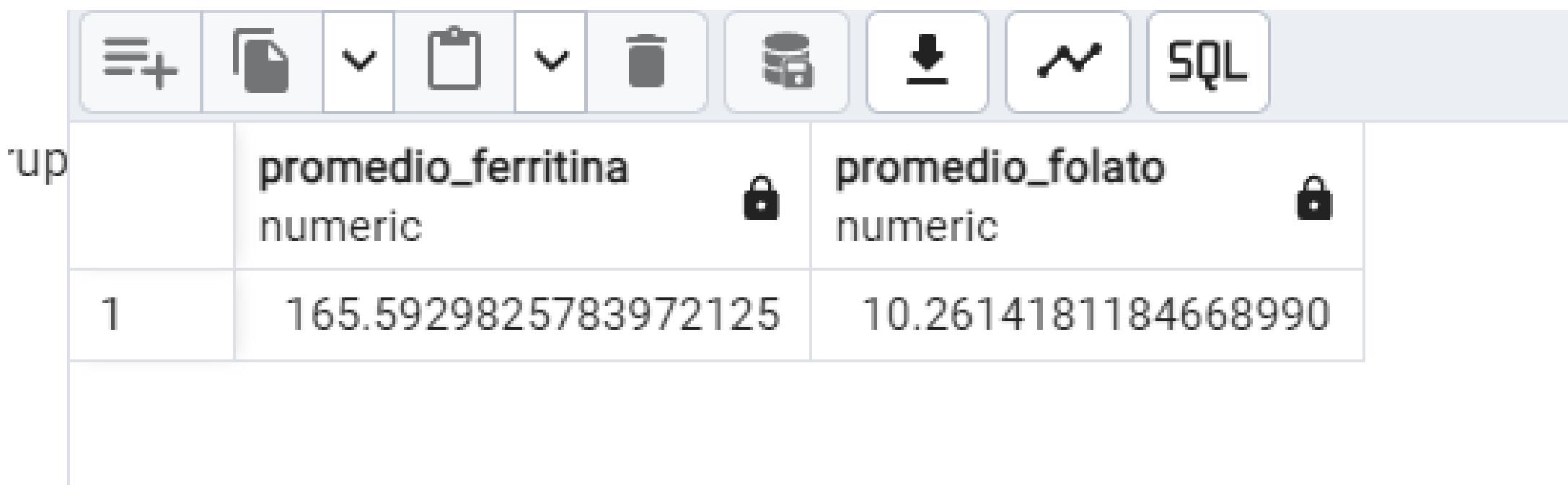
```
SELECT AVG(mc.valor_ferritina) AS promedio_ferritina,  
      AVG(mc.valor_folato) AS promedio_folato  
FROM Marcadores_Completos mc  
JOIN Pacientes_Mayores50 pm ON mc.folio_id = pm.folio_id;
```

4 - Consultas y optimización heuristica

4 - Consultas y optimización heuristica

CONSULTA 3

Usando vistas: obtener los valores promedio de ferritina y folato en pacientes mayores de 50 años



The screenshot shows a database interface with a toolbar at the top containing various icons for operations like insert, delete, and export. Below the toolbar is a results grid displaying two rows of data. The first row is a header row with column names and types: 'promedio_ferritina' (numeric) and 'promedio_folato' (numeric). The second row contains the actual values: 165.5929825783972125 and 10.2614181184668990 respectively. Both columns have a lock icon next to their names.

	promedio_ferritina numeric	promedio_folato numeric
1	165.5929825783972125	10.2614181184668990

4 - Optimización Heuristica

ÁRBOLES DE CONSULTA

Representación gráfica de una consulta en álgebra relacional, donde:

- Nodos hoja: Relaciones de entrada (tablas).
 - Nodos internos: Operaciones de álgebra relacional (selección, proyección, unión, etc.).
-
- **Procesamiento:**
 - Se ejecutan las operaciones internas cuando sus operandos están disponibles.
 - Los resultados intermedios reemplazan a los nodos procesados.
 - El proceso avanza hacia el nodo raíz, generando la relación final.

4 - Optimización Heuristica

OPTIMIZACIÓN HEURISTICA

El analizador genera un árbol de consultas basado en álgebra relacional, pero suele ser ineficiente.

- **Optimización:**
 - Transformación del árbol inicial en uno equivalente más eficiente.
 - Reducción de costos de ejecución y mejora del rendimiento.

Conceptos previos

4 - Optimización Heuristica

REGLAS DE LA HEURISTICA

- 1) **Aplicar primero las operaciones SELECT más restrictivas:** Reducen el número de filas (tuplas) lo antes posible.
- 2) **Ejecutar operaciones PROJECT tempranas:** Reducen el número de columnas (atributos)
- 3) **Mover las operaciones SELECT y PROJECT lo más abajo posible**
Aplicarlas cerca de las hojas para reducir los datos en etapas iniciales.
- 4) **Reemplazar productos cartesianos por operaciones JOIN**
- 5) **Priorizar SELECT y JOIN restrictivas**
- 6) **Eliminar operaciones redundantes:** Evitar repeticiones innecesarias de SELECT o PROJECT.
- 7) **Ejecutar operaciones de agregación al final:** Realizar cálculos como SUM, AVG o COUNT en los datos ya reducidos.

4 - Optimización Heurística

CONSULTA

Seleccionar los pacientes que tengan un nivel de glucosa mayor a 100 mg/dL, dentro de un rango específico de edad (por ejemplo entre 30 y 50 años). Recoleistar el ID del paciente, su edad, la fecha del examen, el valor de glucosa y el riesgo de hipertensión

