UNIVERSIDAD TÉCNICA

PARTICULAR DE LOJA“CIENCIAS

DE LA COMPUTACIÓN”

Prácticum 1.2

Nombre Estudiante:

Joe Miguel Churo Calderón

Carlos Agustín Salas Churo

Nombre Docente:

Juan Carlos Torres

Tema:

Proyecto Integrador de Saberes

Periodo:

Abril 2023 - Agosto 2023

Contenido

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 Gestiónde 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 informaciónde acceso y las condiciones de vivienda en el Ecuador. A través del uso de técnicas de exploracióny visualización de la información, se busca desarrollar e implementar una solución informática queutilice 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 encontraránrecursos 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 partirde 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 primertrimestre del año 2023.

- **area:** Es un tipo de dato de texto, en la encuesta describe el área en donde seencuentra 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 laencuesta 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"remplazo".
- vivienda: Dato numérico, este dato es un identificador de la vivienda dentro delconglomerado
- **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 estecampo.

- **vi02**: Se refiere al tipo de vivienda, es un dato de tipo numérico, consta de 7opciones que describen el tipo de la vivienda.
- **vi03a:** Es la primera parte de un catálogo para describir el techo o cubierta de lavivienda, 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 estaprimera 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 seindica 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 esteapartado 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 lavivienda, 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 cuartosdedicados solo para dormir.
- **vi07a**: Es un tipo de dato numérico el cual detalla el número de cuartos utilizadosexclusivamente 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 paralo cual se proporcionan 4 opciones
- **vi09**: Es el principio de un conjunto de catálogos, este primer apartado es un tipode 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 3opciones 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 instalaciónsanitaria 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 yen 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 lavivienda 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 recibeagua la vivienda en este campo se tienen 4 opciones.
- **vi11:** Este es un dato numérico, donde se detalla que tipo de ducha tiene lavivienda para la que se ofrecen 3 opciones.
- **vi12:** Este dato numérico describe el tipo de alumbrado de vivienda para estecampo con 4 opciones.
- **vi13**: Tipo de dato numérico en este campo se detalla de qué manera se elimina lamayor 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 contraria pasar a vi151.
- **vi141**: Es un dato numérico en el cual se espera el valor que estarían dispuestos apagar por un arriendo
- **vi142:** Dato numérico, donde se detalla si su pago mensual de arriendo incluye unservicio de agua, para lo que tiene 2 opciones.
- **vi143:** Dato numérico, donde se detalla si su pago mensual de arriendo se incluyeel 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 opcionespara 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.
- **v1531 a vi546**: Datos de tipo numérico describe el combustible más frecuente enla viviendavi531 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.
    - vi1536 y vi1546: Combustible Gas y gasto mensual.
- **estrato**: tipo de dato numérico, detalla el estrato socioeconómico al que pertenecela vivienda.
- **fexp**: Es un tipo de dato numérico, detalla el factor de expansión utilizado paraajustar los resultados.
- **upm**: Dato numérico, detalla la Unidad primaria de muestreo.
- i**d_vivienda:** Dato de tipo texto o numérico, es el identificador único de lavivienda.

- **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 serealizó el estudio.
- **mes**: Dato de tipo texto o numérico que detalla el mes en el que se realizó elestudio.

Datos Complementarios

Los datos complementarios que se mencionarán se basan en el conjunto de datos dela Encuesta Nacional de Empleo, Desempleo y Subempleo (ENEMDU) del Instituto Nacional deEstadística 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 específicospara 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 quepuede 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 unaperspectiva 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 lavivienda

- 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)

- 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)

.5 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)

.6 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)

.7 Análisis de la correlación entre el nivel de instrucción y la forma de eliminación de labasura

- Dato base usado: vi13 (Forma de eliminación de la basura)
- Dato complementario usado: p10a (Nivel de instrucción)

.8 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)

.9 Análisis 10: Examinación de la correlación entre el género y el número de cuartosdedicados solo para dormir

- Dato base usado: vi07 (Número de cuartos para dormir)
- Dato complementario usado: p02 (Sexo)

.10 Análisis 11: Evaluación de la relación entre la actividad del hogar y la

presencia devehículos

- Dato base usado: vi1511 (Presencia de vehículos)
- Dato complementario usado: p21 (Actividad hogar)

.11 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)

.12 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)

.13 Dato complementario usado: p10a (Nivel de instrucción)

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

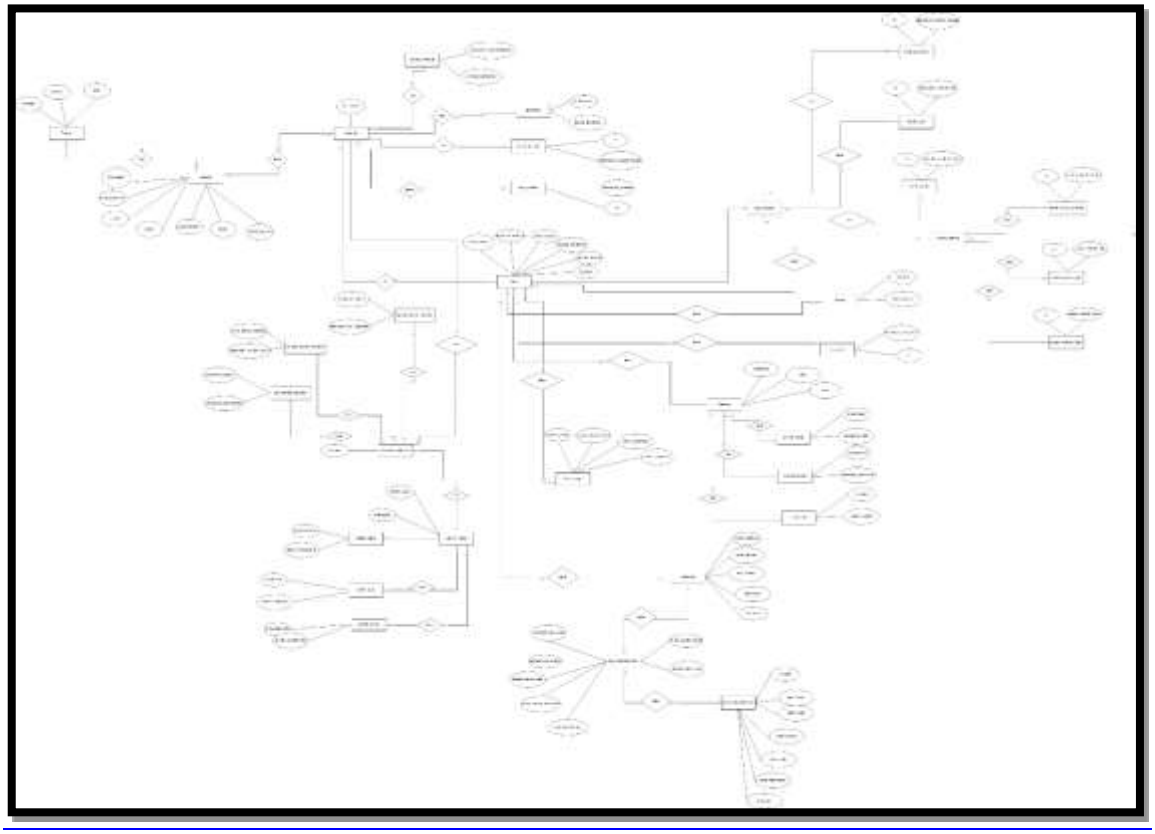
- Dato base usado: vi01 (Vía de acceso a la vivienda)
- Dato complementario usado: p02 (Sexo)

.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)

