



UNIVERSIDAD TECNICA PARTICULAR DE LOJA

“CIENCIAS DE LA COMPUTACIÓN”

Prácticum 1.2

BASE DE DATOS AVANZADO

Nombre Estudiante:

Carlos Agustin Salas Churo

Nombre Docente:

Juan Carlos Morocho

Tema:

Proyecto Integrador de Saberes

Periodo:

Abril 2023 - Agosto 2023

Contenido

Introducción	3
Herramientas	4
Planificación del Trabajo	4
Repositorio GitHub	4
Datos Base	4
Datos Complementarios	10
Posibles análisis para realizar	11
Diseño Conceptual	13
Diseño Lógico Relacional	13
Diseño Físico	14
Implementación y Carga	14
Script SQL	32
Descarga en formato CSV	32
Bibliografía	33

Introducción:

El proyecto integrador de saberes es un proyecto coordinado desde la asignatura Prácticum 1.2, diseñado con el propósito de aplicar las capacidades y competencias adquiridas durante los primeros niveles de la carrera de Computación, específicamente en las áreas de Gestión de Datos y Programación de aplicaciones. Este proyecto involucra a las asignaturas de cuarto ciclo, Base de Datos Avanzada y Programación Avanzada, para desarrollar una solución centrada en el análisis de datos reales relacionados con problemas de interés social.

El objetivo principal es analizar datos específicos relacionados con la información de acceso y las condiciones de vivienda en el Ecuador. A través del uso de técnicas de exploración y visualización de la información, se busca desarrollar e implementar una solución informática que utilice los conocimientos adquiridos en las áreas de bases de datos y programación. Para ello, se emplearán bases de datos MySQL y entornos interactivos de análisis de datos como Apache Zeppelin.

El resultado final de este proyecto estará orientado al análisis e interpretación de los datos recopilados, con el fin de obtener hallazgos relevantes que aporten información y conocimiento sobre la situación del acceso y las condiciones de vivienda en el país.

Herramientas

- GitHub: Para guardar el proyecto dentro de un repositorio donde se encontraran recursos y archivos relacionados con el proyecto.
- Draw.io: Se uso en la creación los modelos de diseño conceptual y lógico de la base de datos. Draw.io es excelente para crear diagramas de manera eficiente y proporciona una variedad de plantillas y opciones de personalización para adaptarse a diferentes necesidades de modelado de datos.
- MySQL Workbench: Para el diseño físico de la base de datos. Esta herramienta proporciona una variedad de características para el diseño de bases de datos, incluyendo la capacidad de diseñar modelos físicos, generar scripts SQL a partir de modelos, y más.

Planificación del Trabajo

Repositorio GitHub

Enlace al repositorio GitHub del Proyecto Integrador de Saberes:

https://github.com/CarlosSalas8/Proyecto_Integrador.git

Datos Base

- Fuente: INEC
- Conjunto de datos base: <https://rebrand.ly/248avfq>
- Formato: CSV
- Volumen: 26580 registros

Los datos que usaremos son datos del INEC de las encuestas realizadas llamadas ENEMDU el formato es un CSV, el cual tiene un total de 57 columnas, tiene un volumen de 26580 registros, este archivo contiene información sobre las características de vivienda en el Ecuador, a partir de los resultados de encuesta ENEMDU realizados por el INEC en el primer trimestre del año 2023.

- **area:** Es un tipo de dato de texto, en la encuesta describe el área en donde se encuentra la vivienda, es un campo de selección el tipo de área entre rural y urbana.
- **ciudad:** Es un tipo de dato de texto, describe la ciudad en la que se realiza la encuesta por ende es la ciudad en la que se encuentra la vivienda.
- **conglomerado:** Es un tipo de dato numérico, describe al conjunto de vivienda donde se encuentran, para estar en el mismo conjunto deben pertenecer a la misma parroquia, cada parroquia se divide en zonas la amanzanada y la dispersa.
- **panelm:** Tipo de dato numérico, es una identificación que se les da a los conglomerados, cada panel está conformado por 7 viviendas “originales” y 3 de “reemplazo”.
- **vivienda:** Dato numérico, este dato es un identificador de la vivienda dentro del conglomerado
- **hogar:** es un identificador del hogar dentro de la vivienda
- **vi01:** Se refiere a la vía de acceso a la vivienda, es un tipo de dato numérico, describe mediante unas opciones la vía de acceso consta de 6 opciones para este campo.

- **vi02:** Se refiere al tipo de vivienda, es un dato de tipo numérico, consta de 7 opciones que describen el tipo de la vivienda.
- **vi03a:** Es la primera parte de un catálogo para describir el techo o cubierta de la vivienda, un tipo de dato numérico. En este apartado se describe el material del techo con 7 opciones.
- **vi03b:** Esta es la segunda y última parte del catálogo, este es un tipo de dato numérico, que describe el estado del techo o cubierta.
- **vi04a:** Este es un catálogo para describir el piso o suelo de la vivienda, en esta primera parte se describe el material del suelo de la vivienda con 8 opciones.
- **vi04b:** Es la segunda parte del catálogo, este es un dato numérico en el que se indica el estado del piso de la vivienda para lo cual se dan 3 opciones.
- **vi05a:** Es la primera parte de un catálogo que describe las paredes de la vivienda, es un tipo de dato numérico, en este apartado se describe el material de las paredes para lo cual se dan 7 opciones.
- **vi05b:** Es la segunda parte del catálogo, este es un tipo de dato numérico, en este apartado se indica el estado de las paredes para lo cual se dan 3 opciones.
- **vi06:** Es un dato numérico en el que se espera el número de cuartos de la vivienda, sin incluir cuartos de cocina, baño, garajes o los dedicados exclusivamente a negocios.
- **vi07:** Es un tipo de dato numérico donde se detalla el número de cuartos dedicados solo para dormir.

- **vi07a:** Es un tipo de dato numérico el cual detalla el número de cuartos utilizados exclusivamente para negocios, locales destinados a actividades comerciales en la vivienda.
- **vi07b:** Dato numérico, que indica el número de cuartos exclusivos para cocinar.
- **vi08:** Es un tipo de dato numérico, que indica el material con el que cocinan para lo cual se proporcionan 4 opciones
- **vi09:** Es el principio de un conjunto de catálogos, este primer apartado es un tipo de dato numérico, que describe el tipo de servicio higiénico con el que cuenta la vivienda para lo cual se dan 5 opciones, este es un caso especial pues si la respuesta es entre 1 y 4 se pasara a vi10.
- **vi09a:** Ya que la respuesta a vi09 fue 5, se pasa a este apartado de tipo numérico en el que se describe que realizan los miembros del hogar para lo cual se brinda 3 opciones de igual forma si se elige entre 1 y 10 se pasa a vi10
- **vi09b:** Es un tipo de dato numérico, en el que se detalla que tipo de instalación sanitaria utilizan los miembros del hogar.
- **vi10:** Es un tipo de dato numérico, en el cual se detalla de donde obtiene agua el hogar para lo cual se dan 7 opciones, en caso de que la opción sea 1 pasar a 10a y en caso de ser entre 4 y 7 igualmente se pasa a 10a
- **vi101:** Es un tipo de dato numérico, en el que se detalla si la vivienda posee un medidor de agua, en este campo tiene 2 opciones.
- **vi102:** Es un tipo de dato numérico, en el que se detalla si el agua que obtiene la vivienda es de la junta de agua para lo cual se dan 2 opciones.

- **vi10a:** Es un tipo de dato numérico, en este apartado se detalla de donde recibe agua la vivienda en este campo se tienen 4 opciones.
- **vi11:** Este es un dato numérico, donde se detalla que tipo de ducha tiene la vivienda para la que se ofrecen 3 opciones.
- **vi12:** Este dato numérico describe el tipo de alumbrado de vivienda para este campo con 4 opciones.
- **vi13:** Tipo de dato numérico en este campo se detalla de qué manera se elimina la mayor parte de la basura.
- **vi14:** Tipo de dato numérico, en el cual se detalla la forma que ocupa la vivienda en el hogar o la tenencia, en este campo se dan 7 opciones. En caso de elegir 1 pasar a las siguientes de lo contrario pasar a vi151.
- **vi141:** Es un dato numérico en el cual se espera el valor que estarían dispuestos a pagar por un arriendo
- **vi142:** Dato numérico, donde se detalla si su pago mensual de arriendo incluye un servicio de agua, para lo que tiene 2 opciones.
- **vi143:** Dato numérico, donde se detalla si su pago mensual de arriendo se incluye el de un servicio de luz, para lo que se dan 2 opciones.
- **vi144:** Es un tipo de dato numérico, en el cual se detalla si tiene algún parentesco con el propietario de la vivienda, en el cual tiene 2 opciones.
- **vi1511:** Dato numérico, describe si el hogar tiene vehículos indica 2 opciones para 2 tipos de vehículos en este caso, vehículos y motos en caso de elegir la opción 2 pasar a la siguiente sección
- **vi152:1** Dato numérico, indica el número de vehículos que tiene el hogar.

- **vi1522:** Dato numérico, indica el número de motos que tiene el hogar.
- **vi1531 a vi546:** Datos de tipo numérico describe el combustible más frecuente en la vivienda
 vi531 y vi541: Combustible Super y gasto mensual
 - vi1532 y vi1542: Combustible Extra y gasto mensual.
 - vi1533 y vi1543: Combustible Diésel y gasto mensual.
 - vi1534 y vi1544: Combustible Ecopais y gasto mensual.
 - vi1535 y vi1545: Combustible Electricidad y gasto mensual. [OBJ]
 - vi1536 y vi1546: Combustible Gas y gasto mensual.
- **estrato:** tipo de dato numérico, detalla el estrato socioeconómico al que pertenece la vivienda.
- **fexp:** Es un tipo de dato numérico, detalla el factor de expansión utilizado para ajustar los resultados.
- **upm:** Dato numérico, detalla la Unidad primaria de muestreo.
- **id_vivienda:** Dato de tipo texto o numérico, es el identificador único de la vivienda.
- **id_hogar:** Dato de tipo texto o numérico, es el identificador único del hogar.
- **periodo:** Dato de texto o numérico, detalla el periodo de tiempo en el que se realizó el estudio.
- **mes:** Dato de tipo texto o numérico que detalla el mes en el que se realizó el estudio.