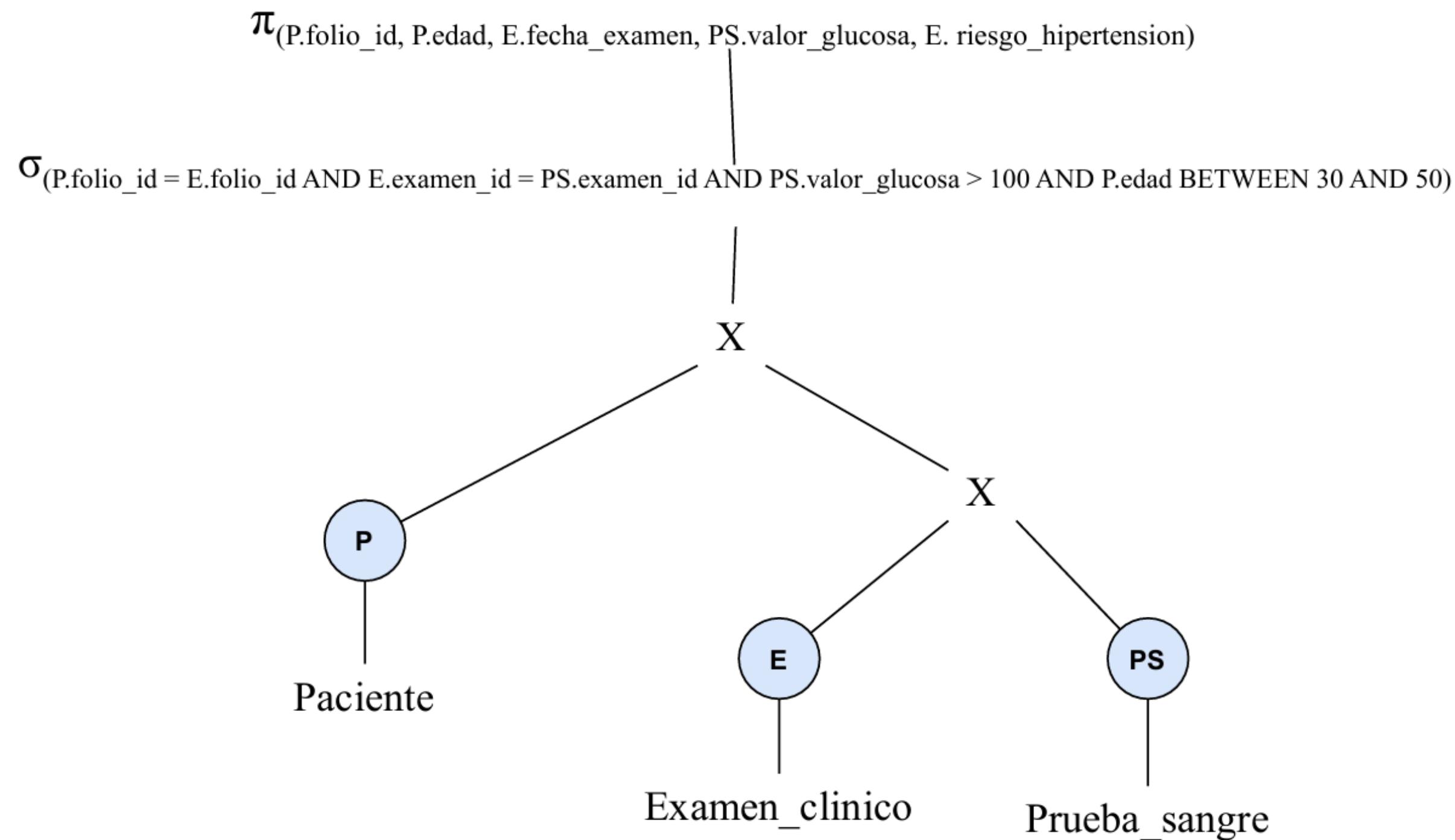
Consulta inicial

```
SELECT P.folio_id, P.edad, E.fecha_examen, PS.valor_glucosa,  
E.riesgo_hipertension  
FROM Paciente P, Examen_clinico E, Prueba_sangre PS  
WHERE P.folio_id = E.folio_id AND  
      E.examen_id = PS.examen_id AND  
      PS.valor_glucosa > 100 AND  
      P.edad BETWEEN 30 AND 50;
```

4 - Optimización Heurística

ÁRBOL INICIAL

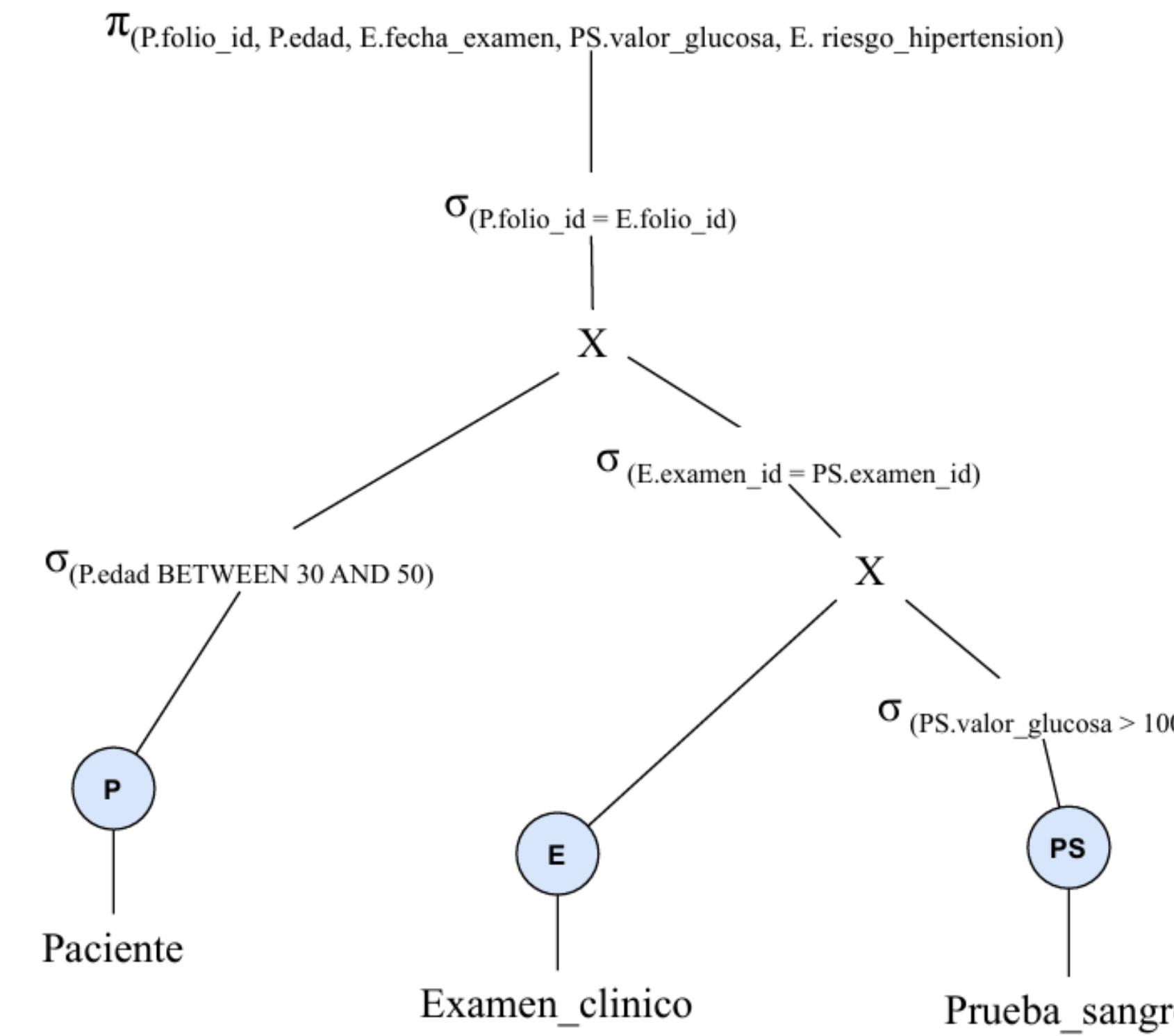
El árbol inicial representa la consulta SQL, donde todas las condiciones del WHERE y las columnas especificadas en el SELECT están agrupadas en una misma operación de selección y proyección.