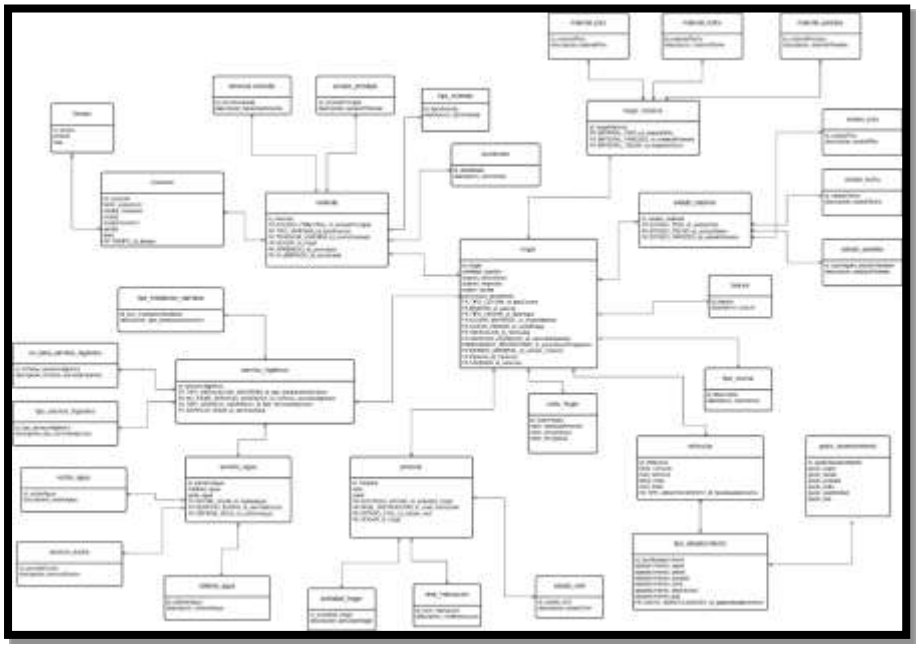
Diseño Conceptual

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, lacual 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



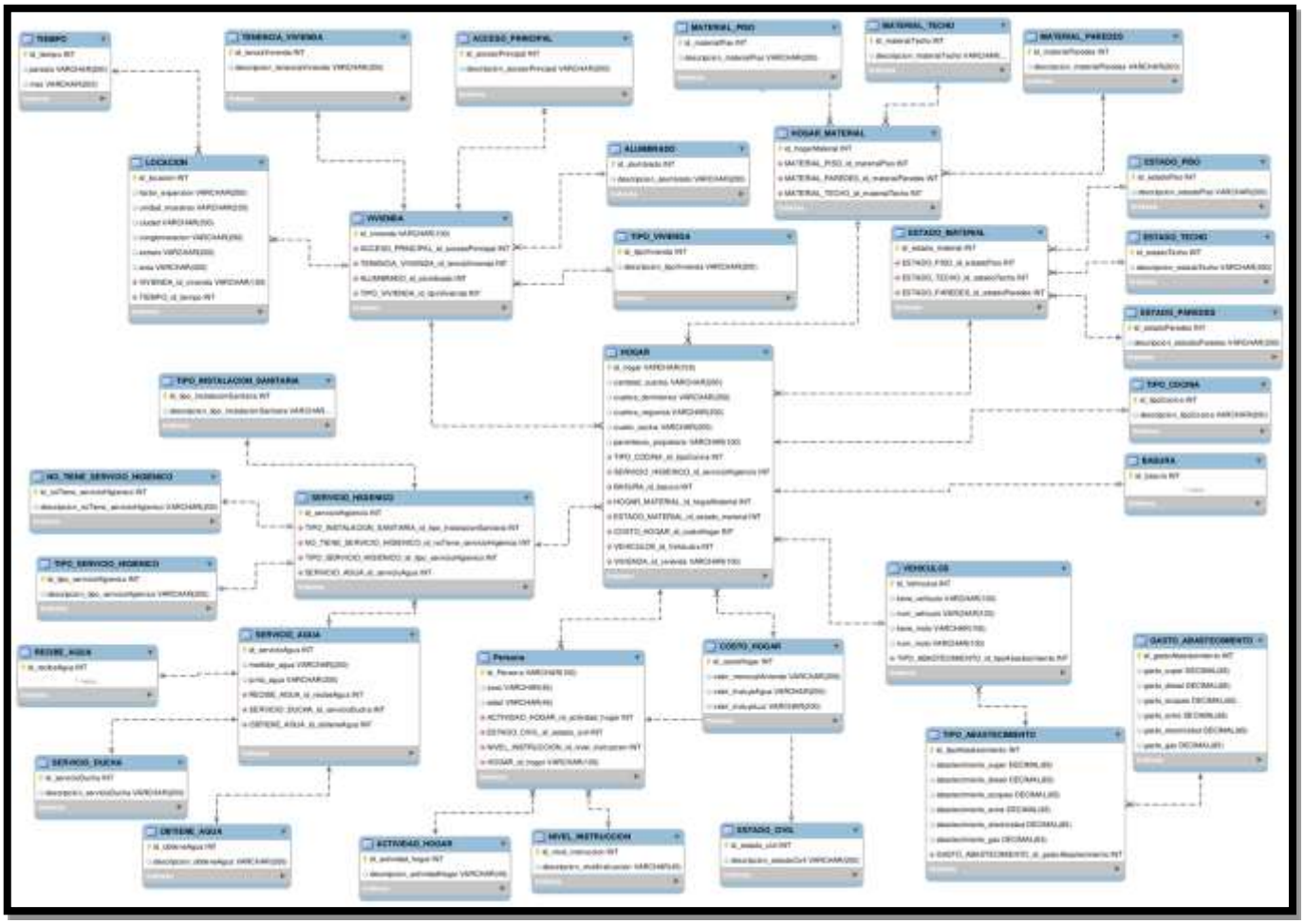
Diseño Lógico Relacional

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 pude visualizar a través de este link: https://drive.google.com/file/d/14ZFaxAIm1Cn6Xslxc2nDzG2fd6vb_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

El proceso de implementación y carga de una base de datos que contiene información socioeconómica relevante para el tercer trimestre, proporcionada por el Instituto Nacional de Estadística y Censos (INEC). El objetivo principal fue unir datos correspondientes a vivienda y personas provenientes de dos archivos CSV distintos y generar un nuevo archivo CSV combinado. Para lograr este cometido, se utilizó MySQL Workbench para diseñar el modelo físico de la base de datos y DataGrip para cargar los datos y realizar operaciones SQL.

En una primera etapa, se obtuvieron dos archivos CSV proporcionados por el INEC. Uno contenía datos de vivienda, mientras que el otro contenía información de personas. Estos datos correspondían al tercer trimestre y eran cruciales para complementar y enriquecer la base de datos. Antes de la carga, se realizó una revisión para garantizar la calidad y coherencia de los datos, asegurando que estuvieran listos para la inserción en la base de datos.

Mediante MySQL Workbench, se diseñó el modelo físico de la base de datos, definiendo las tablas, atributos y restricciones necesarias. Se establecieron las claves primarias y foráneas para mantener la integridad referencial entre las tablas. Este paso fue esencial para garantizar que la información estuviera correctamente estructurada y se pudiera acceder de manera eficiente.

Con la base de datos lista, se procedió a utilizar DataGrip para cargar los datos de los CSV en las tablas correspondientes. Para evitar conflictos con las claves foráneas, se decidió llenar primero aquellas tablas que no dependían de otras, asegurando así que todos los registros fueran insertados de manera exitosa. Posteriormente, se cargaron los datos en las tablas con claves foráneas, garantizando que las relaciones entre entidades estuvieran correctamente establecidas.

Una vez que se contaba con los datos en la base de datos, se llevaron a cabo operaciones JOIN utilizando DataGrip. Estas operaciones permitieron combinar la información de vivienda y personas, tomando como referencia las claves foráneas que vinculaban ambas tablas. De esta manera, se logró unir la información de manera coherente y completa, preparando el terreno para la generación del nuevo archivo CSV.

Finalmente, se utilizó DataGrip para generar un nuevo archivo CSV que contenía los datos combinados de vivienda y personas. Este archivo representaba una fuente valiosa de información para futuros análisis y aplicaciones específicas que requieran datos integrados. Gracias al proceso de unificación realizado mediante consultas SQL, se aseguró que el nuevo CSV reflejara la correlación adecuada entre los datos de ambas categorías.