Datos Complementarios

Los datos complementarios que se mencionarán se basan en el conjunto de datos de la Encuesta Nacional de Empleo, Desempleo y Subempleo (ENEMDU) del Instituto Nacional de Estadística y Censos (INEC) de Ecuador.

- Fuente: <https://www.ecuadorencifras.gob.ec/enemdu-trimestral/>
- Nombre del csv: BDDenemdu_personas_2022_anual.csv
- Link: https://www.ecuadorencifras.gob.ec/documentos/web-inec/EMPLEO/2023/Trimestre_I/2_BDD_DATOS_ABIERTOS_ENEMDU_2023_I_TRIMESTRE_CSV.zip
- **p02: Sexo** - Este es un dato numérico que representa el género del individuo, nos permite realizar análisis cuantitativos centrados en las diferencias de género. Como dato complementario, nos permite revelar patrones y tendencias de comportamiento específicos para cada género.
- **p03: Edad** - Este es un dato numérico que representa la edad del individuo, lo cual lo convierte en una variable clave para segmentar y analizar la población de estudio en base a grupos de edad. Como dato complementario, nos proporciona una dimensión de análisis que puede revelar patrones y tendencias de comportamiento específicas de cada grupo etario.
- **p06: Estado civil** - Este es un dato numérico que indica el estado civil de un individuo. Nos permite realizar análisis cuantitativos centrados en las diferencias de comportamiento y tendencias entre diferentes estados civiles. Como dato complementario, nos proporciona una perspectiva adicional para entender cómo el estado civil puede influir en otras dimensiones de la vida del individuo.

- **p10a:** Nivel de instrucción - Este es un dato numérico que representa el nivel de educación alcanzado por un individuo. Es una variable crucial para comprender la composición educativa de nuestra población de estudio. Como dato complementario, nos ofrece una dimensión de análisis adicional para entender cómo el nivel educativo puede afectar e interactuar con otras variables del estudio.
- **p21:** Actividad hogar - Este es un dato numérico que refleja la actividad que el individuo realiza para contribuir en su hogar. Como dato complementario, nos permite comprender más profundamente cómo estas actividades en el hogar pueden estar correlacionadas con otras variables del estudio, lo que puede revelar tendencias y patrones de comportamiento específicos.

Posibles análisis para realizar

1. Evaluación del impacto del nivel de instrucción en el tipo de vivienda
 - Dato base: vi02 (Tipo de vivienda)
 - Dato complementario: p10a (Nivel de instrucción)
2. Análisis 2: Análisis de la correlación entre el estado civil y la vía de acceso a la vivienda
 - Dato base usado: vi01 (Vía de acceso a la vivienda)
 - Dato complementario usado: p06 (Estado civil)
3. Estudio de la relación entre la actividad del hogar y el material de construcción de la vivienda
 - Dato base usado: vi03a, vi04a, vi05a (Material de techo, piso, paredes)
 - Dato complementario usado: p21 (Actividad hogar)
4. Examinación de la relación entre el estado civil y la disposición a pagar arriendo
 - Dato base usado: vi141 (Valor que estarían dispuestos a pagar por un arriendo)
 - Dato complementario usado: p06 (Estado civil)
5. Exploración de la influencia de la edad en el tipo de servicio higiénico

- Dato base usado: vi09 (Tipo de servicio higiénico)
 - Dato complementario usado: p03 (Edad)
6. Evaluación de la relación entre la edad y la forma de eliminación de basura
- Dato base usado: vi13 (Forma de eliminación de basura)
 - Dato complementario usado: p03 (Edad)
7. Análisis de la correlación entre la actividad del hogar y la fuente de agua
- Dato base usado: vi10 (Fuente de agua)
 - Dato complementario usado: p21 (Actividad hogar)
8. Análisis de la correlación entre el nivel de instrucción y la forma de eliminación de la basura
- Dato base usado: vi13 (Forma de eliminación de la basura)
 - Dato complementario usado: p10a (Nivel de instrucción)
9. Investigación de la influencia del nivel de instrucción en la tenencia de la vivienda
- Dato base usado: vi14 (Tenencia de la vivienda)
 - Dato complementario usado: p10a (Nivel de instrucción)
10. Análisis 10: Examinación de la correlación entre el género y el número de cuartos dedicados solo para dormir
- Dato base usado: vi07 (Número de cuartos para dormir)
 - Dato complementario usado: p02 (Sexo)
11. Análisis 11: Evaluación de la relación entre la actividad del hogar y la presencia de vehículos
- Dato base usado: vi1511 (Presencia de vehículos)
 - Dato complementario usado: p21 (Actividad hogar)
12. Análisis de la influencia de la edad en el tipo de combustible más frecuente
- Dato base usado: vi1531-vi1546 (Combustible más frecuente y gasto mensual)
 - Dato complementario usado: p03 (Edad)
13. Estudio de la correlación entre el nivel de instrucción y el número de cuartos para negocios
- Dato base usado: vi07a (Número de cuartos para negocios)
 - Dato complementario usado: p10a (Nivel de instrucción)

14. Análisis 14: Exploración de la influencia del género en el tipo de acceso a la vivienda

- Dato base usado: vi01 (Vía de acceso a la vivienda)
- Dato complementario usado: p02 (Sexo)

15. Investigación de la relación entre el género y la disposición a pagar arriendo

- Dato base usado: vi141 (Valor que estarían dispuestos a pagar por un arriendo)
- Dato complementario usado: p02 (Sexo)

Diseño Conceptual

El modelo conceptual se encuentra en el repositorio de GitHub con el siguiente link:

https://github.com/CarlosSalas8/Proyecto_Integrador.git, en el directorio "Base de datos avanzado" en la carpeta "Modelo Conceptual", en el archivo llamado "ModeloConceptual.drawio.png". Este modelo fue diseñado utilizando la herramienta draw.io, la cual es ampliamente utilizada para la creación de diagramas. Además, el modelo está disponible en draw.io en la siguiente dirección:

https://drive.google.com/file/d/1i4_uEHOkiqX4YAabrTxUxp0lJBqV5ulE/view?usp=sharing

Diseño Lógico Relacional

El Modelo Lógico Relacional se encuentra alojado en el repositorio de GitHub en el siguiente enlace: https://github.com/CarlosSalas8/Proyecto_Integrador.git específicamente, se puede acceder al modelo, en el directorio "Base de datos avanzado" dentro de la carpeta 'Modelo Lógico Relacional', en el archivo llamado "ModeloLogicoRelacional.drawio.png". Este modelo igual que el conceptual fue creado en draw.io por lo cual se puede visualizar a través de este link:

https://drive.google.com/file/d/14ZFaxAIml1Cn6Xslxc2nDzG2fd6vb_984/view?usp=sharing

Diseño Físico

El modelo físico igualmente se encuentra alojado en el repositorio de GitHub:

https://github.com/CarlosSalas8/Proyecto_Integrador.git en el directorio "Base de datos avanzado" en la carpeta "Modelo Físico" donde se encuentra un archivo llamado "MODELO.mdw" que se realizó utilizando la herramienta MySQL Workbench, la cual es reconocida por su eficacia en el diseño y la manipulación de bases de datos, aunque también se podrá visualizar fácilmente en una imagen llamada "Modelo Fisico.png"

Implementación y Carga

Evidencias de la implementación y carga en MySQL

- Creación Base De Datos: mydb

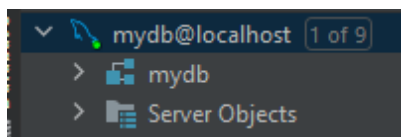
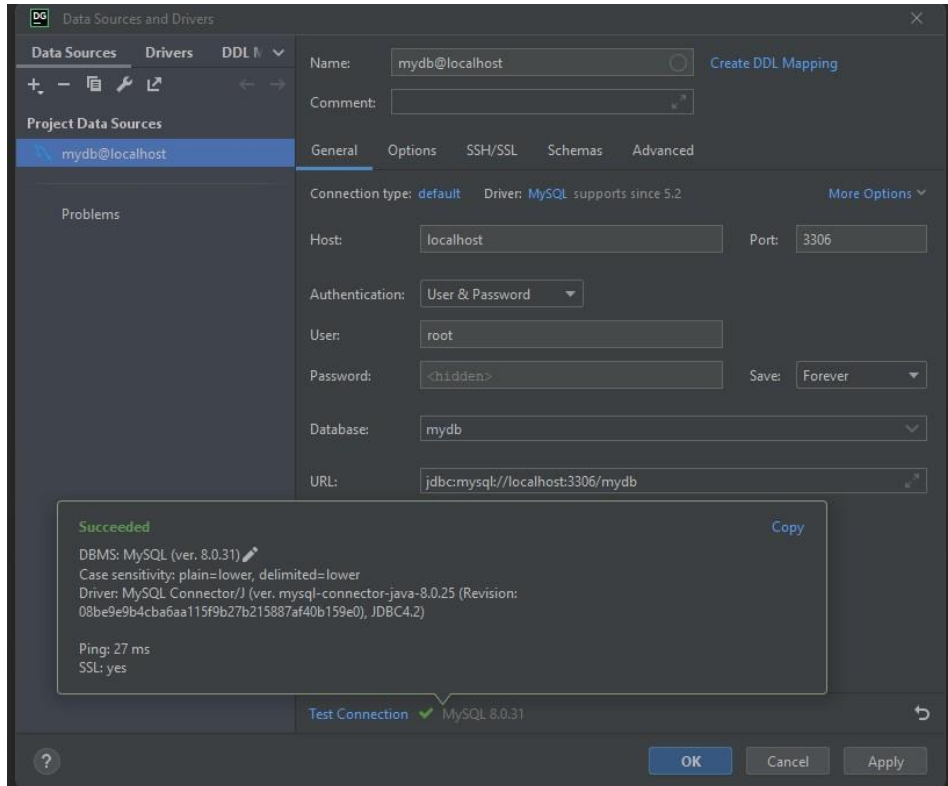
```
-- MySQL Script generated by MySQL Workbench
-- Sun Jul 2 19:00:05 2023
-- Model: New Model    Version: 1.0
-- MySQL Workbench Forward Engineering

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';

-- Schema mydb
-- -----
DROP SCHEMA IF EXISTS `mydb` ;

-- Schema mydb
-- -----
CREATE SCHEMA IF NOT EXISTS `mydb` DEFAULT CHARACTER SET utf8 ;
USE `mydb` ;
```

- Conexión MySQL



- Import Data From: enemdu_vivienda_hogar_2023_i_trimestre.