4 - Optimización Heurística

ÁRBOL SECUNDARIO: DESCOMPOSICIÓN DE SELECCIONES

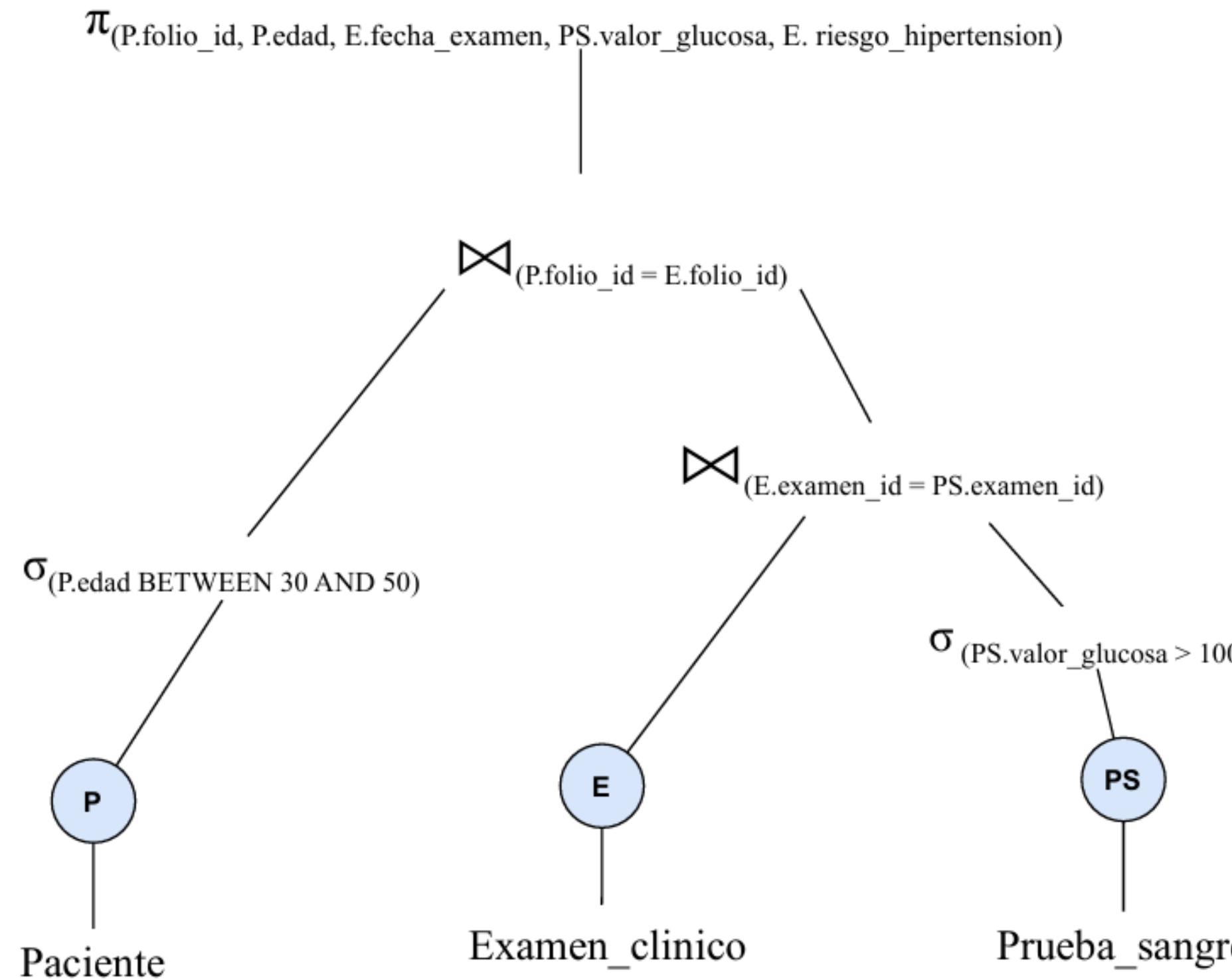
En este caso se hace para reducir la cantidad de datos manejados. Se separan las condiciones de selección en operaciones independientes:



4 - Optimización Heurística

ÁRBOL TERCIARIO: REEMPLAZO DE SELECCIONES POR UNIONES NATURALES (\bowtie):

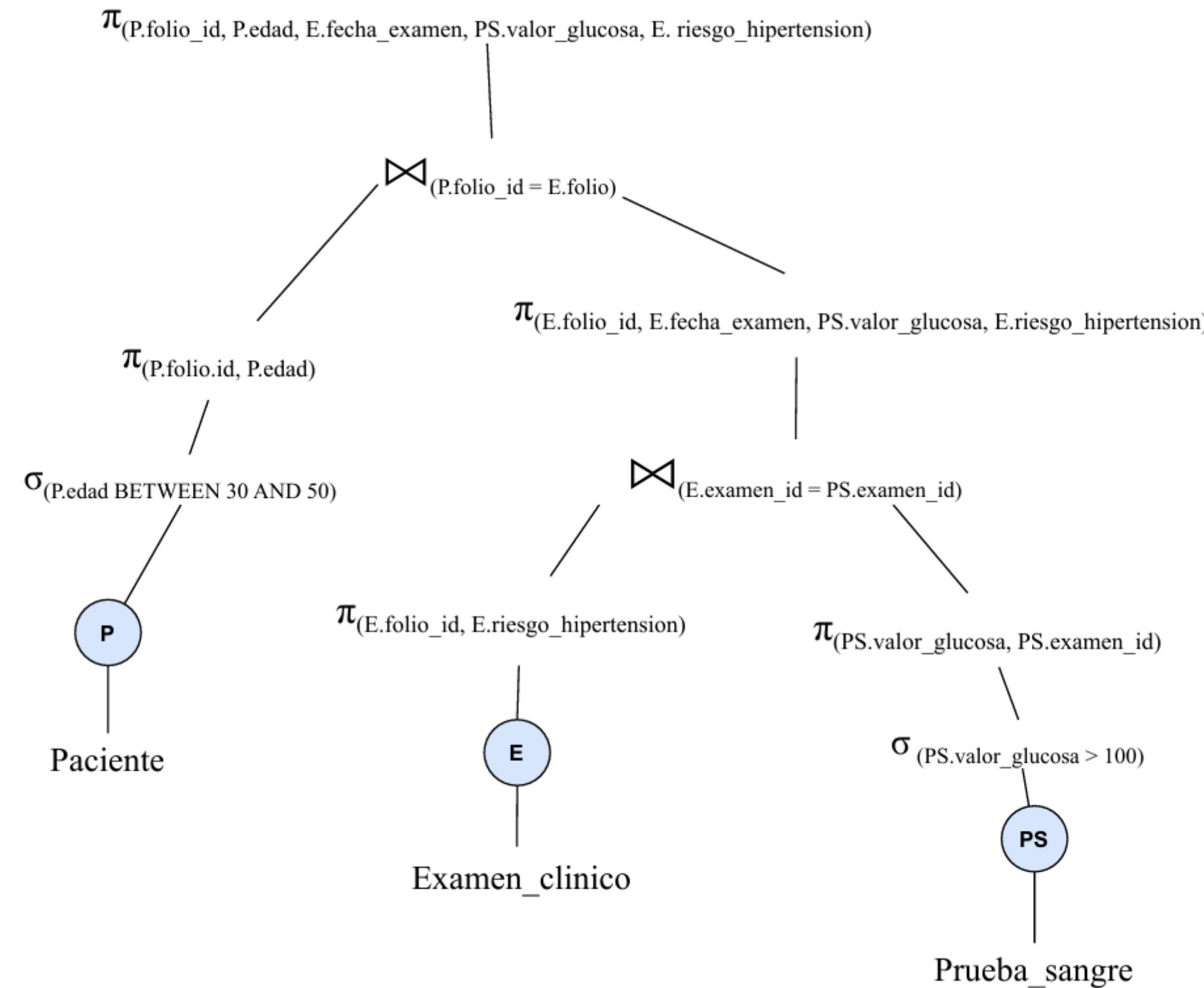
se hace para simplificar y mejorar el rendimiento