La implementación y carga de la base de datos, así como la generación del archivo CSV unificado, fueron llevadas a cabo exitosamente. El proceso demostró ser efectivo para combinar datos provenientes de diferentes fuentes y crear una base de datos bien estructurada, facilitando el análisis y manipulación de la información. La correcta utilización de claves foráneas aseguró la integridad referencial de la base de datos, evitando incongruencias en los datos.

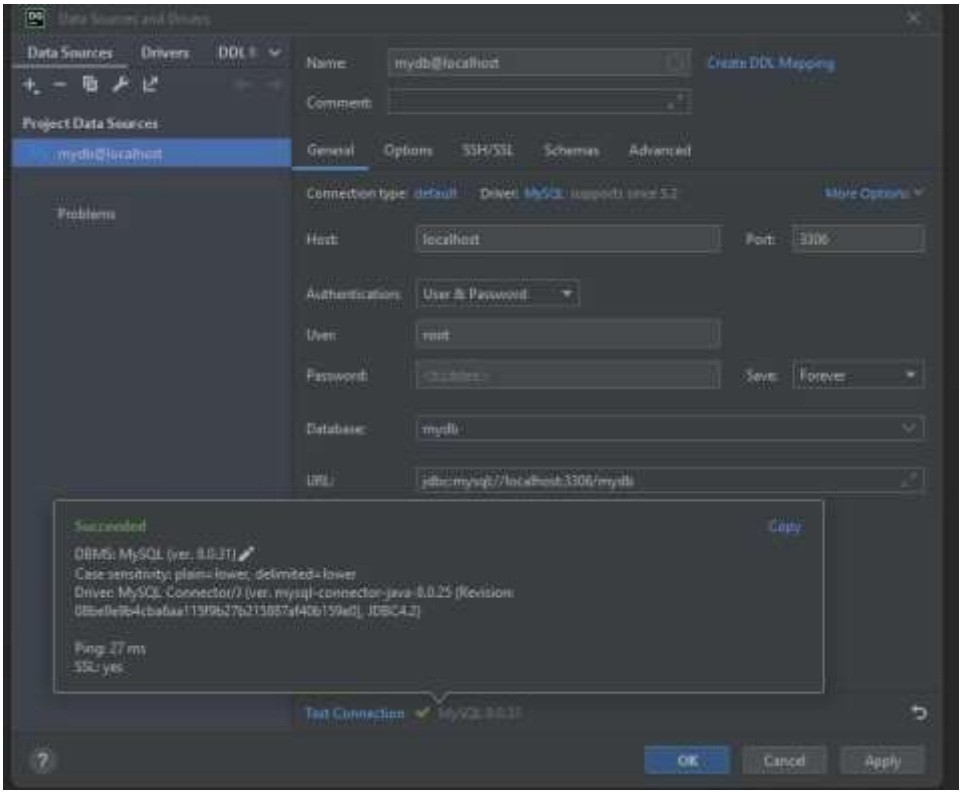
Para la implementación y carga

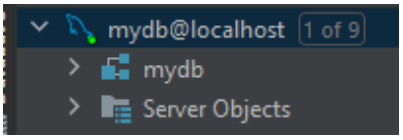
Evidencias de la implementación y carga en MySQL

- Creación Base De Datos: mydb

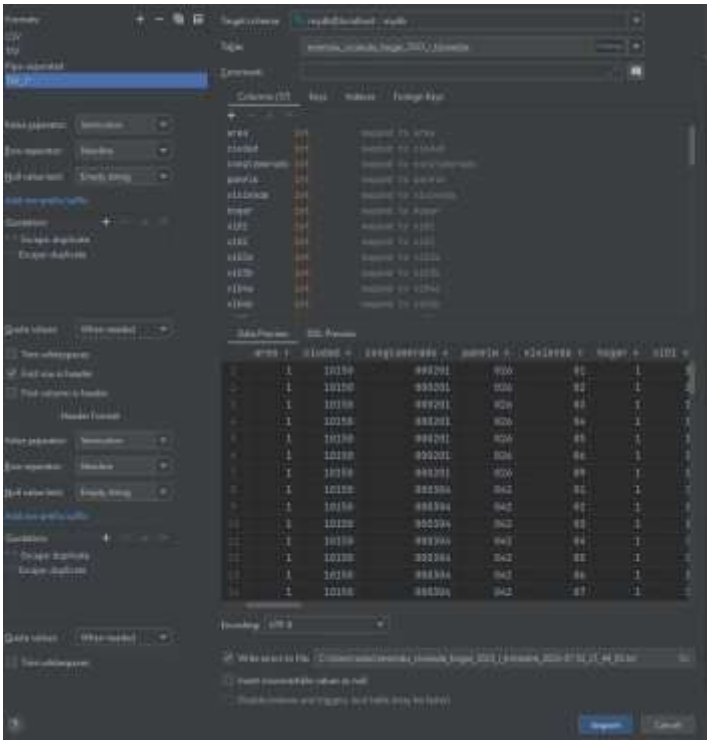
```
-- MySQL Script generated by MySQL Workbench
-- Sun Jul  2 19:00:09 2023
-- Model: New Model    Version: 1.0
-- MySQL Workbench Forward Engineering

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

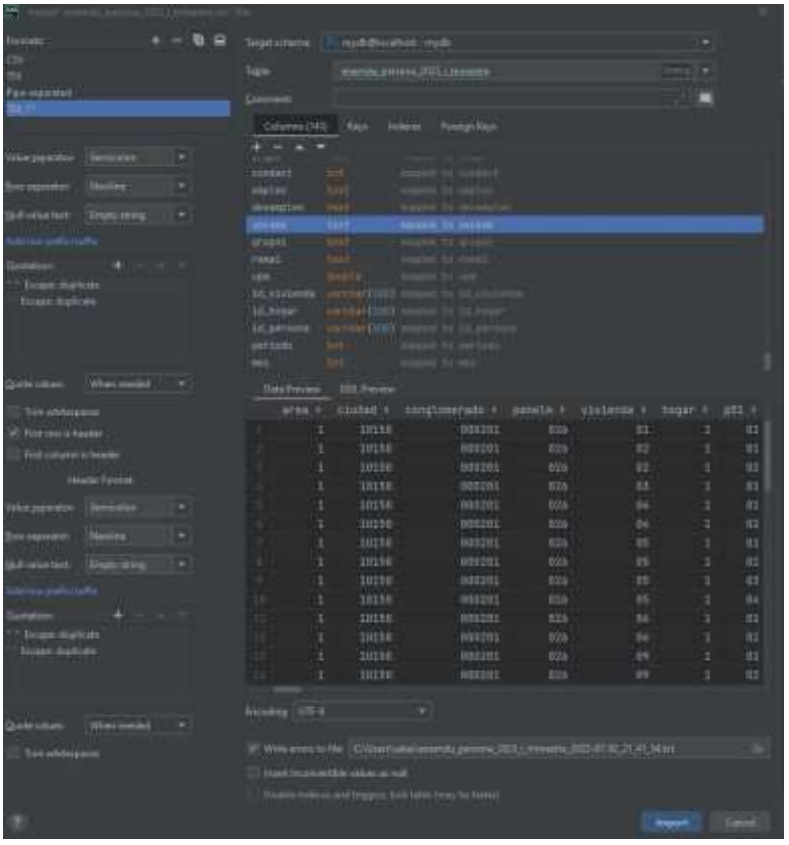




- Import Data From: enemdu_vivienda_hogar_2023_i_trimestre.



- Import Data From: enemdu_persona_2023_i_trimestre.



- mydb


mydb
tables
acceso_principal
actividad_hogar
alumbrado
basura
costo_hogar
enemdu_persona_2023_i_trimestre
enemdu_vivienda_hogar_2023_i_trimestre
estado_civil
estado_material
estado_paredes
estado_piso
estado_techo
gasto_abastecimiento
hogar
hogar_material
locacion
material_paredes
material_piso
material_techo
nivel_instruccion
no_tiene_servicio_higienico
obtiene_agua
persona
recibe_agua
servicio_agua
servicio_ducha
servicio_higienico
tenencia_vivienda
tiempo
tipo_abastecimiento
tipo_cocina
tipo_instalacion_sanitaria
tipo_servicio_higienico
tipo_vivienda
vehiculos
vivienda

- 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_mensualArriendo` ...
✓	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_vehiuclo` 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_TECHO` (`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 15:42:05 INSERT INTO acceso_principal values(1,"Carretera, calle pavimentada o adoquinada")
2070 15:42:05 INSERT INTO acceso_principal values(2,"Empedrado")
2071 15:42:05 INSERT INTO acceso_principal values(3," Lastrado, calle de tierra")
2072 15:42:05 INSERT INTO acceso_principal values(4,"Sendero")
2073 15:42:05 INSERT INTO acceso_principal values(5," Rio, Mar")
2074 15:42:05 INSERT INTO acceso_principal values(6," Otro")
2075 15:42:05 Insert into tenencia_vivienda values (1,"En arriendo ")
2076 15:42:05 Insert into tenencia_vivienda values (2,"Anticresis y/o arriendo")
2077 15:42:05 Insert into tenencia_vivienda values (3,"Propia y la está pagando")
2078 15:42:05 Insert into tenencia_vivienda values (4,"Propia y totalmente pagada")
2079 15:42:05 Insert into tenencia_vivienda values (5,"Cedida")
2080 15:42:05 Insert into tenencia_vivienda values (6,"Recibida por servicios")
2081 15:42:05 Insert into tenencia_vivienda values (7,"Otra")
2082 15:42:05 Insert into alumbrado values (1,"Empresa eléctrica pública")
2083 15:42:05 Insert into alumbrado values (2,"Planta eléctrica privada")
2084 15:42:05 Insert into alumbrado values (3,"Vela, candil, mechero, gas")
2085 15:42:05 Insert into alumbrado values (4,"Ninguno")
2086 15:42:05 insert into tipo_vivienda values (1,"Casa o villa ")
2087 15:42:05 Insert into tipo_vivienda values (2,"Departamento")
2088 15:42:05 Insert into tipo_vivienda values (3,"Cuartos en casa de inquilinato")
2089 15:42:05 Insert into tipo_vivienda values (4,"Mediagua")
2090 15:42:05 Insert into tipo_vivienda values (5,"Rancho, covacha")
2091 15:42:05 Insert into tipo_vivienda values (6,"Choza")
2092 15:42:05 Insert into tipo_vivienda values (7,"Otra")
2093 15:42:05 Insert into estado_paredes values (1,"Bueno")
2094 15:42:05 Insert into estado_paredes values (2,"Regular")
2095 15:42:05 Insert into estado_paredes values (3,"Malo")
2096 15:42:05 Insert into estado_piso values (1,"Bueno")
2097 15:42:05 Insert into estado_piso values (2,"Regular")
2098 15:42:05 Insert into estado_piso values (3,"Malo")
2099 15:42:05 Insert into estado_techo values (1,"Bueno")
2100 15:42:05 Insert into estado_techo values (2,"Regular")
2101 15:42:05 Insert into estado_techo values (3,"Malo")
2102 15:42:05 insert into material_techo values (1,"Hormigón (losa, cemento)")
2103 15:42:05 insert into material_techo values (2,"Fibrocemento,asbesto (eternit, eurolit) ")
2104 15:42:05 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 mametón")
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 respondio')
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 respondio')
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')
```