Formats:

+

-

📄

CSV

TSV

Pipe-separated

TSV, 1*

Value separator:

Semicolon

Row separator:

Newline

Null value text:

Empty string

Add row prefix/suffix

+

-

↕

Quotation:

" " Escape: duplicate

' ' Escape: duplicate

Quote values:

When needed

☐ Trim whitespaces

☒ First row is header

☐ First column is header

Header Format

Value separator:

Semicolon

Row separator:

Newline

Null value text:

Empty string

Add row prefix/suffix

Quotation:

" " Escape: duplicate

' ' Escape: duplicate

Quote values:

When needed

☐ Trim whitespaces

Target schema:

mydb@localhost / mydb

Table:

enemdu_vivienda_hogar_2023_i_trimestre Existing

Comment:

Columns (57)

Keys

Indexes

Foreign Keys

+

-

↕

area	int	mapped to area
ciudad	int	mapped to ciudad
conglomerado	int	mapped to conglomerado
panelm	int	mapped to panelm
vivienda	int	mapped to vivienda
hogar	int	mapped to hogar
vi01	int	mapped to vi01
vi02	int	mapped to vi02
vi03a	int	mapped to vi03a
vi03b	int	mapped to vi03b
vi04a	int	mapped to vi04a
vi04b	int	mapped to vi04b

Data Preview

DDL Preview

	area :	ciudad :	conglomerado :	panelm :	vivienda :	hogar :	vi01 :
1	1	10150	000201	026	01	1	1
2	1	10150	000201	026	02	1	1
3	1	10150	000201	026	03	1	1
4	1	10150	000201	026	04	1	1
5	1	10150	000201	026	05	1	1
6	1	10150	000201	026	06	1	1
7	1	10150	000201	026	09	1	1
8	1	10150	000304	042	01	1	3
9	1	10150	000304	042	02	1	1
10	1	10150	000304	042	03	1	1
11	1	10150	000304	042	04	1	3
12	1	10150	000304	042	05	1	3
13	1	10150	000304	042	06	1	1
14	1	10150	000304	042	07	1	3

Encoding:

UTF-8

☒ Write errors to file: C:\Users\sala\enemdu_vivienda_hogar_2023_i_trimestre_2023-07-02_21_44_05.txt

☐ Insert inconvertible values as null

☐ Disable indexes and triggers, lock table (may be faster)

Import

Cancel

- Import Data From: enemdu_persona_2023_i_trimestre.

Import "enemdu_persona_2023_i_trimestre.csv" File

Formats: CSV, TSV, Pipe-separated, **TSV_1***

Value separator: Semicolon
Row separator: Newline
Null value text: Empty string

Add row prefix/suffix
Quotation: + - ^ _
" " Escape: duplicate
' ' Escape: duplicate

Quote values: When needed

☐ Trim whitespaces
☒ First row is header
☐ First column is header

Header Format

Value separator: Semicolon
Row separator: Newline
Null value text: Empty string

Add row prefix/suffix
Quotation: + - ^ _
" " Escape: duplicate
' ' Escape: duplicate

Quote values: When needed

☐ Trim whitespaces

Target schema: mydb@localhost / mydb

Table: enemdu_persona_2023_i_trimestre Existing

Comment:

Columns (143) Keys Indexes Foreign Keys

Column	Type	Mapping
conduct	int	mapped to conduct
empleo	text	mapped to empleo
desempleo	text	mapped to desempleo
secemp	text	mapped to secemp
grupo1	text	mapped to grupo1
rama1	text	mapped to rama1
upa	double	mapped to upa
id_vivienda	varchar(100)	mapped to id_vivienda
id_hogar	varchar(100)	mapped to id_hogar
id_persona	varchar(100)	mapped to id_persona
periodo	int	mapped to periodo
mes	int	mapped to mes

Data Preview

	area :	ciudad :	conglomerado :	panelm :	vivienda :	hogar :	p01 :
1	1	10150	000201	026	01	1	01
2	1	10150	000201	026	02	1	01
3	1	10150	000201	026	02	1	02
4	1	10150	000201	026	03	1	01
5	1	10150	000201	026	04	1	01
6	1	10150	000201	026	04	1	02
7	1	10150	000201	026	05	1	01
8	1	10150	000201	026	05	1	02
9	1	10150	000201	026	05	1	03
10	1	10150	000201	026	05	1	04
11	1	10150	000201	026	06	1	01
12	1	10150	000201	026	06	1	02
13	1	10150	000201	026	09	1	01
14	1	10150	000201	026	09	1	02

Encoding: UTF-8

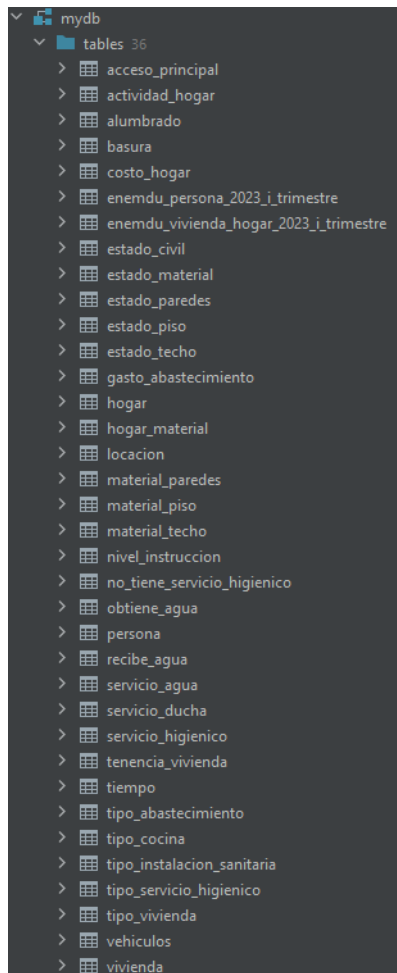
☒ Write errors to file: C:\Users\salas\enemdu_persona_2023_i_trimestre_2023-07-02_21_41_54.txt

☐ Insert inconvertible values as null

☐ Disable indexes and triggers, lock table (may be faster)