4 - Optimización Heurística

ÁRBOL CUATERNARIO: APLICACIÓN TEMPRANA DE PROYECCIONES (Π)

En lugar de cargar todas las columnas de las relaciones, se seleccionan solo las necesarias, se hace bajando las selecciones. Este paso disminuye el volumen de datos que pasa por las uniones y reduce el uso de memoria y tiempo de ejecución.

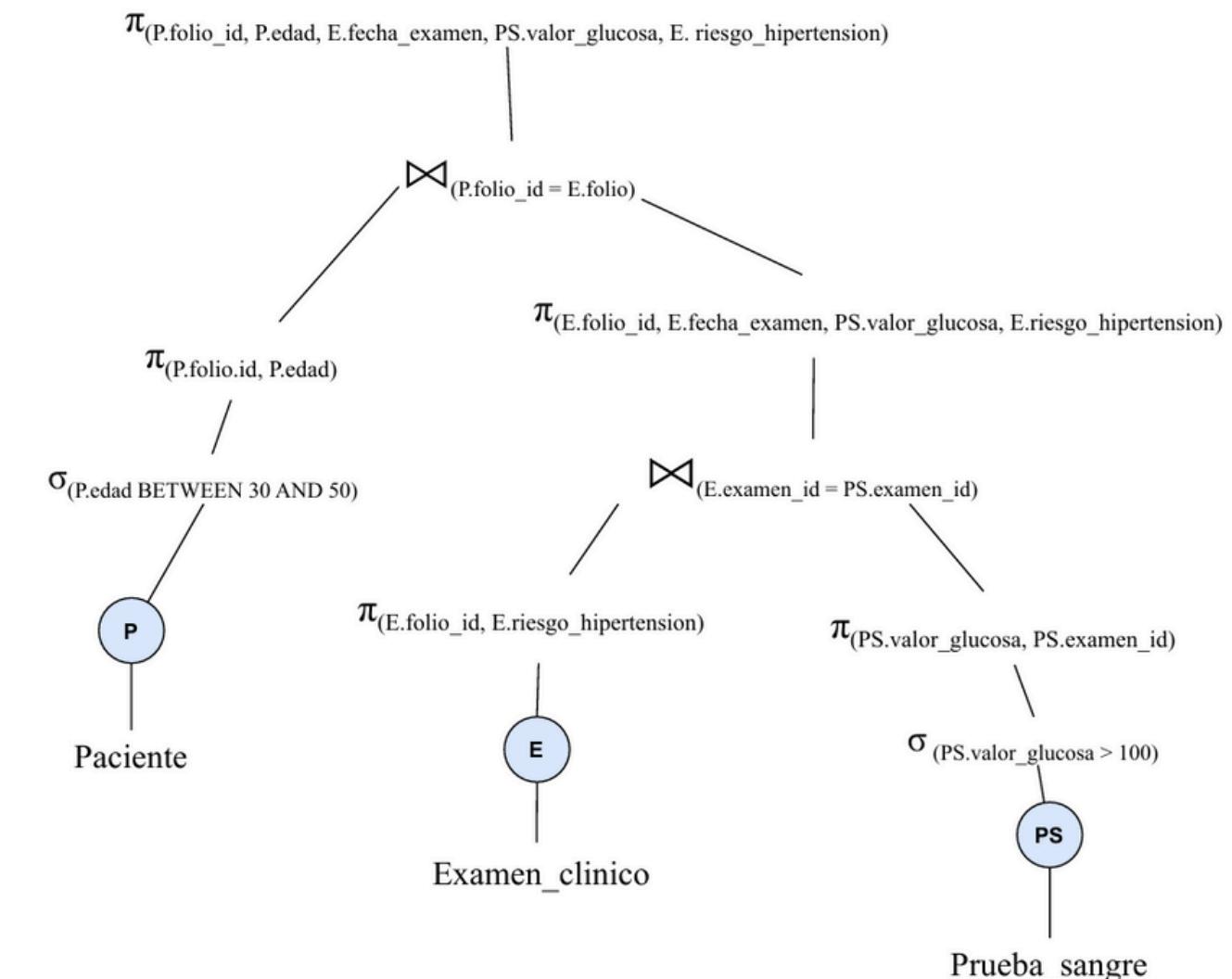


4 - Optimización Heuristica

Árbol optimizado

El arbol final anterior organiza las operaciones para minimizar costos:

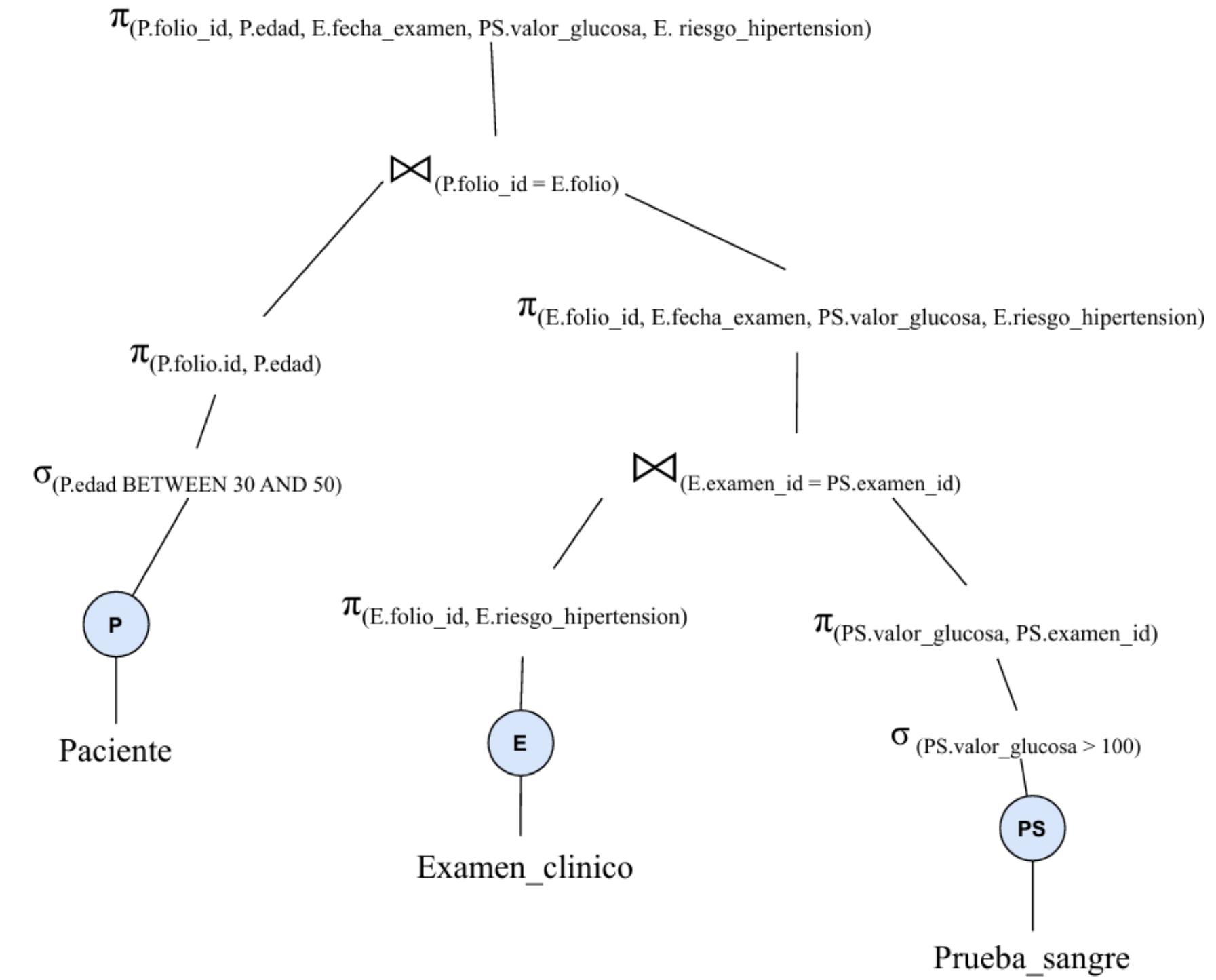
- 1. Selecciones tempranas:** Se eliminan las filas irrelevantes en etapas iniciales
- 2. Uniones naturales:** Se combinan solo las filas necesarias, con atributos claves
- 3. Proyecciones:** Se eliminan las columnas innecesarias para reducir mas el tamaño de los datos
- 4. Proyección final:** Produce el resultado final con las columnas especificadas en el SELECT



Consulta Optimizada

```

SELECT P.folio_id, P.edad, E.fecha_examen,
PS.valor_glucosa, E.riesgo_hipertension
FROM ( SELECT folio_id, edad
      FROM Paciente
      WHERE edad BETWEEN 30 AND 50) AS P
JOIN (
      SELECT folio_id, examen_id, fecha_examen,
      riesgo_hipertension
      FROM Examen_clinico) AS E ON P.folio_id =
E.folio_id
JOIN ( SELECT examen_id, valor_glucosa
      FROM Prueba_sangre
      WHERE valor_glucosa > 100) AS PS
      ON E.examen_id = PS.examen_id;
  