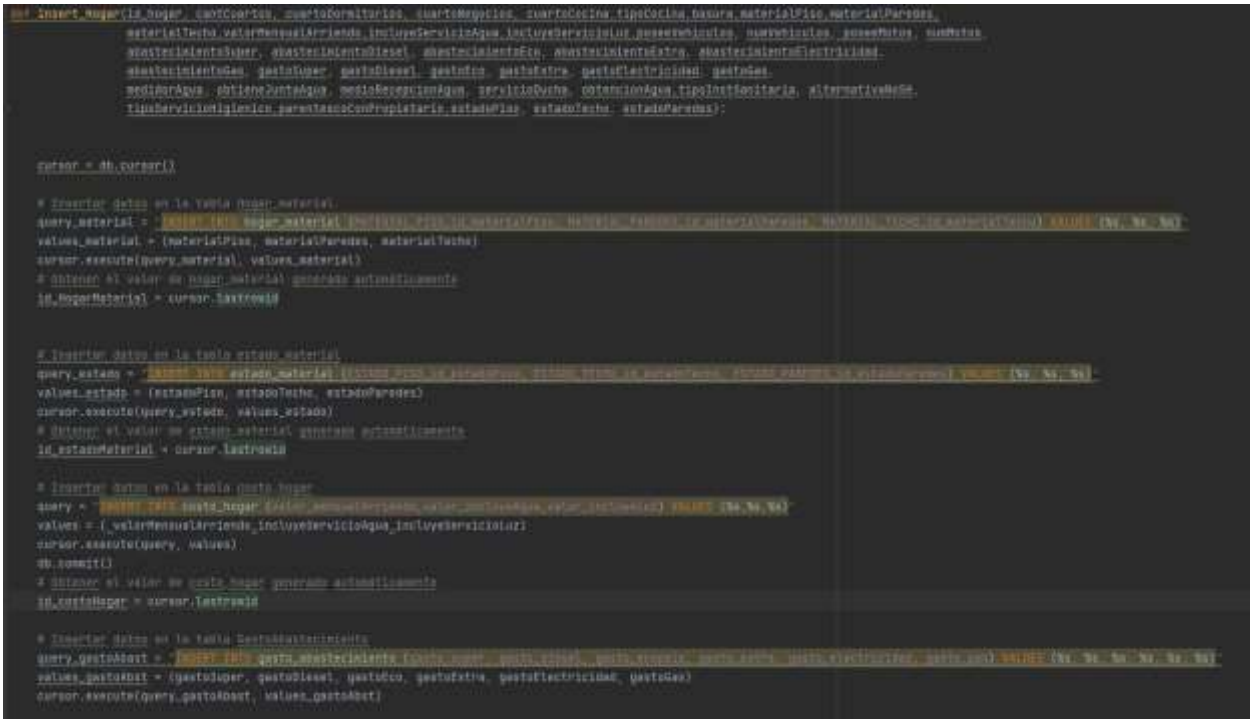
```
2153 19:42:05 INSERT INTO servicio_ducha VALUES (1,'Exclusivo del hogar')
2154 19:42:05 INSERT INTO servicio_ducha VALUES (2,'Compatido con otros hogares')
2155 19:42:05 INSERT INTO servicio_ducha VALUES (3,'No tiene')
2156 19:42:05 INSERT INTO tipo_cocina VALUES (1,'Gas')
2157 19:42:05 INSERT INTO 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,'Viudo(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 algun 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 algun servicio')
2180 19:42:05 INSERT INTO actividad_hogar VALUES (5,'Ayudar en algun negocio familiar')
2181 19:42:05 INSERT INTO actividad_hogar VALUES (6,'Ayudar en el trabajo de algun familiar')
2182 19:42:05 INSERT INTO actividad_hogar VALUES (7,' Aprendiz remunerado')
2183 19:42:05 INSERT INTO actividad_hogar VALUES (8,'Labores agricolas, cuidado animales')
2184 19:42:05 INSERT INTO actividad_hogar VALUES (9,'Estudiante que realiza algun 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
```