Import Cancel

- mydb



- Creación de Tablas en mydb

✓	2032	19:35:51	CREATE TABLE IF NOT EXISTS 'mydb'.ACCESO_PRINCIPAL' ('id_accesoPrincipal' INT NOT NULL, 'descripcion_accesoPrincipal' VARCH...
✓	2033	19:35:51	CREATE TABLE IF NOT EXISTS 'mydb'.TENENCIA_VIVIENDA' ('id_tenciaVivienda' INT NOT NULL, 'descripcion_tenenciaVivienda' VARC...
✓	2034	19:35:51	CREATE TABLE IF NOT EXISTS 'mydb'.ALUMBRADO' ('id_alumbrado' INT NOT NULL, 'descripcion_alumbrado' VARCHAR(200) NULL, P...
✓	2035	19:35:51	CREATE TABLE IF NOT EXISTS 'mydb'.TIPO_VIVIENDA' ('id_tipoVivienda' INT NOT NULL, 'descripcion_tipoVivienda' VARCHAR(200) NU...
✓	2036	19:35:51	CREATE TABLE IF NOT EXISTS 'mydb'.VIVIENDA' ('id_vivienda' VARCHAR(100) NOT NULL, 'ACCESO_PRINCIPAL_id_accesoPrincipal' I...
✓	2037	19:35:51	CREATE TABLE IF NOT EXISTS 'mydb'.TIEMPO' ('id_tiempo' INT NOT NULL AUTO_INCREMENT, 'periodo' VARCHAR(200) NULL, 'mes...
✓	2038	19:35:51	CREATE TABLE IF NOT EXISTS 'mydb'.LOCACION' ('id_locacion' INT NOT NULL AUTO_INCREMENT, 'factor_expansion' VARCHAR(200...
✓	2039	19:35:51	CREATE TABLE IF NOT EXISTS 'mydb'.TIPO_INSTALACION_SANITARIA' ('id_tipo_instalacionSanitaria' INT NOT NULL, 'descripcion_tipo...
✓	2040	19:35:51	CREATE TABLE IF NOT EXISTS 'mydb'.NO_TIENE_SERVICIO_HIGIENICO' ('id_noTiene_servicioHigienico' INT NOT NULL, 'descripcion_...
✓	2041	19:35:51	CREATE TABLE IF NOT EXISTS 'mydb'.TIPO_SERVICIO_HIGIENICO' ('id_tipo_servicioHigienico' INT NOT NULL, 'descripcion_tipo_servici...
✓	2042	19:35:51	CREATE TABLE IF NOT EXISTS 'mydb'.RECIBE_AGUA' ('id_recibeAgua' INT NOT NULL, 'descripcion_recibeAgua' VARCHAR(200) NULL...
✓	2043	19:35:51	CREATE TABLE IF NOT EXISTS 'mydb'.SERVICIO_DUCHA' ('id_servicioDucha' INT NOT NULL, 'descripcion_servicioDucha' VARCHAR(2...
✓	2044	19:35:51	CREATE TABLE IF NOT EXISTS 'mydb'.OBTIENE_AGUA' ('id_obtieneAgua' INT NOT NULL, 'descripcion_obtieneAgua' VARCHAR(200) N...
✓	2045	19:35:51	CREATE TABLE IF NOT EXISTS 'mydb'.SERVICIO_AGUA' ('id_servicioAgua' INT NOT NULL AUTO_INCREMENT, 'medidor_agua' VARC...
✓	2046	19:35:51	CREATE TABLE IF NOT EXISTS 'mydb'.SERVICIO_HIGIENICO' ('id_servicioHigiencio' INT NOT NULL AUTO_INCREMENT, 'TIPO_INSTA...
✓	2047	19:35:51	CREATE TABLE IF NOT EXISTS 'mydb'.TIPO_COCINA' ('id_tipoCocina' INT NOT NULL, 'descripcion_tipoCocina' VARCHAR(200) NULL, ...
✓	2048	19:35:51	CREATE TABLE IF NOT EXISTS 'mydb'.BASURA' ('id_basura' INT NOT NULL, 'descripcion_basura' VARCHAR(200) NULL, PRIMARY K...
✓	2049	19:35:51	CREATE TABLE IF NOT EXISTS 'mydb'.MATERIAL_PISO' ('id_materialPiso' INT NOT NULL, 'descripcion_materialPiso' VARCHAR(200) NU...
✓	2050	19:35:51	CREATE TABLE IF NOT EXISTS 'mydb'.MATERIAL_PAREDES' ('id_materialParedes' INT NOT NULL, 'descripcion_materialParedes' VARC...
✓	2051	19:35:51	CREATE TABLE IF NOT EXISTS 'mydb'.MATERIAL_TECHO' ('id_materialTecho' INT NOT NULL, 'descripcion_materialTecho' VARCHAR(2...
✓	2052	19:35:51	CREATE TABLE IF NOT EXISTS 'mydb'.HOGAR_MATERIAL' ('id_hogarMaterial' INT NOT NULL AUTO_INCREMENT, 'MATERIAL_PISO_...
✓	2053	19:35:51	CREATE TABLE IF NOT EXISTS 'mydb'.COSTO_HOGAR' ('id_costoHogar' INT NOT NULL AUTO_INCREMENT, 'valor_mensualAlriendo' ...
✓	2054	19:35:51	CREATE TABLE IF NOT EXISTS 'mydb'.GASTO_ABASTECIMIENTO' ('id_gastoAbastecimiento' INT NOT NULL AUTO_INCREMENT, 'gast...
✓	2055	19:35:51	CREATE TABLE IF NOT EXISTS 'mydb'.TIPO_ABASTECIMIENTO' ('id_tipoAbastecimiento' INT NOT NULL AUTO_INCREMENT, 'abasteci...
✓	2056	19:35:51	CREATE TABLE IF NOT EXISTS 'mydb'.VEHICULOS' ('id_Vehiculos' INT NOT NULL AUTO_INCREMENT, 'tiene_vehiculo' VARCHAR(100...
✓	2057	19:35:51	CREATE TABLE IF NOT EXISTS 'mydb'.ESTADO_PISO' ('id_estadoPiso' INT NOT NULL, 'descripcion_estadoPiso' VARCHAR(200) NULL, ...
✓	2058	19:35:51	CREATE TABLE IF NOT EXISTS 'mydb'.ESTADO_TECHE' ('id_estadoTecho' INT NOT NULL, 'descripcion_estadoTecho' VARCHAR(200) ...
✓	2059	19:35:51	CREATE TABLE IF NOT EXISTS 'mydb'.ESTADO_PAREDES' ('id_estadoParedes' INT NOT NULL, 'descripcion_estadosParedes' VARCHA...
✓	2060	19:35:51	CREATE TABLE IF NOT EXISTS 'mydb'.ESTADO_MATERIAL' ('id_estado_material' INT NOT NULL AUTO_INCREMENT, 'ESTADO_PISO_...
✓	2061	19:35:51	CREATE TABLE IF NOT EXISTS 'mydb'.HOGAR' ('id_hogar' VARCHAR(100) NOT NULL, 'cantidad_cuartos' VARCHAR(200) NULL, 'cuart...
✓	2062	19:35:51	CREATE TABLE IF NOT EXISTS 'mydb'.ACTIVIDAD_HOGAR' ('id_actividad_hogar' INT NOT NULL, 'descripcion_actividadHogar' VARCH...
✓	2063	19:35:51	CREATE TABLE IF NOT EXISTS 'mydb'.ESTADO_CIVIL' ('id_estado_civil' INT NOT NULL, 'descripcion_estadoCivil' VARCHAR(200) NULL...
✓	2064	19:35:51	CREATE TABLE IF NOT EXISTS 'mydb'.NIVEL_INSTRUCCION' ('id_nivel_instruccion' INT NOT NULL, 'descripcion_nivelInstruccion' VAR...
✓	2065	19:35:51	CREATE TABLE IF NOT EXISTS 'mydb'.Persona' ('id_Persona' VARCHAR(100) NOT NULL, 'sexo' VARCHAR(45) NULL, 'edad' VARCHA...
✓	2066	19:35:51	SET SQL_MODE=@OLD_SQL_MODE
✓	2067	19:35:51	SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS

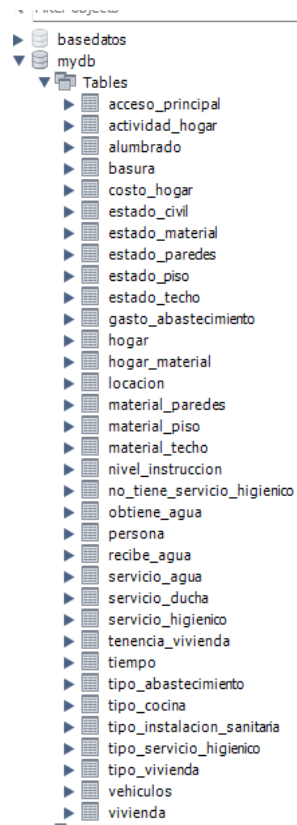
- INSERT Tablas Catálogos

✓	2069	19:42:05	INSERT INTO acceso_principal values(1,"Carretera, calle pavimentada o adoquinada")
✓	2070	19:42:05	INSERT INTO acceso_principal values(2,"Empedrado")
✓	2071	19:42:05	INSERT INTO acceso_principal values(3,"Lastrado, calle de tierra")
✓	2072	19:42:05	INSERT INTO acceso_principal values(4,"Sendero")
✓	2073	19:42:05	INSERT INTO acceso_principal values(5,"Río, Mar")
✓	2074	19:42:05	INSERT INTO acceso_principal values(6,"Otro")
✓	2075	19:42:05	Insert into tenencia_vivienda values (1,"En amiendo ")
✓	2076	19:42:05	Insert into tenencia_vivienda values (2,"Anticresis y/o amiendo")
✓	2077	19:42:05	Insert into tenencia_vivienda values (3,"Propia y la está pagando")
✓	2078	19:42:05	Insert into tenencia_vivienda values (4,"Propia y totalmente pagada")
✓	2079	19:42:05	Insert into tenencia_vivienda values (5,"Cedida")
✓	2080	19:42:05	Insert into tenencia_vivienda values (6,"Recibida por servicios")
✓	2081	19:42:05	Insert into tenencia_vivienda values (7,"Otra")
✓	2082	19:42:05	Insert into alumbrado values (1,"Empresa eléctrica pública")
✓	2083	19:42:05	Insert into alumbrado values (2,"Planta eléctrica privada")
✓	2084	19:42:05	Insert into alumbrado values (3,"Vela, candil, mechero, gas")
✓	2085	19:42:05	Insert into alumbrado values (4,"Ninguno")
✓	2086	19:42:05	insert into tipo_vivienda values (1,"Casa o villa ")
✓	2087	19:42:05	Insert into tipo_vivienda values (2,"Departamento")
✓	2088	19:42:05	Insert into tipo_vivienda values (3,"Cuartos en casa de inquilinato")
✓	2089	19:42:05	Insert into tipo_vivienda values (4,"Mediagua")
✓	2090	19:42:05	Insert into tipo_vivienda values (5,"Rancho, covacha")
✓	2091	19:42:05	Insert into tipo_vivienda values (6,"Choza")
✓	2092	19:42:05	Insert into tipo_vivienda values (7,"Otra")
✓	2093	19:42:05	Insert into estado_paredes values (1,"Bueno")
✓	2094	19:42:05	Insert into estado_paredes values (2,"Regular")
✓	2095	19:42:05	Insert into estado_paredes values (3,"Malo")
✓	2096	19:42:05	Insert into estado_piso values (1,"Bueno")
✓	2097	19:42:05	Insert into estado_piso values (2,"Regular")
✓	2098	19:42:05	Insert into estado_piso values (3,"Malo")
✓	2099	19:42:05	Insert into estado_techo values (1,"Bueno")
✓	2100	19:42:05	Insert into estado_techo values (2,"Regular")
✓	2101	19:42:05	Insert into estado_techo values (3,"Malo")
✓	2102	19:42:05	insert into material_techo values (1,"Hormigón (losa, cemento)")
✓	2103	19:42:05	insert into material_techo values (2,"Fibrocemento,asbesto (etemit, eurolit) ")
✓	2104	19:42:05	insert into material_techo values (3,"Zinc, Aluminio")