```



5- Desarrollo de Interfaz Gráfica



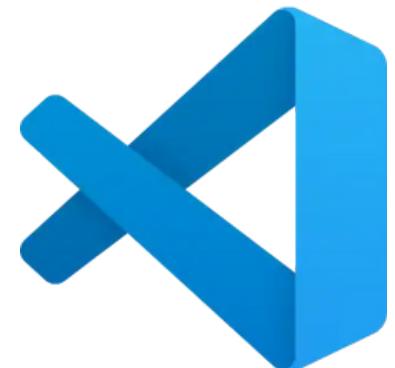
5- Desarrollo de Interfaz Gráfica



Herramientas Utilizadas



matplotlib

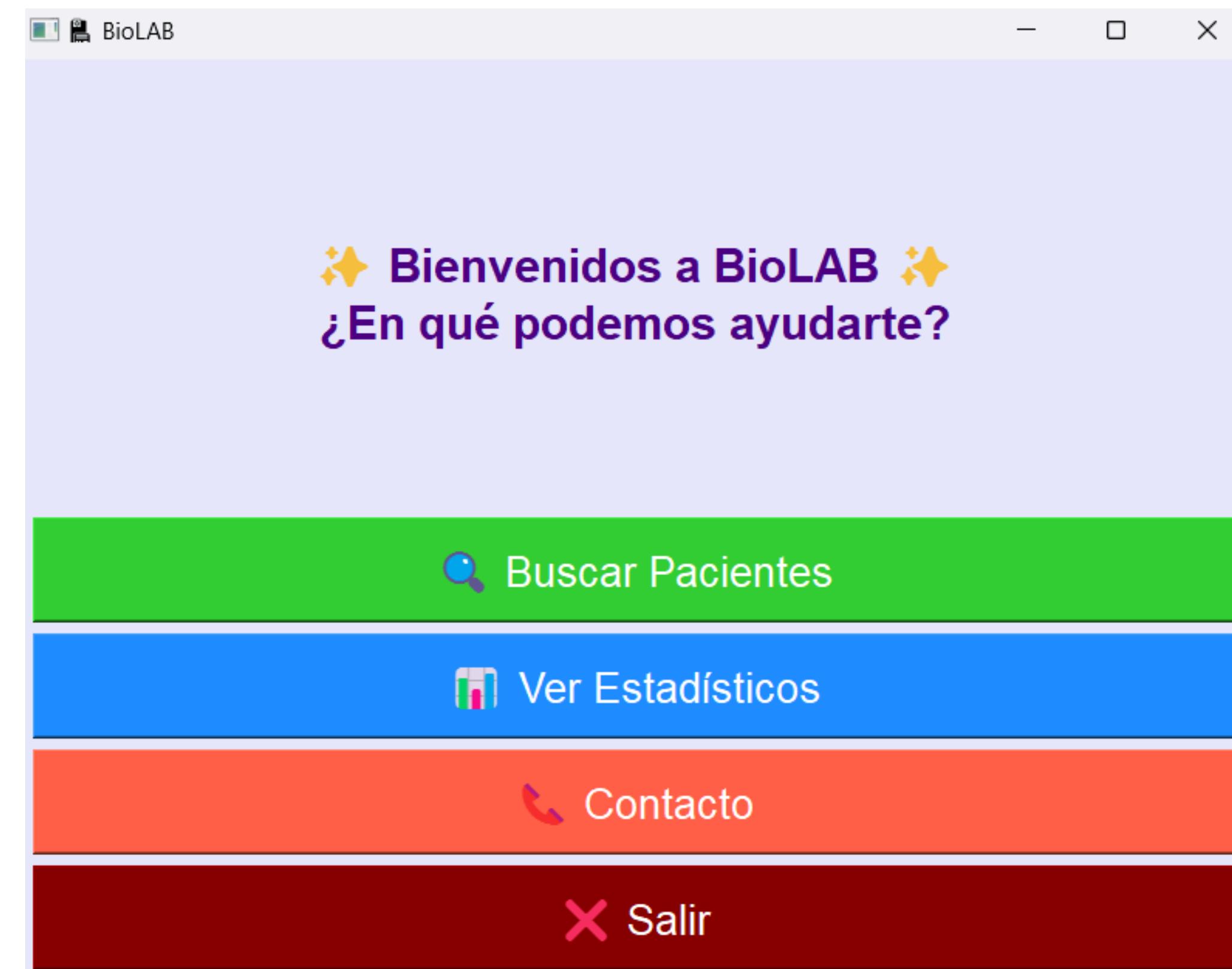


Github

Estructura de la interfaz

4 - Desarrollo de Interfaz Gráfica

Ventana principal



Estructura de la interfaz

4 - Desarrollo de Interfaz Gráfica

Botón “Buscar pacientes”

The screenshot shows a graphical user interface for searching patients. At the top, there is a green header bar with the text "Buscar Pacientes" and a magnifying glass icon. Below this is a light gray main area containing two input fields: one for searching by folio (labeled "ID") and one for searching by sex (labeled "Sexo"). Both fields have placeholder text: "Ingresa el folio del paciente" and "Ingresa el sexo del paciente (M/F)". Below these fields is a green button labeled "Buscar" with a magnifying glass icon. At the bottom, there is a table with columns for "Folio", "Sexo", "Edad", and "Detalles". The entire interface has a clean, modern design with a white background and clear labeling.

Estructura de la interfaz

4 - Desarrollo de Interfaz Gráfica

Busqueda por filtro

The screenshot displays a graphical user interface for searching patient information. At the top, there is a search bar containing the number "1000". Below it, a pink-bordered input field is labeled "Buscar por sexo:" and contains the text "Ingrresa el sexo del paciente (M/F)". To the right of this field is a green button with a magnifying glass icon and the word "Buscar". At the bottom, a table lists patient data. The columns are labeled "Folio", "Sexo", "Edad", and "Detalles". The first row shows Folio 1, Sexo F, Edad 80, and a yellow "Ver Detalles" button under Detalles.

	Folio	Sexo	Edad	Detalles
1	1000	F	80	Ver Detalles

Estructura de la interfaz

4 - Desarrollo de Interfaz Gráfica

Busqueda por filtro

The screenshot shows a graphical user interface for searching patients. At the top, there is a search bar containing the text "1000". Below the search bar, the text "Buscar por sexo:" is displayed, followed by a pink input field labeled "Ingresa el sexo del paciente (M/F)". To the right of this input field, two green callout bubbles contain SQL queries: "SELECT * FROM Paciente WHERE folio_id = '---'" and "SELECT * FROM Paciente WHERE sexo = '---'". A large green button at the bottom right features a magnifying glass icon and the word "Buscar". At the bottom of the interface, there is a table with four columns: "Folio", "Sexo", "Edad", and "Detalles". The first row of the table contains the values "1 1000", "F", "80", and a yellow button labeled "Ver Detalles".

	Folio	Sexo	Edad	Detalles
1	1000	F	80	Ver Detalles

Estructura de la interfaz

4 - Desarrollo de Interfaz Gráfica

Busqueda por filtro

The screenshot displays a graphical user interface with a sidebar on the left and a main content area on the right.

Search Bar: At the top, there is a search bar labeled "Buscar por sexo:" containing the letter "M". To the right of the search bar is a green button with a magnifying glass icon labeled "Buscar".

Table: Below the search bar is a table with the following columns: Folio, Sexo, Edad, and Detalles. The table contains 11 rows of data, all showing "M" in the Sexo column and "Ver Detalles" in the Detalles column.

	Folio	Sexo	Edad	Detalles
1	1006	M	40	Ver Detalles
2	1008	M	56	Ver Detalles
3	1011	M	55	Ver Detalles
4	1012	M	25	Ver Detalles
5	1015	M	74	Ver Detalles
6	1016	M	56	Ver Detalles
7	1018	M	32	Ver Detalles
8	1019	M	20	Ver Detalles
9	1020	M	76	Ver Detalles
10	1021	M	40	Ver Detalles
11	1022	M	42	Ver Detalles

Sidebar: On the left side, there is a sidebar with several items:

- A purple square containing the number "1000".
- A blue square containing the number "1000".
- A pink square containing the text "Ingresos".
- A green square.
- A white square containing the number "1" and the number "1000".

Estructura de la interfaz

4 - Desarrollo de Interfaz Gráfica

Detalles de examen clínico

Detalles del Paciente: 1000

Detalles del Paciente con Folio: 1000

	Campo	Valor
1	folio_id	1000
2	sexo	F
3	edad	80
4	peso	62.03
5	estatura	1.75
6	medida_cintura	91.99
7	segundamedicion...	60.39
8	segundamedicion...	1.75
9	segundamedicion...	91.55
10	tension_arterial	132/75
11	sueno_horas	6.3
12	masa_corporal	20.25
13	actividad_total	159.92

Estructura de la interfaz

4 - Desarrollo de Interfaz Gráfica

Botón “Ver estadísticos”