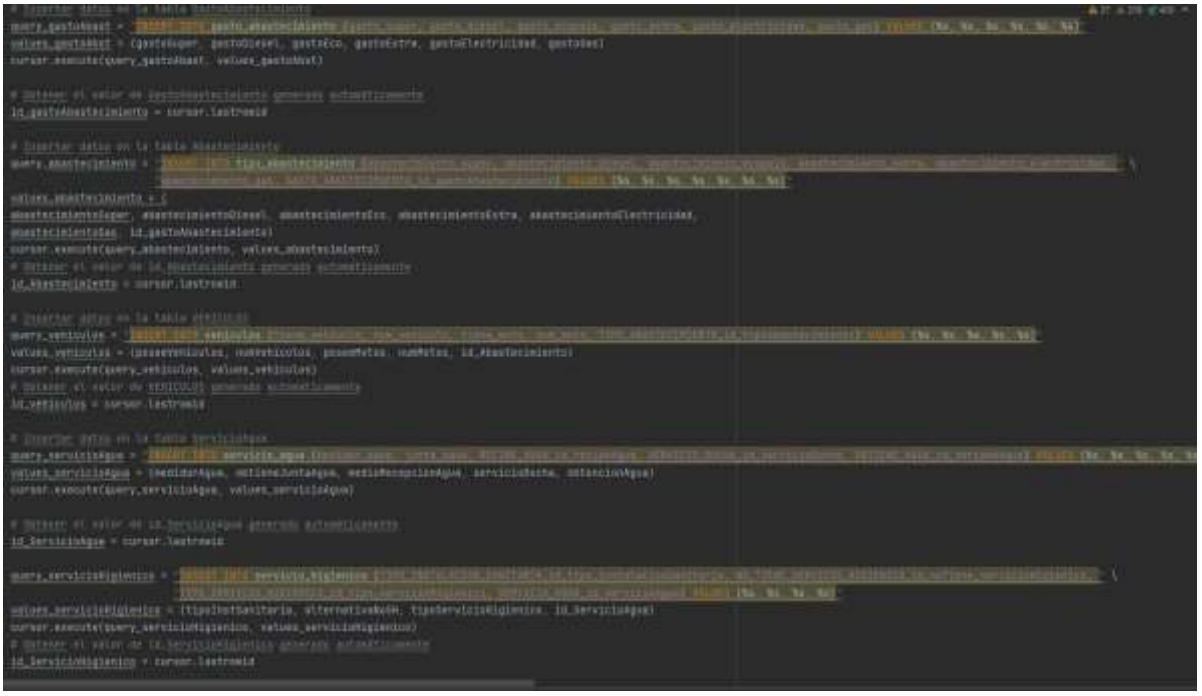
Evidencia


basedatos
mydb
Tables
acceso_principal
actividad_hogar
alumbrado
basura
costo_hogar
estado_civil
estado_material
estado_paredes
estado_piso
estado_techo
gasto_abastecimiento
hogar
hogar_material
locacion
material_paredes
material_piso
material_techo
nivel_instruccion
no_tiene_servicio_higienico
obtiene_agua
persona
recibe_agua
servicio_agua
servicio_ducha
servicio_higienico
tenencia_vivienda
tiempo
tipo_abastecimiento
tipo_cocina
tipo_instalacion_sanitaria
tipo_servicio_higienico
tipo_vivienda
vehiculos
vivienda