✓	2104	19:42:05	insert into material_techo values (3,"Zinc, Aluminio")
✓	2105	19:42:05	insert into material_techo values (4,"Teja")
✓	2106	19:42:05	insert into material_techo values (5,"Palma, paja u hoja")
✓	2107	19:42:05	insert into material_techo values (6,"Otro Material")
✓	2108	19:42:05	insert into material_piso values (1,"Duela, parquet, tablón tratado o piso flotante")
✓	2109	19:42:05	insert into material_piso values (2,"Cerámica, baldosa, vinil o porcelanato")
✓	2110	19:42:05	insert into material_piso values (3,"Mármol o marmetón")
✓	2111	19:42:05	insert into material_piso values (4,"Ladrillo o cemento")
✓	2112	19:42:05	insert into material_piso values (5,"Tabla / tablón no tratado ")
✓	2113	19:42:05	insert into material_piso values (6,"Caña")
✓	2114	19:42:05	insert into material_piso values (7,"Tierra")
✓	2115	19:42:05	insert into material_piso values (8,"Otro Material ")
✓	2116	19:42:05	insert into material_paredes values (1,"Hormigón/Ladrillo o Bloque")
✓	2117	19:42:05	insert into material_paredes values (2,"Asbesto/Cemento (Fibrolit) ")
✓	2118	19:42:05	insert into material_paredes values (3,"Adobe o Tapia")
✓	2119	19:42:05	insert into material_paredes values (4,"Madera")
✓	2120	19:42:05	insert into material_paredes values (5,"Caña revestida o bahareque ")
✓	2121	19:42:05	insert into material_paredes values (6,"Caña no revestida o estera")
✓	2122	19:42:05	insert into material_paredes values (7,"Otra Material")
✓	2123	19:42:05	INSERT INTO tipo_servicio_higienico VALUES (1,'Excusado y alcantarillado')
✓	2124	19:42:05	INSERT INTO tipo_servicio_higienico VALUES (2,'Excusado y pozo séptico')
✓	2125	19:42:05	INSERT INTO tipo_servicio_higienico VALUES (3,'Excusado y pozo ciego')
✓	2126	19:42:05	INSERT INTO tipo_servicio_higienico VALUES (4,'Letrina')
✓	2127	19:42:05	INSERT INTO tipo_servicio_higienico VALUES (5,'No tiene')
✓	2128	19:42:05	INSERT INTO no_tiene_servicio_higienico VALUES (1,'Descarga directa al mar, río, lago o quebrada')
✓	2129	19:42:05	INSERT INTO no_tiene_servicio_higienico VALUES (2,'Van al monte, campo, bota la basura en paquete')
✓	2130	19:42:05	INSERT INTO no_tiene_servicio_higienico VALUES (3,'Usan una instalación sanitaria cercana y/o prestada')
✓	2131	19:42:05	INSERT INTO no_tiene_servicio_higienico VALUES (4,'No respondo')
✓	2132	19:42:05	INSERT INTO tipo_instalacion_sanitaria VALUES (1,'Excusado y alcantarillado')
✓	2133	19:42:05	INSERT INTO tipo_instalacion_sanitaria VALUES (2,'Excusado y pozo séptico')
✓	2134	19:42:05	INSERT INTO tipo_instalacion_sanitaria VALUES (3,'Excusado y pozo ciego')
✓	2135	19:42:05	INSERT INTO tipo_instalacion_sanitaria VALUES (4,'Letrina')
✓	2136	19:42:05	INSERT INTO tipo_instalacion_sanitaria VALUES (5,'No respondo')
✓	2137	19:42:05	INSERT INTO basura VALUES (1,'Contratan el servicio')
✓	2138	19:42:05	INSERT INTO basura VALUES (2,'Servicio municipal')
✓	2139	19:42:05	INSERT INTO basura VALUES (3,'Botan a la calle, quebrada, río')

2153	19:42:05	INSERT INTO servicio_ducha VALUES (1,'Exclusivo del hogar')
2154	19:42:05	INSERT INTO servicio_ducha VALUES (2,'Compartido con otros hogares')
2155	19:42:05	INSERT INTO servicio_ducha VALUES (3,'No tiene')
2156	19:42:05	INSERT INTO tipo_cocina VALUES (1,'Gas')
2157	19:42:05	INSERT INTO tipo_cocina VALUES (2,'Leña, carbón')
2158	19:42:05	INSERT INTO tipo_cocina VALUES (3,'Electricidad')
2159	19:42:05	INSERT INTO tipo_cocina VALUES (4,'Otro')
2160	19:42:05	INSERT INTO estado_civil VALUES (1,'Casado(a)')
2161	19:42:05	INSERT INTO estado_civil VALUES (2,'Separado(a)')
2162	19:42:05	INSERT INTO estado_civil VALUES (3,'Divorciado(a)')
2163	19:42:05	INSERT INTO estado_civil VALUES (4,'Vudo(a)')
2164	19:42:05	INSERT INTO estado_civil VALUES (5,'Unión libre')
2165	19:42:05	INSERT INTO estado_civil VALUES (6,'Soltero(a)')
2166	19:42:05	INSERT INTO nivel_instruccion VALUES (1,'Ninguno')
2167	19:42:05	INSERT INTO nivel_instruccion VALUES (2,'Centro de alfabetización')
2168	19:42:05	INSERT INTO nivel_instruccion VALUES (3,'Jardín de infantes')
2169	19:42:05	INSERT INTO nivel_instruccion VALUES (4,'Primaria')
2170	19:42:05	INSERT INTO nivel_instruccion VALUES (5,'Educación Básica')
2171	19:42:05	INSERT INTO nivel_instruccion VALUES (6,'Secundaria')
2172	19:42:05	INSERT INTO nivel_instruccion VALUES (7,'Educación Media')
2173	19:42:05	INSERT INTO nivel_instruccion VALUES (8,'Superior no universitario')
2174	19:42:05	INSERT INTO nivel_instruccion VALUES (9,'Superior Universitario')
2175	19:42:05	INSERT INTO nivel_instruccion VALUES (10,'Post-grado')
2176	19:42:05	INSERT INTO actividad_hogar VALUES (1,'Atender negocio propio')
2177	19:42:05	INSERT INTO actividad_hogar VALUES (2,'Fabricar algún producto')
2178	19:42:05	INSERT INTO actividad_hogar VALUES (3,'Hacer algo en casa por un ingreso')
2179	19:42:05	INSERT INTO actividad_hogar VALUES (4,'Brindar algún servicio')
2180	19:42:05	INSERT INTO actividad_hogar VALUES (5,'Ayudar en algún negocio familiar')
2181	19:42:05	INSERT INTO actividad_hogar VALUES (6,'Ayudar en el trabajo de algún familiar')
2182	19:42:05	INSERT INTO actividad_hogar VALUES (7,'Aprender remunerado')
2183	19:42:05	INSERT INTO actividad_hogar VALUES (8,'Labores agrícolas, cuidado animales')
2184	19:42:05	INSERT INTO actividad_hogar VALUES (9,'Estudiante que realiza algún trabajo')
2185	19:42:05	INSERT INTO actividad_hogar VALUES (10,'Trabajar para otra familia')
2186	19:42:05	INSERT INTO actividad_hogar VALUES (11,'Otra actividad por un ingreso')
2187	19:42:05	INSERT INTO actividad_hogar VALUES (12,'No realiza ninguna actividad')
2188	19:58:07	SELECT * FROM mydb.hogar LIMIT 0, 1000