Estructura de la interfaz

4 - Desarrollo de Interfaz Gráfica

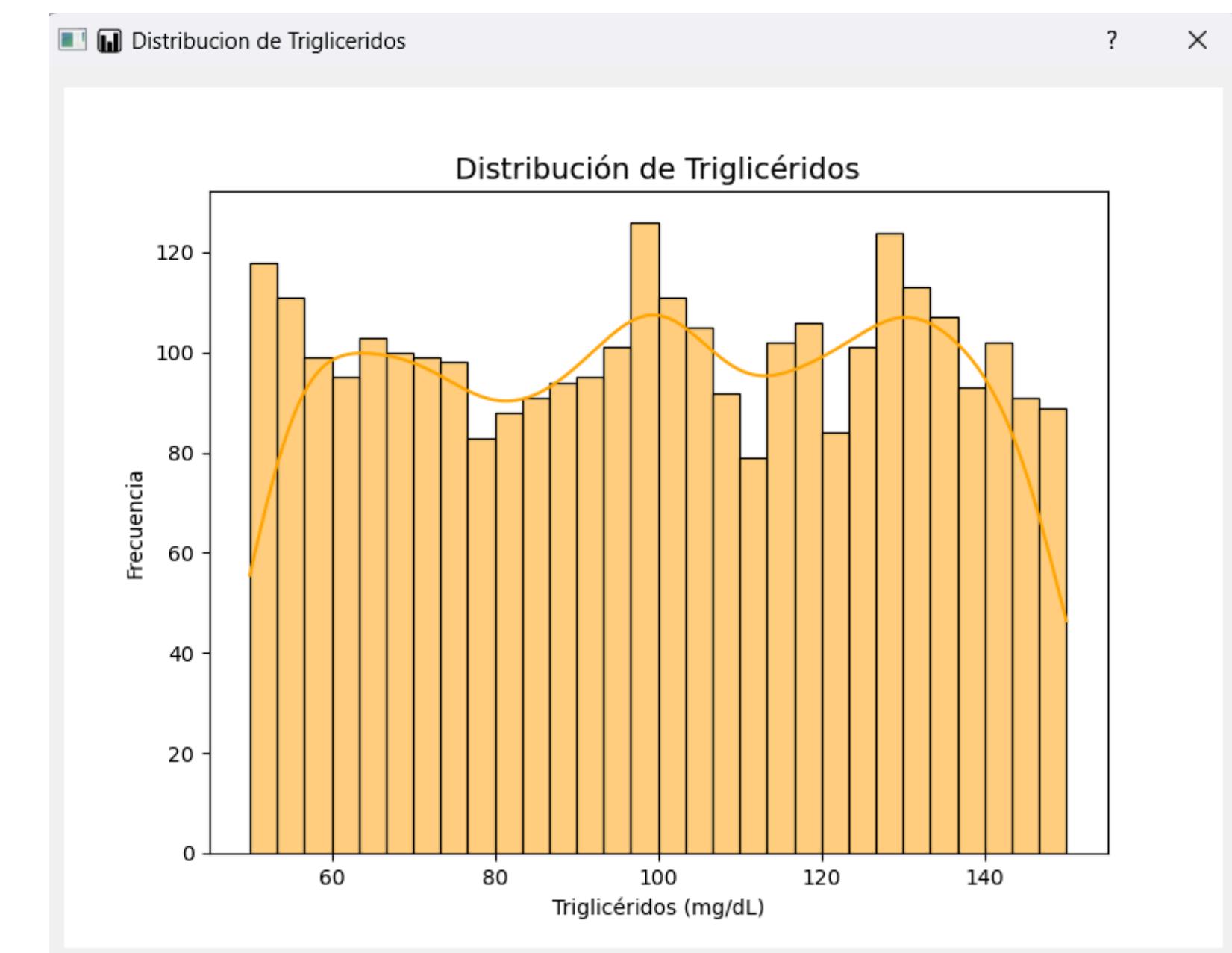
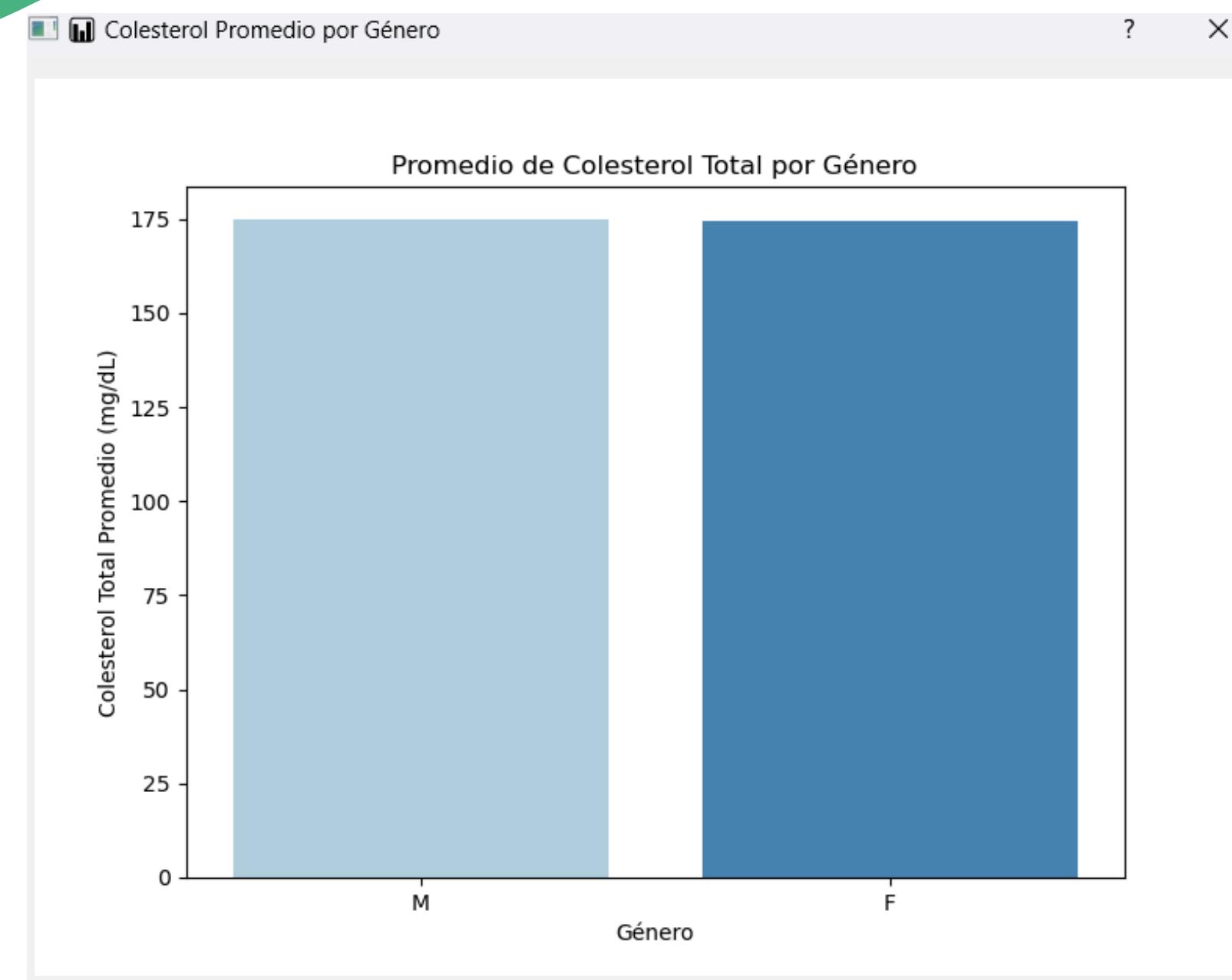
Visualización de graficas