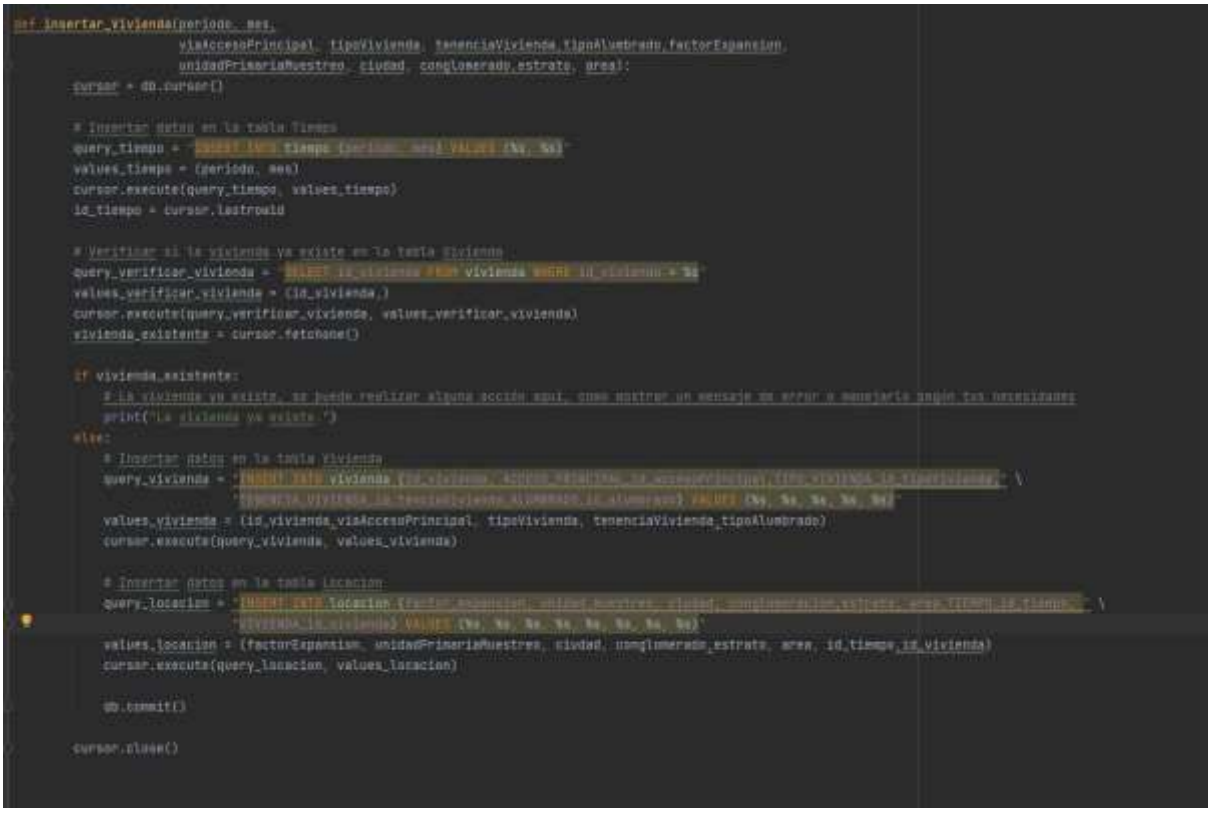
INSERTAR DATOS EN LAS TABLAS

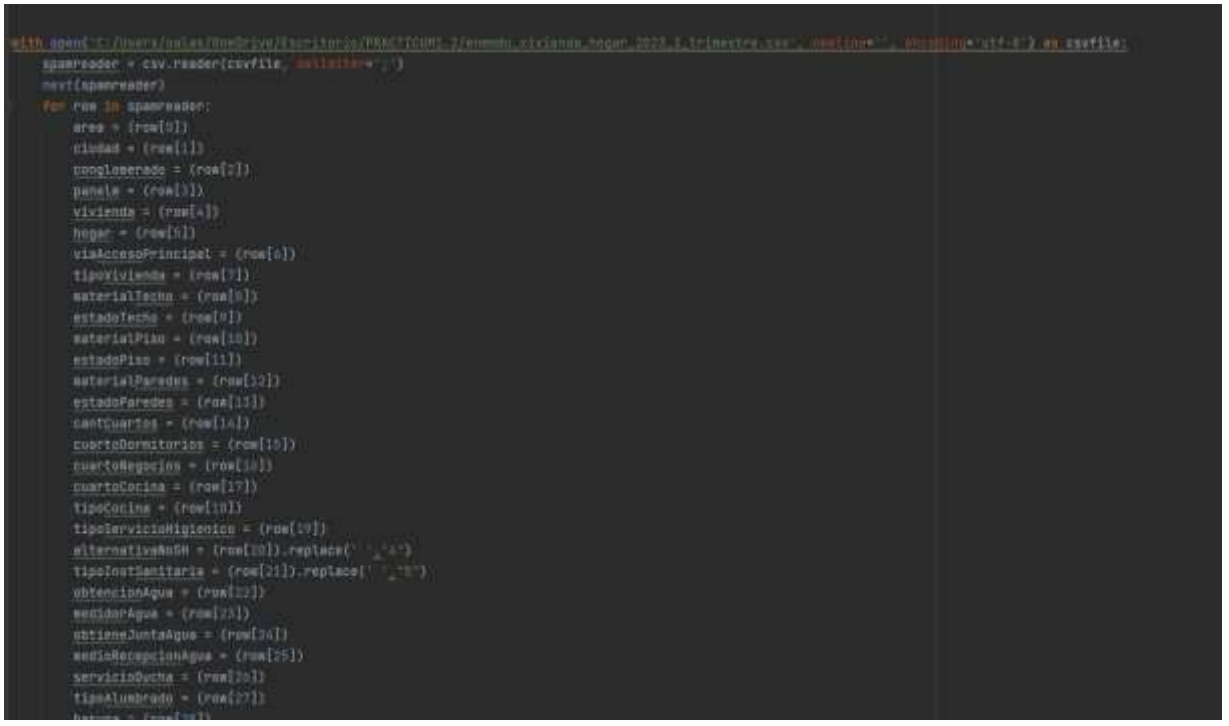
- HOGAR





- VIVIENDA



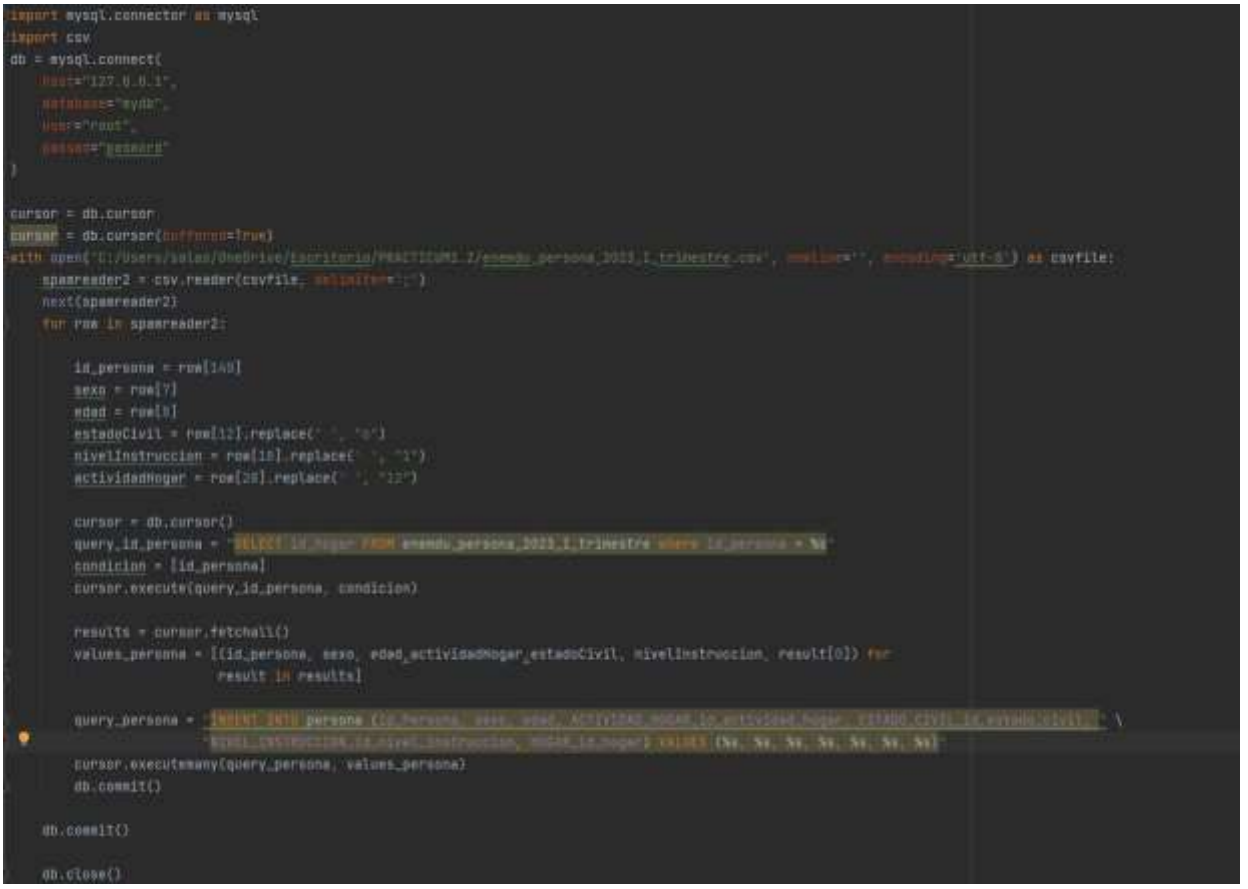


```
with open('C:/Users/calas/OneDrive/Escritorio/PRACTICUMI_2/enemdu_vivienda_hogar_2023_3_trimestre.csv', newline='', encoding='utf-8') as csvfile:
    spamreader = csv.reader(csvfile, delimiter=';')
    next(spamreader)
    for row in spamreader:
        area = (row[0])
        ciudad = (row[1])
        conglomerado = (row[2])
        panela = (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])
        alternativaAnSH = (row[20]).replace(' ','a')
        tipoInstSanitaria = (row[21]).replace(' ','S')
        obtencionAgua = (row[22])
        medidorAgua = (row[23])
        obtieneJuntaAgua = (row[24])
        medioRecepcionAgua = (row[25])
        servicioDucha = (row[26])
        tipoAlumbrado = (row[27])
        horas = (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



- E

jemplo Estado

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

Material Estado

Paredes

Estado Piso

id_estadoPiso	descripcion_estadoPiso
1	Bueno
2	Regular
3	Malo
NULL	NULL

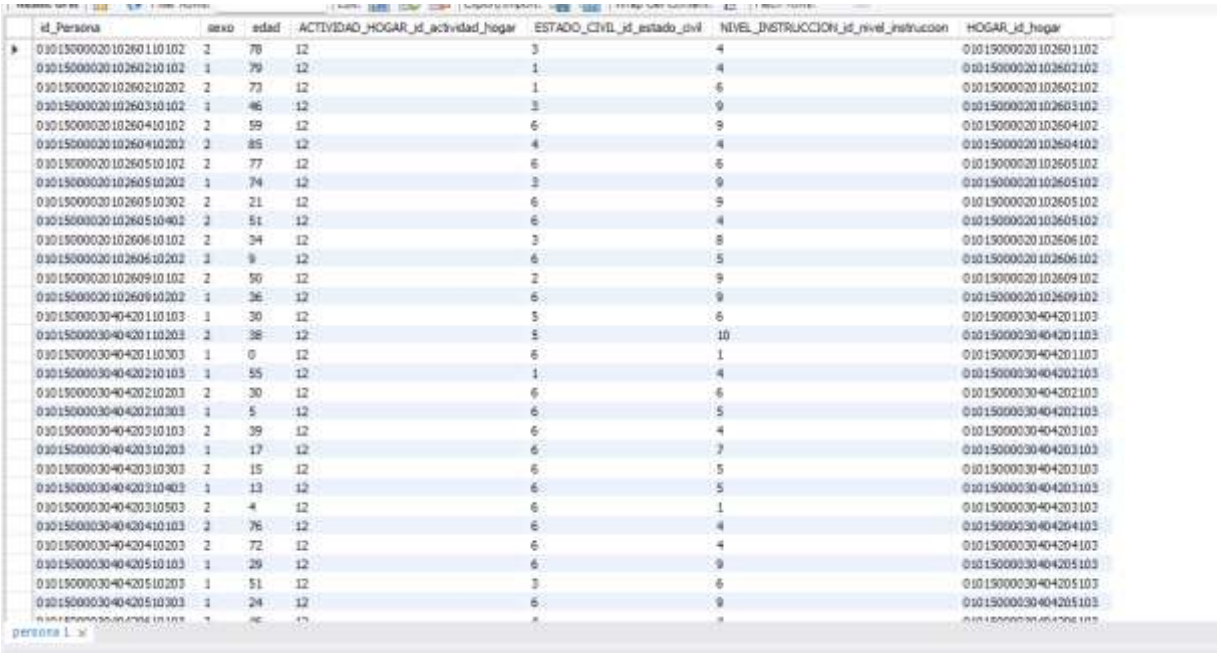
Estado Techo

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

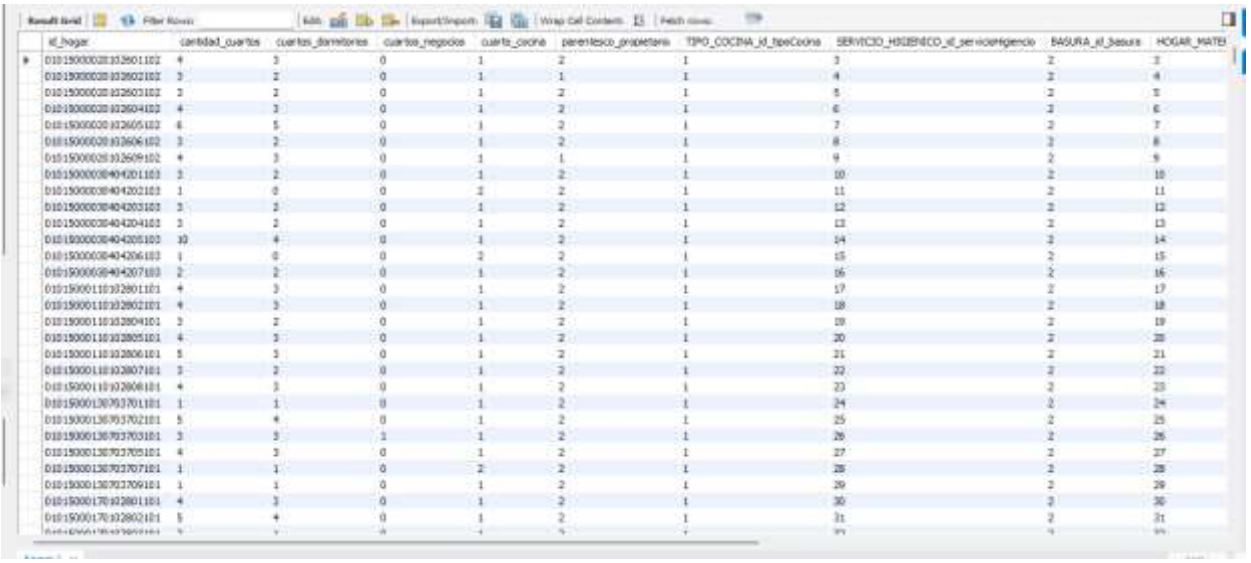
Persona



id_Persona	sexo	edad	ACTIVIDAD_HOGAR_id_actividad_hogar	ESTADO_CIVIL_id_estado_civil	NIVEL_INSTRUCCION_id_nivel_instrucoon	HOGAR_id_hogar
0101500002010260110102	2	78	12	3	4	010150000201026011 02
0101500002010260210102	1	79	12	1	4	010150000201026021 02
0101500002010260210202	2	73	12	1	6	010150000201026021 02
0101500002010260310102	1	46	12	3	9	010150000201026031 02
0101500002010260410102	2	59	12	6	9	010150000201026041 02
0101500002010260410202	2	85	12	4	4	010150000201026041 02
0101500002010260510102	2	77	12	6	6	010150000201026051 02
0101500002010260510202	1	74	12	3	9	010150000201026051 02
0101500002010260510302	2	21	12	6	9	010150000201026051 02
0101500002010260510402	2	51	12	6	4	010150000201026051 02
0101500002010260610102	2	34	12	3	8	010150000201026061 02
0101500002010260610202	2	9	12	6	5	010150000201026061 02
0101500002010260910102	2	50	12	2	9	010150000201026091 02
0101500002010260910202	1	36	12	6	9	010150000201026091 02
0101500003040420110103	1	30	12	5	6	010150000304042011 03
0101500003040420110203	2	38	12	5	10	010150000304042011 03
0101500003040420110303	1	0	12	6	1	010150000304042011 03
0101500003040420210103	1	55	12	1	4	010150000304042021 03
0101500003040420210203	2	30	12	6	6	010150000304042021 03
0101500003040420210303	1	5	12	6	5	010150000304042021 03
0101500003040420310103	2	39	12	6	4	010150000304042031 03
0101500003040420310203	1	17	12	6	7	010150000304042031 03
0101500003040420310303	2	15	12	6	5	010150000304042031 03
0101500003040420310403	1	13	12	6	5	010150000304042031 03
0101500003040420310503	2	4	12	6	1	010150000304042031 03