Evidencia



INSERTAR DATOS EN LAS TABLAS

- HOGAR

```
def insert_Hogar(id_hogar, cantCuartos, cuartoDormitorios, cuartoNegocios, cuartoCocina, tipoCocina, basura, materialPiso, materialParedes,
                materialTecho, valorMensualArriendo, incluyeServicioAgua, incluyeServicioLuz, poseeVehiculos, numVehiculos, poseeMotos, numMotos,
                abastecimientoSuper, abastecimientoDiesel, abastecimientoEco, abastecimientoExtra, abastecimientoElectricidad,
                abastecimientoGas, gastoSuper, gastoDiesel, gastoEco, gastoExtra, gastoElectricidad, gastoGas,
                medidorAgua, obtieneJuntaAgua, medioRecepcionAgua, servicioDucha, obtencionAgua, tipoInstSanitaria, alternativaNoSH,
                tipoServicioHigienico, parentescoConPropietario, estadoPiso, estadoTecho, estadoParedes):

    cursor = db.cursor()

    # Insertar datos en la tabla Hogar_material
    query_material = "INSERT INTO hogar_material (MATERIAL_PISO_id_materialPiso, MATERIAL_PAREDES_id_materialParedes, MATERIAL_Techo_id_materialTecho) VALUES (%s, %s, %s)"
    values_material = (materialPiso, materialParedes, materialTecho)
    cursor.execute(query_material, values_material)
    # Obtener el valor de hogar_material generado automáticamente
    id_HogarMaterial = cursor.lastrowid

    # Insertar datos en la tabla estado_material
    query_estado = "INSERT INTO estado_material (ESTADO_PISO_id_estadoPiso, ESTADO_Techo_id_estadoTecho, ESTADO_PAREDES_id_estadoParedes) VALUES (%s, %s, %s)"
    values_estado = (estadoPiso, estadoTecho, estadoParedes)
    cursor.execute(query_estado, values_estado)
    # Obtener el valor de estado_material generado automáticamente
    id_estadoMaterial = cursor.lastrowid

    # Insertar datos en la tabla costo_hogar
    query = "INSERT INTO costo_hogar (valor_mensualArriendo, valor_incluyeAgua, valor_incluyeLuz) VALUES (%s, %s, %s)"
    values = (_valorMensualArriendo, incluyeServicioAgua, incluyeServicioLuz)
    cursor.execute(query, values)
    db.commit()
    # Obtener el valor de costo_hogar generado automáticamente
    id_costoHogar = cursor.lastrowid

    # Insertar datos en la tabla GastoAbastecimiento
    query_gastoAbast = "INSERT INTO gasto_abastecimiento (gasto_super, gasto_diesel, gasto_ecopais, gasto_extra, gasto_electricidad, gasto_gas) VALUES (%s, %s, %s, %s, %s, %s)"
    values_gastoAbast = (gastoSuper, gastoDiesel, gastoEco, gastoExtra, gastoElectricidad, gastoGas)
    cursor.execute(query_gastoAbast, values_gastoAbast)
```

```
# Insertar datos en la tabla GastoAbastecimiento
query_gastoAbast = "INSERT INTO gasto_abastecimiento (gasto_super, gasto_diesel, gasto_ecopais, gasto_extra, gasto_electricidad, gasto_gas) VALUES (%s, %s, %s, %s, %s, %s)"
values_gastoAbast = (gastoSuper, gastoDiesel, gastoEco, gastoExtra, gastoElectricidad, gastoGas)
cursor.execute(query_gastoAbast, values_gastoAbast)

# Obtener el valor de GastoAbastecimiento generado automáticamente
id_gastoAbastecimiento = cursor.lastrowid

# Insertar datos en la tabla Abastecimiento
query_abastecimiento = "INSERT INTO tipo_abastecimiento (abastecimiento_super, abastecimiento_diesel, abastecimiento_ecopais, abastecimiento_extra, abastecimiento_electricidad, " \
    "abastecimiento_gas, GASTO_ABASTECIMIENTO_id_gastoAbastecimiento) VALUES (%s, %s, %s, %s, %s, %s, %s)"
values_abastecimiento = (
    abastecimientoSuper, abastecimientoDiesel, abastecimientoEco, abastecimientoExtra, abastecimientoElectricidad,
    abastecimientoGas, id_gastoAbastecimiento)
cursor.execute(query_abastecimiento, values_abastecimiento)
# Obtener el valor de id_Abastecimiento generado automáticamente
id_Abastecimiento = cursor.lastrowid

# Insertar datos en la tabla VEHICULOS
query_vehiculos = "INSERT INTO vehiculos (tiene_vehiculo, num_vehiculo, tiene_moto, num_moto, TIPO_ABASTECIMIENTO_id_tipoAbastecimiento) VALUES (%s, %s, %s, %s, %s)"
values_vehiculos = (poseeVehiculos, numVehiculos, poseeMotos, numMotos, id_Abastecimiento)
cursor.execute(query_vehiculos, values_vehiculos)
# Obtener el valor de VEHICULOS generado automáticamente
id_vehiculos = cursor.lastrowid

# Insertar datos en la tabla ServicioAgua
query_servicioAgua = "INSERT INTO servicio_agua (medidor_agua, junta_agua, RECIBE_AGUA_id_recibeAgua, SERVICIO_DUCHA_id_servicioDucha, OBTIENE_AGUA_id_obtieneAgua) VALUES (%s, %s, %s, %s, %s)"
values_servicioAgua = (medidorAgua, obtieneJuntaAgua, medioRecepcionAgua, servicioDucha, obtencionAgua)
cursor.execute(query_servicioAgua, values_servicioAgua)

# Obtener el valor de id_ServicioAgua generado automáticamente
id_ServicioAgua = cursor.lastrowid

query_servicioHigienico = "INSERT INTO servicio_higienico (TIPO_INSTALACION_SANITARIA_id_tipoInstalacionSanitaria, NO_TIENE_SERVICIO_HIGIENICO_id_noTieneServicioHigienico, " \
    "TIPO_SERVICIO_HIGIENICO_id_tipoServicioHigienico, SERVICIO_AGUA_id_servicioAgua) VALUES (%s, %s, %s, %s)"
values_servicioHigienico = (tipoInstSanitaria, alternativaNoSH, tipoServicioHigienico, id_ServicioAgua)
cursor.execute(query_servicioHigienico, values_servicioHigienico)
# Obtener el valor de id_ServicioHigienico generado automáticamente
id_ServicioHigienico = cursor.lastrowid
```



```

# Obtener el valor de id_ServicioAgua generado automáticamente
id_ServicioAgua = cursor.lastrowid

query_servicioHigienico = "INSERT INTO servicio_higienico (TIPO_INSTALACION_SANITARIA_id_tipo_instalacionSanitaria, NO_TIENE_SERVICIO_HIGIENICO_id_noTiene_servicioHigienico, \
    TIPO_SERVICIO_HIGIENICO_id_tipo_servicioHigienico, SERVICIO_AGUA_id_servicioAgua) VALUES (%s, %s, %s, %s)"
values_servicioHigienico = (tipoInstSanitaria, alternativaMoSH, tipoServicioHigienico, id_ServicioAgua)
cursor.execute(query_servicioHigienico, values_servicioHigienico)
# Obtener el valor de id_ServicioHigienico generado automáticamente
id_ServicioHigienico = cursor.lastrowid

query_id_persona = "SELECT id_Vivienda FROM enemdu_vivienda_hogar_2023_1_trimestre where id_hogar = %s"
condicion = [id_hogar]
cursor.execute(query_id_persona, condicion)
results = cursor.fetchall()

query_hogar = "INSERT INTO hogar (id_hogar, cantidad_cuartos, cuartos_dormitorios, cuartos_negocios, cuarto_cocina, parentesco_propietario, TIPO_COCINA_id_tipoCocina, \
    SERVICIO_HIGIENICO_id_servicioHigienico, BASURA_id_basura, HOGAR_MATERIAL_id_hogarMaterial, ESTADO_MATERIAL_id_estado_material, COSTO_HOGAR_id_costoHogar, \
    VEHICULOS_id_Vehiculos, VIVIENDA_id_vivienda) VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s)"
values_hogar = [(id_hogar, cantCuartos, cuartoDormitorios, cuartoNegocios, cuartoCocina, parentescoConPropietario, tipoCocina, id_ServicioHigienico, basura, id_HogarMaterial, id_estadoMaterial, id_costoHogar, id_vehiculos, result[0]) for result in results]
cursor.executemany(query_hogar, values_hogar)

db.commit()

```

- VIVIENDA

```

def insertar_Vivienda(periodo, mes,
    viaAccesoPrincipal, tipoVivienda, tenenciaVivienda, tipoAlumbrado, factorExpansion,
    unidadPrimariaMuestreo, ciudad, conglomerado, estrato, area):
    cursor = db.cursor()

    # Insertar datos en la tabla Tiempo
    query_tiempo = "INSERT INTO tiempo (periodo, mes) VALUES (%s, %s)"
    values_tiempo = (periodo, mes)
    cursor.execute(query_tiempo, values_tiempo)
    id_tiempo = cursor.lastrowid

    # Verificar si la vivienda ya existe en la tabla Vivienda
    query_verificar_vivienda = "SELECT id_vivienda FROM vivienda WHERE id_vivienda = %s"
    values_verificar_vivienda = (id_vivienda,)
    cursor.execute(query_verificar_vivienda, values_verificar_vivienda)
    vivienda_existente = cursor.fetchone()

    if vivienda_existente:
        # La vivienda ya existe, se puede realizar alguna acción aquí, como mostrar un mensaje de error o manejarlo según tus necesidades
        print("La vivienda ya existe.")
    else:
        # Insertar datos en la tabla Vivienda
        query_vivienda = "INSERT INTO vivienda (id_vivienda, ACCESO_PRINCIPAL_id_accesoPrincipal, TIPO_VIVIENDA_id_tipoVivienda, \
            TENENCIA_VIVIENDA_id_tenciaVivienda, ALUMBRADO_id_alumbrado) VALUES (%s, %s, %s, %s, %s)"
        values_vivienda = (id_vivienda, viaAccesoPrincipal, tipoVivienda, tenenciaVivienda, tipoAlumbrado)
        cursor.execute(query_vivienda, values_vivienda)

        # Insertar datos en la tabla Locacion
        query_locacion = "INSERT INTO locacion (factor_expansion, unidad_muestreo, ciudad, conglomeracion, estrato, area, TIEMPO_id_tiempo, \
            VIVIENDA_id_vivienda) VALUES (%s, %s, %s, %s, %s, %s, %s, %s)"
        values_locacion = (factorExpansion, unidadPrimariaMuestreo, ciudad, conglomerado, estrato, area, id_tiempo, id_vivienda)
        cursor.execute(query_locacion, values_locacion)

        db.commit()

    cursor.close()

```

```
with open('C:/Users/salas/OneDrive/Escritorio/PRACTICUM1.2/enemdu.vivienda.hogar.2023_I_trimestre.csv', newline='', encoding='utf-8') as csvfile:
    spamreader = csv.reader(csvfile, delimiter=',')
    next(spamreader)
    for row in spamreader:
        area = (row[0])
        ciudad = (row[1])
        conglomerado = (row[2])
        panelm = (row[3])
        vivienda = (row[4])
        hogar = (row[5])
        viaAccesoPrincipal = (row[6])
        tipoVivienda = (row[7])
        materialTecho = (row[8])
        estadoTecho = (row[9])
        materialPiso = (row[10])
        estadoPiso = (row[11])
        materialParedes = (row[12])
        estadoParedes = (row[13])
        cantCuartos = (row[14])
        cuartoDormitorios = (row[15])
        cuartoNegocios = (row[16])
        cuartoCocina = (row[17])
        tipoCocina = (row[18])
        tipoServicioHigienico = (row[19])
        alternativaNoSH = (row[20]).replace(' ','4")
        tipoInstSanitaria = (row[21]).replace(' ','5")
        obtencionAgua = (row[22])
        medidorAgua = (row[23])
        obtieneJuntaAgua = (row[24])
        medioRecepcionAgua = (row[25])
        servicioDucha = (row[26])
        tipoAlumbrado = (row[27])
        basura = (row[28])
```

```

tiposervicioHigienico = (row[19])
alternativaNoSH = (row[20]).replace(' ','4")
tipoInstSanitaria = (row[21]).replace(' ','5")
obtencionAgua = (row[22])
medidorAgua = (row[23])
obtieneJuntaAgua = (row[24])
medioRecepcionAgua = (row[25])
servicioDucha = (row[26])
tipoAlumbrado = (row[27])
basura = (row[28])
tenenciaVivienda = (row[29])
valorMensualArriendo = (row[30])
incluyeServicioAgua = (row[31]).replace(' ','3")
incluyeServicioLuz = (row[32]).replace(' ','3")
parentescoConPropietario = (row[33]).replace(' ','2")
poseeVehiculos = (row[34])
numVehiculos = (row[35]).replace(' ','0")
poseeMotos = (row[36])
numMotos = (row[37]).replace(' ','0")
abastecimientoSuper = (row[38]).replace(' ','0")
gastoSuper = (row[39]).replace(' ','0")
abastecimientoExtra = (row[40]).replace(' ','0")
gastoExtra = (row[41]).replace(' ','0")
abastecimientoDiesel = (row[42]).replace(' ','0")
gastoDiesel = (row[43]).replace(' ','0")
abastecimientoEco = (row[44]).replace(' ','0")
gastoEco = (row[45]).replace(' ','0")
abastecimientoElectricidad = (row[46]).replace(' ','0")
gastoElectricidad = (row[47]).replace(' ','0")
abastecimientoGas = (row[48]).replace(' ','0")
gastoGas = (row[49]).replace(' ','0")
estrato = (row[50])
factorExpansion = (row[51])
unidadPrimariaMuestreo = (row[52])
id_vivienda = (row[53])
id_hogar = (row[54])
periodo = (row[55])
mes = (row[56])