Estructura de la interfaz

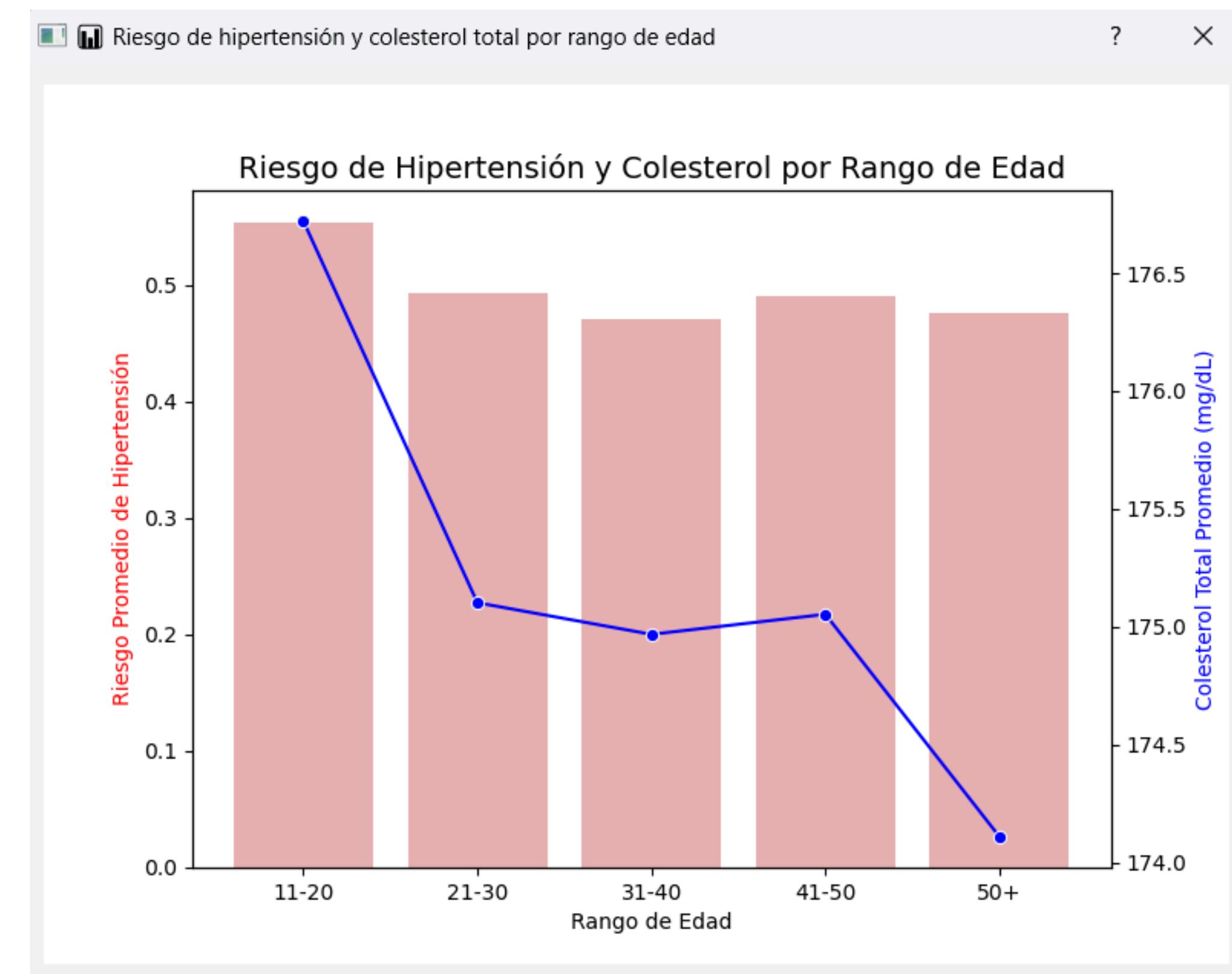
4 - Desarrollo de Interfaz Gráfica

Visualización de graficas



4 - Desarrollo de Interfaz Gráfica

Visualización de graficas



6 - Resultados y conclusiones



Resultados

- ✓ Se diseñó e implementó una base de datos escalable en PostgreSQL, con un modelo relacional eficiente que asegura integridad referencial y elimina redundancias.
- ✓ Se optimizó una consulta mediante reglas heurísticas, mejorando el rendimiento y reduciendo costos de ejecución.
- ✓ Se desarrolló una interfaz gráfica intuitiva con PyQt, que incluyó módulos para la búsqueda de pacientes, visualización de estadísticas y análisis interactivo con gráficos.
- ✓ El sistema resultante facilita el análisis clínico y bioquímico, siendo una herramienta útil para la toma de decisiones en salud pública.



Conclusiones

- 💡 El proyecto consolidó conocimientos técnicos en diseño e implementación de bases de datos, optimización de consultas y desarrollo de interfaces gráficas.
- 💡 La solución creada es eficiente, práctica y escalable, con capacidad para adaptarse a futuros requerimientos o mayores volúmenes de datos.
- 💡 La integración de información clínica y bioquímica demuestra el valor de las bases de datos en la gestión de factores de riesgo en salud pública.
- 💡 Idem al anterior, se destaca el rol como futuro bioinformático en buscar solución a manipular grandes cantidades de datos medicos como en este caso.



Bibliografia

- Fundamentos de bases de datos (Elmasri, Navathe): Elmasri, R., & Navathe, S. B. (2020). Fundamentos de bases de datos (7a ed.). Pearson.
- Web de Python Docs: Python Software Foundation. (2023). Python Documentation.
<https://docs.python.org/3/>
- Web de Qt Designer: The Qt Company. (2023). Qt Designer. <https://www.qt.io/qt-designer>
- Web de Matplotlib: Matplotlib Developers. (2023). Matplotlib Documentation.
<https://matplotlib.org/stable/contents.html>
- Web de Kaggle: Kaggle, Inc. (2023). Kaggle Datasets. <https://www.kaggle.com/datasets>.
- Rios, M. (2024). BioLab-TPFinal-BD [Repositorio en GitHub]. Recuperado de
<https://github.com/MatiasRiosMR/BioLab-TPFinal-BD>



Matias Rios

matias.rios@ingenieria.uner.edu.ar

Licenciatura en Bioinformática

Bases de Datos

Diciembre 2024

¡Gracias por su atención!

Puede encontrar el proyecto realizado
escaneando el siguiente QR o accediendo a :

github.com/MatiasRiosMR/BioLab-TPFinal-BD