0101500003040420410103	2	76	12	6	4	010150000304042041 03
0101500003040420410203	2	72	12	6	4	010150000304042041 03
0101500003040420510103	1	29	12	6	9	010150000304042051 03
0101500003040420510203	1	51	12	3	6	010150000304042051 03
0101500003040420510303	1	24	12	6	9	010150000304042051 03

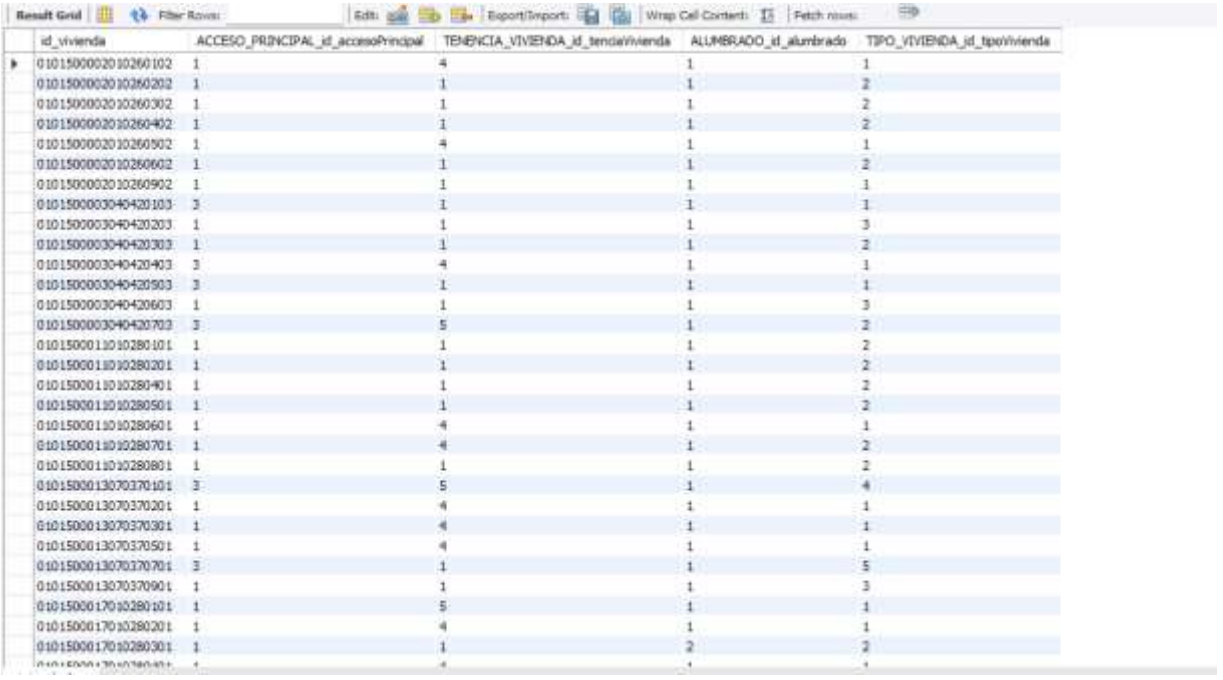
persona 1 ×

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_MATE
010150000201026011 02	4	3	0	1	2	1	3	2	3
010150000201026021 02	3	2	0	1	1	1	4	2	4
010150000201026031 02	3	2	0	1	2	1	5	2	5
010150000201026041 02	4	3	0	1	2	1	6	2	6
010150000201026051 02	6	5	0	1	2	1	7	2	7
010150000201026061 02	3	2	0	1	2	1	8	2	8
010150000201026091 02	4	3	0	1	1	1	9	2	9
010150000304042011 03	3	2	0	1	2	1	10	2	10
010150000304042021 03	1	0	0	2	2	1	11	2	11
010150000304042031 03	3	2	0	1	2	1	12	2	12
010150000304042041 03	3	2	0	1	2	1	13	2	13
010150000304042051 03	10	4	0	1	2	1	14	2	14
010150000304042061 03	1	0	0	2	2	1	15	2	15
010150000304042071 03	2	2	0	1	2	1	16	2	16
010150001101028011 01	4	3	0	1	2	1	17	2	17
010150001101028021 01	4	3	0	1	2	1	18	2	18
010150001101028041 01	3	2	0	1	2	1	19	2	19
010150001101028051 01	4	3	0	1	2	1	20	2	20
010150001101028061 01	5	3	0	1	2	1	21	2	21
010150001101028071 01	3	2	0	1	2	1	22	2	22
010150001101028081 01	4	3	0	1	2	1	23	2	23
010150001307037011 01	1	1	0	1	2	1	24	2	24
010150001307037021 01	5	4	0	1	2	1	25	2	25
010150001307037031 01	3	3	1	1	2	1	26	2	26
010150001307037051 01	4	3	0	1	2	1	27	2	27
010150001307037071 01	1	1	0	2	2	1	28	2	28
010150001307037091 01	1	1	0	1	2	1	29	2	29
010150001701028011 01	4	3	0	1	2	1	30	2	30
010150001701028021 01	5	4	0	1	2	1	31	2	31

Vivienda



id_vivienda	ACCESO_PRINCIPAL_id_accesoPrincipal	TENENCIA_VIVIENDA_id_tenenciaVivienda	ALUMBRADO_id_alumbrado	TIPO_VIVIENDA_id_tipoVivienda
010150000201026010 2	1	4	1	1
010150000201026020 2	1	1	1	2
010150000201026030 2	1	1	1	2
010150000201026040 2	1	1	1	2
010150000201026050 2	1	4	1	1
010150000201026060 2	1	1	1	2
010150000201026090 2	1	1	1	1
010150000304042010 3	3	1	1	1
010150000304042020 3	1	1	1	3
010150000304042030 3	1	1	1	2
010150000304042040 3	3	4	1	1
010150000304042050 3	3	1	1	1
010150000304042060 3	1	1	1	3
010150000304042070 3	3	5	1	2
010150001101028010 1	1	1	1	2
010150001101028020 1	1	1	1	2
010150001101028040 1	1	1	1	2
010150001101028050 1	1	1	1	2
010150001101028060 1	1	4	1	1
010150001101028070 1	1	4	1	2
010150001101028080 1	1	1	1	2
010150001307037010 1	3	5	1	4
010150001307037020 1	1	4	1	1
010150001307037030 1	1	4	1	1
010150001307037050 1	1	4	1	1
010150001307037070 1	3	1	1	5
010150001307037090 1	1	1	1	3
010150001701028010 1	1	5	1	1
010150001701028020 1	1	4	1	1
010150001701028030 1	1	1	2	2

Script SQL

Enlace al Script SQL de creación y carga de la base de datos. Un solo script que crea labase 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 ProyectoIntegrador/Base Datos

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