```

```

insert_Hogar(id_hogar, cantCuartos, cuartoDormitorios, cuartoNegocios, cuartoCocina, tipoCocina, basura,
materialPiso, materialParedes,
materialTecho, valorMensualArriendo, incluyeServicioAgua, incluyeServicioLuz, poseeVehiculos,
numVehiculos, poseeMotos, numMotos, abastecimientoSuper, abastecimientoDiesel, abastecimientoEco,
abastecimientoExtra, abastecimientoElectricidad, abastecimientoGas, gastoSuper, gastoDiesel,
gastoEco, gastoExtra, gastoElectricidad, gastoGas,
medidorAgua, obtieneJuntaAgua, medioRecepcionAgua, servicioDucha, obtencionAgua, tipoInstSanitaria,
alternativaNoSH,
tipoServicioHigienico, parentescoConPropietario, estadoPiso, estadoTecho, estadoParedes)

insertar_Vivienda(periodo, mes,
viaAccesoPrincipal, tipoVivienda, tenenciaVivienda, tipoAlumbrado, factorExpansion,
unidadPrimariaMuestreo, ciudad, conglomerado, estrato, area)

```

- INSERTAR PERSONA

```
import mysql.connector as mysql
import csv

db = mysql.connect(
    host="127.0.0.1",
    database="mydb",
    user="root",
    passwd="password"
)

cursor = db.cursor
cursor = db.cursor(buffered=True)
with open('C:/Users/salas/OneDrive/Escritorio/PRACTICUM1.2/enemdu_persona_2023_I_trimestre.csv', newline='', encoding='utf-8') as csvfile:
    spamreader2 = csv.reader(csvfile, delimiter=',')
    next(spamreader2)
    for row in spamreader2:

        id_persona = row[140]
        sexo = row[7]
        edad = row[8]
        estadoCivil = row[12].replace(' ', "6")
        nivelInstruccion = row[18].replace(' ', "1")
        actividadHogar = row[28].replace(' ', "12")

        cursor = db.cursor()
        query_id_persona = "SELECT id_hogar FROM enemdu_persona_2023_I_trimestre where id_persona = %s"
        condicion = [id_persona]
        cursor.execute(query_id_persona, condicion)

        results = cursor.fetchall()
        values_persona = [(id_persona, sexo, edad, actividadHogar, estadoCivil, nivelInstruccion, result[0]) for
            result in results]

        query_persona = "INSERT INTO persona (id_Persona, sexo, edad, ACTIVIDAD_HOGAR_id_actividad_hogar, ESTADO_CIVIL_id_estado_civil, " \
            "NIVEL_INSTRUCCION_id_nivel_instruccion, HOGAR_id_hogar) VALUES (%s, %s, %s, %s, %s, %s, %s)"

        cursor.executemany(query_persona, values_persona)
        db.commit()

    db.commit()

db.close()
```

- Ejemplo Estado Material

Estado Paredes

	id_estadoParedes	descripcion_estadosParedes
▶	1	Bueno
	2	Regular
	3	Malo
✱	NULL	NULL

Estado Piso

	id_estadoPiso	descripcion_estadoPiso
▶	1	Bueno
	2	Regular
	3	Malo
⌵	NULL	NULL

Estado Techo

	id_estadoTecho	descripcion_estadoTecho
▶	1	Bueno
	2	Regular
	3	Malo

Estado Material

	id_estado_material	ESTADO_PISO_id_estadoPiso	ESTADO_Techo_id_estadoTecho	ESTADO_PAREDES_id_estadoParedes
▶	1	2	3	2
	2	2	3	2
	3	2	3	2
	4	1	1	1
	5	1	1	1
	6	1	3	1
	7	3	3	2
	8	2	2	2
	9	2	3	2
	10	1	1	1
	11	2	2	2
	12	1	1	1
	13	2	3	2
	14	1	1	1
	15	2	1	2
	16	2	1	2
	17	1	1	1
	18	1	1	1
	19	1	1	1
	20	1	1	1
	21	1	2	1
	22	2	2	2
	23	1	1	1
	24	2	2	2
	25	1	1	1
	26	2	2	2
	27	1	2	1
	28	3	3	3
	29	2	1	2

Persona

	id_Persona	sexo	edad	ACTIVIDAD_HOGAR_id_actividad_hogar	ESTADO_CIVIL_id_estado_civil	NIVEL_INSTRUCCION_id_nivel_instruccion	HOGAR_id_hogar
	0101500002010260110102	2	78	12	3	4	01015000020102601102
	0101500002010260210102	1	79	12	1	4	01015000020102602102
	0101500002010260210202	2	73	12	1	6	01015000020102602102
	0101500002010260310102	1	46	12	3	9	01015000020102603102
	0101500002010260410102	2	59	12	6	9	01015000020102604102
	0101500002010260410202	2	85	12	4	4	01015000020102604102
	0101500002010260510102	2	77	12	6	6	01015000020102605102
	0101500002010260510202	1	74	12	3	9	01015000020102605102
	0101500002010260510302	2	21	12	6	9	01015000020102605102
	0101500002010260510402	2	51	12	6	4	01015000020102605102
	0101500002010260610102	2	34	12	3	8	01015000020102606102
	0101500002010260610202	2	9	12	6	5	01015000020102606102
	0101500002010260910102	2	50	12	2	9	01015000020102609102
	0101500002010260910202	1	36	12	6	9	01015000020102609102
	0101500003040420110103	1	30	12	5	6	01015000030404201103
	0101500003040420110203	2	38	12	5	10	01015000030404201103
	0101500003040420110303	1	0	12	6	1	01015000030404201103
	0101500003040420210103	1	55	12	1	4	01015000030404202103
	0101500003040420210203	2	30	12	6	6	01015000030404202103
	0101500003040420210303	1	5	12	6	5	01015000030404202103
	0101500003040420310103	2	39	12	6	4	01015000030404203103
	0101500003040420310203	1	17	12	6	7	01015000030404203103
	0101500003040420310303	2	15	12	6	5	01015000030404203103
	0101500003040420310403	1	13	12	6	5	01015000030404203103
	0101500003040420310503	2	4	12	6	1	01015000030404203103
	0101500003040420410103	2	76	12	6	4	01015000030404204103
	0101500003040420410203	2	72	12	6	4	01015000030404204103
	0101500003040420510103	1	29	12	6	9	01015000030404205103
	0101500003040420510203	1	51	12	3	6	01015000030404205103
	0101500003040420510303	1	24	12	6	9	01015000030404205103
	0101500003040420610103	2	46	12	4	4	01015000030404206103

persona 1 x

Hogar

Result Grid									
Filter Rows:									
Edit: Export/Import: Wrap Cell Content: Fetch rows:									
id_hogar	cantidad_cuartos	cuartos_dormitorios	cuartos_negocios	cuarto_cocina	parentesco_propietario	TIPO_COCINA_id_tipoCocina	SERVICIO_HIGIENICO_id_servicioHigienco	BASURA_id_basura	HOGAR_MATER
01015000020102601102	4	3	0	1	2	1	3	2	3
01015000020102602102	3	2	0	1	1	1	4	2	4
01015000020102603102	3	2	0	1	2	1	5	2	5
01015000020102604102	4	3	0	1	2	1	6	2	6
01015000020102605102	6	5	0	1	2	1	7	2	7
01015000020102606102	3	2	0	1	2	1	8	2	8
01015000020102609102	4	3	0	1	1	1	9	2	9
01015000030404201103	3	2	0	1	2	1	10	2	10
01015000030404202103	1	0	0	2	2	1	11	2	11
01015000030404203103	3	2	0	1	2	1	12	2	12
01015000030404204103	3	2	0	1	2	1	13	2	13
01015000030404205103	10	4	0	1	2	1	14	2	14
01015000030404206103	1	0	0	2	2	1	15	2	15
01015000030404207103	2	2	0	1	2	1	16	2	16
01015000110102801101	4	3	0	1	2	1	17	2	17
01015000110102802101	4	3	0	1	2	1	18	2	18
01015000110102804101	3	2	0	1	2	1	19	2	19
01015000110102805101	4	3	0	1	2	1	20	2	20
01015000110102806101	5	3	0	1	2	1	21	2	21
01015000110102807101	3	2	0	1	2	1	22	2	22
01015000110102808101	4	3	0	1	2	1	23	2	23
01015000130703701101	1	1	0	1	2	1	24	2	24
01015000130703702101	5	4	0	1	2	1	25	2	25
01015000130703703101	3	3	1	1	2	1	26	2	26
01015000130703705101	4	3	0	1	2	1	27	2	27
01015000130703707101	1	1	0	2	2	1	28	2	28
01015000130703709101	1	1	0	1	2	1	29	2	29
01015000170102801101	4	3	0	1	2	1	30	2	30
01015000170102802101	5	4	0	1	2	1	31	2	31
01015000170102803101	2	1	0	1	2	1	32	2	32

Vivienda

Result Grid					
Filter Rows:					
Edit: Export/Import: Wrap Cell Content: Fetch rows:					
id_vivienda	ACCESO_PRINCIPAL_id_accesoPrincipal	TENENCIA_VIVIENDA_id_tenciaVivienda	ALUMBRADO_id_alumbrado	TIPO_VIVIENDA_id_tipoVivienda	
0101500002010260102	1	4	1	1	
0101500002010260202	1	1	1	2	
0101500002010260302	1	1	1	2	
0101500002010260402	1	1	1	2	
0101500002010260502	1	4	1	1	
0101500002010260602	1	1	1	2	
0101500002010260902	1	1	1	1	
0101500003040420103	3	1	1	1	
0101500003040420203	1	1	1	3	
0101500003040420303	1	1	1	2	
0101500003040420403	3	4	1	1	
0101500003040420503	3	1	1	1	
0101500003040420603	1	1	1	3	
0101500003040420703	3	5	1	2	
0101500011010280101	1	1	1	2	
0101500011010280201	1	1	1	2	
0101500011010280401	1	1	1	2	
0101500011010280501	1	1	1	2	
0101500011010280601	1	4	1	1	
0101500011010280701	1	4	1	2	
0101500011010280801	1	1	1	2	
0101500013070370101	3	5	1	4	
0101500013070370201	1	4	1	1	
0101500013070370301	1	4	1	1	
0101500013070370501	1	4	1	1	
0101500013070370701	3	1	1	5	
0101500013070370901	1	1	1	3	
0101500017010280101	1	5	1	1	
0101500017010280201	1	4	1	1	
0101500017010280301	1	1	2	2	
0101500017010280401	1	4	1	1	

Script SQL

Enlace al Script SQL de creación y carga de la base de datos. Un solo script que crea la base de datos y carga los datos, se encuentra en la carpeta Script en

ProyectoIntegrador/BaseDatos https://github.com/CarlosSalas8/Proyecto_Integrador.git

Descarga en formato CSV

El enlace al CSV generado, se encuentra en la carpeta Formato_Csv ubicado en Proyecto Integrador/Base Datos

https://github.com/CarlosSalas8/Proyecto_Integrador.git

```
• SELECT *
  INTO OUTFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/EDENMU_Hogar_Persona.csv'
  FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY ''
  LINES TERMINATED BY '\n'
FROM (

  SELECT

    persona.id_Persona as Persona,
    persona.HOGAR_id_hogar as Hogar,
    persona.sexo as Sexo,
    persona.edad as Edad,
    persona.ACTIVIDAD_HOGAR_id_actividad_hogar as ActividadHogar,
    persona.ESTADO_CIVIL_id_estado_civil as EstadoCivil,
    persona.NIVEL_INSTRUCCION_id_nivel_instruccion as NivelInstruccion,

    vivienda.ACCESO_PRINCIPAL_id_accesoPrincipal as ViaAccesoPrincipal,
    vivienda.TENENCIA_VIVIENDA_id_tenciaVivienda as TenenciaVivienda,
    vivienda.ALUMBRADO_id_alumbrado as TipoAlumbrado,
    vivienda.TIPO_VIVIENDA_id_tipoVivienda as TipoVivienda,

    vehiculos.tiene_vehiculo as TieneVehiculo,
    vehiculos.num_vehiculo as NumeroVehiculos,
    vehiculos.tiene_moto as TieneMoto,
    vehiculos.num_moto as NumeroMotos,

    tipo_abastecimiento.abastecimiento_super as AbastecimientoSuper,
    tipo_abastecimiento.abastecimiento_diesel as AbastecimientoDiesel,
    tipo_abastecimiento.abastecimiento_ecopais as AbastecimientoEco,
    tipo_abastecimiento.abastecimiento_extra as AbastecimientoExtra,
    tipo_abastecimiento.abastecimiento_electricidad as AbastecimientoElectricidad,
    tipo_abastecimiento.abastecimiento_gas as AbastecimientoGas,

    gasto_abastecimiento.gasto_super as GastoSuper,
    gasto_abastecimiento.gasto_diesel as GastoDiesel,
```



```

tipo_abastecimiento.tipo_abastecimiento_gas as TipoAbastecimientoGas,

gasto_abastecimiento.gasto_super as GastoSuper,
gasto_abastecimiento.gasto_diesel as GastoDiesel,
gasto_abastecimiento.gasto_ecopais as GastoEco,
gasto_abastecimiento.gasto_extra as GastoExtra,
gasto_abastecimiento.gasto_electricidad as GastoElectricidad,
gasto_abastecimiento.gasto_gas as GastoGas,

costo_hogar.valor_mensualArriendo as ValorMensualArriendo,
costo_hogar.valor_incluyeAgua as IncluyeServicioAgua,
costo_hogar.valor_incluyeLuz as IncluyeServicioLuz,

hogar.cantidad_cuartos as CantidadDeCuartos,
hogar.cuartos_dormitorios as CuartoDormitorios,
hogar.cuartos_negocios as CuartoNegocio,
hogar.cuarto_cocina as CuartoCocina,
hogar.parentesco_propietario as ParentescoPropietario,
hogar.BASURA_id_basura as Basura,
hogar.TIPO_COCINA_id_tipoCocina as TipoCocina,
hogar.VIVIENDA_id_vivienda as Vivienda,

hogar_material.MATERIAL_PISO_id_materialPiso as MaterialPiso,
hogar_material.MATERIAL_PAREDES_id_materialParedes as MaterialParedes,
hogar_material.MATERIAL_TECHO_id_materialTecho as MaterialTecho,

estado_material.ESTADO_PISO_id_estadoPiso as EstadoParedes,
estado_material.ESTADO_TECHO_id_estadoTecho as EstadoTecho,
estado_material.ESTADO_PAREDES_id_estadoParedes as EstadoParedes,

hogar_material.MATERIAL_PISO_id_materialPiso as MaterialPiso,
hogar_material.MATERIAL_PAREDES_id_materialParedes as MaterialParedes,
hogar_material.MATERIAL_TECHO_id_materialTecho as MaterialTecho,

estado_material.ESTADO_PISO_id_estadoPiso as EstadoParedes,
estado_material.ESTADO_TECHO_id_estadoTecho as EstadoTecho,
estado_material.ESTADO_PAREDES_id_estadoParedes as EstadoParedes,

servicio_agua.medidor_agua as MedidorDeAgua,
servicio_agua.junta_agua as ObtieneJuntaAgua,
servicio_agua.RECIBE_AGUA_id_recibeAgua as MedioRecibeAgua,
servicio_agua.OBTIENE_AGUA_id_obtieneAgua as MedioObtieneAgua,
servicio_agua.SERVICIO_DUCHA_id_servicioDucha as ServicioDucha,

servicio_higienico.TIPO_SERVICIO_HIGIENICO_id_tipo_servicioHigienico as TipoDeServicioHigienico,
servicio_higienico.NO_TIENE_SERVICIO_HIGIENICO_id_noTiene_servicioHigienico as NoTieneServicioHigienico,
servicio_higienico.TIPO_INSTALACION_SANITARIA_id_tipo_instalacionSanitaria as TipoInstalacionSanitaria,

locacion.unidad_muestreo as UnidadMuestreo,
locacion.ciudad as Ciudad,
locacion.conglomeracion as Conglomeracion,
locacion.estrato as Estrato,
locacion.area as Area,
replace(locacion.factor_expansion, ',', '.') as FactorExpansion,

tiempo.periodo as Periodo,

```

```
FROM persona

JOIN hogar ON persona.HOGAR_id_hogar = hogar.id_hogar

JOIN servicio_higienico ON hogar.SERVICIO_HIGIENICO_id_servicioHigiencio = servicio_higienico.id_servicioHigiencio

JOIN servicio_agua ON servicio_higienico.SERVICIO_AGUA_id_servicioAgua = servicio_agua.id_servicioAgua

JOIN hogar_material ON hogar.HOGAR_MATERIAL_id_hogarMaterial = hogar_material.id_hogarMaterial

JOIN estado_material ON hogar.ESTADO_MATERIAL_id_estado_material = estado_material.id_estado_material

JOIN vehiculos ON hogar.VEHICULOS_id_Vehiculos = vehiculos.id_Vehiculos

JOIN tipo_abastecimiento ON vehiculos.TIPO_ABASTECIMIENTO_id_tipoAbastecimiento = tipo_abastecimiento.id_tipoAbastecimiento

JOIN gasto_abastecimiento ON tipo_abastecimiento.GASTO_ABASTECIMIENTO_id_gastoAbastecimiento = gasto_abastecimiento.id_gastoAbastecimiento

JOIN costo_hogar ON hogar.COSTO_HOGAR_id_costoHogar = costo_hogar.id_costoHogar

JOIN vivienda ON hogar.VIVIENDA_id_vivienda = vivienda.id_vivienda

JOIN locacion ON locacion.VIVIENDA_id_vivienda = vivienda.id_vivienda

JOIN tiempo ON locacion.TIEMPO_id_tiempo = tiempo.id_tiempo

) AS subconsulta;
```

Bibliografía

- Instituto Nacional de Estadística y Censos. (2023). ENEMDU Trimestral, de*
<https://www.ecuadorencifras.gob.ec/enemdu-trimestral/>
- Instituto Nacional de Estadística y Censos. (2023). Datos abiertos ENEMDU: Primer trimestre 2023 Recuperado el 2 de julio de 2023, de*
https://www.ecuadorencifras.gob.ec/documentos/web-inec/EMPLEO/2023/Trimestre_I/2_BDD_DATOS_ABIERTOS_ENEMDU_2023_I_TRIMESTRE_CSV.zip
- Instituto Nacional de Estadística y Censos (INEC). (2021). Anual 2021 Mercado Laboral y Pobreza. Recuperado de:* https://www.ecuadorencifras.gob.ec/documentos/web-inec/EMPLEO/2021/Anual-2021/Anual_2021_Mercado%20Laboral%20y%20Pobreza%20v1.pdf
- Instituto Nacional de Estadística y Censos (INEC). (Año de Acceso). Resultados Provinciales: Loja. Recuperado de:* <https://www.ecuadorencifras.gob.ec/wp-content/descargas/Manu-lateral/Resultados-provinciales/loja.pdf>