```
tipo_abastecimiento.abastecimiento_gas as AbastecimientoGas,

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


costo_hogar.valor_mensualArriendo as ValorMnsualArriendo,
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 TipoInsatalacionSanitaria,



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.MOBAR_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.MOGAR_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;
```

Complemento de Programación

Explicación de herramientas utilizadas

Para el proyecto integrador de saberes, específicamente para la parte de PROGRAMACIÓN AVANZADA, utilizamos diversas herramientas, consiste en la integración de varios lenguajes y frameworks que nos permitan realizar diversos análisis correspondientes al tema de vivienda y personas para el tercer trimestre de la INEC. Con la parte de la base de datos terminada y lista para su uso, de igual forma con el csv generado de la vivienda y persona, decidimos tomar esos recursos para poder realizar distintos análisis y presentarlos por medio distintos gráficos según sea su conveniencia, integrando varios lenguajes de programación que, a lo largo de los ciclos, los integrantes del grupo fueron conociendo y aprendiendo para ponerlos en práctica, las herramientas o lenguajes utilizados fueron:

Apache Zeppelin: Es una herramienta de cuadernos interactivos y colaborativos que permite trabajar con código, visualizaciones y texto en un mismo documento. Es compatible con varios lenguajes de programación, como Scala, Python, SQL, entre otros, lo que lo hace ideal para análisis de datos y

presentaciones.

Spark: Apache Spark es un potente motor de procesamiento de datos en tiempo real y en batch. Proporciona una interfaz fácil de usar para el procesamiento distribuido, lo que lo convierte en una herramienta esencial para manejar grandes conjuntos de datos y realizar operaciones complejas de análisis de datos.

SQL: SQL (Structured Query Language) es un lenguaje de programación diseñado para administrar y consultar bases de datos relacionales. Permite realizar consultas sobre los datos almacenados en la base de datos, lo que es esencial para realizar análisis de datos y obtener información específica.

Scala: Es un lenguaje de programación versátil que se ejecuta en la plataforma Java Virtual Machine (JVM). Es una opción popular para el procesamiento de datos en Apache Spark debido a su eficiencia y capacidad para aprovechar al máximo las características de Spark.

Angular: Angular es un popular framework de desarrollo de aplicaciones web front-end. Se utiliza para construir interfaces de usuario interactivas y dinámicas que permiten visualizar los resultados y análisis de datos generados por la aplicación.

Python: Python es un lenguaje de programación ampliamente utilizado y muy versátil. Es especialmente popular en el ámbito del análisis de datos debido a sus numerosas bibliotecas y herramientas, como Pandas y Matplotlib, que facilitan la manipulación y visualización de datos.

Explicación de comandos y sentencias usadas para consultar y visualizar datos tanto a nivel del archivo fuente, como a nivel de base de datos MySQL.

Como mencionamos anteriormente, hicimos uso de Apache Zeppelin gracias a su versatilidad para trabajar con una variedad de intérpretes, lo que nos permitió integrar varias tecnologías en un solo lugar. En este entorno, uno de los primeros pasos que seguimos fue cargar el archivo CSV generado por nuestro componente de base de datos. Para realizar esta tarea, utilizamos Scala junto con Apache Spark. En particular, la interfaz SparkSession de Spark resulta extremadamente útil ya que proporciona métodos para leer archivos CSV de manera eficiente.

Para realizar la lectura, declaramos una variable inmutable llamada `data` que inicializamos con nuestra instancia de SparkSession existente. Luego, invocamos el método `read` en esta instancia,

lo que nos permitió comenzar el proceso de lectura del archivo.

Para garantizar que Spark interpretara correctamente nuestro archivo CSV, configuramos varias opciones:

- Con **.option("inferSchema", "true")**, le indicamos a Spark que debería intentar inferir automáticamente el esquema de los datos a medida que los lee. Esta característica es crucial ya que nos ahorra tener que especificar manualmente el tipo de cada columna.

- Utilizamos **.option("header", "true")** para que Spark entienda que la primera fila de nuestro CSV contiene los encabezados de las columnas.

- Configuramos el delimitador de nuestro archivo CSV con **.option("delimiter", ",")**. En nuestro caso, cada valor se separa con una coma.

Finalmente, especificamos la ruta de nuestro archivo CSV con el método **.csv().** En nuestro caso, este archivo se encontraba en la ruta */home/joe/Música/integrador.csv*.

```
val data = spark
.read
.option("inferSchema", "true")
.option("header", "true")
.option("delimiter", ",")
.csv("/home/joe/Música/integrador.csv")
```

A través de este proceso, pudimos cargar nuestro archivo CSV en un DataFrame de Spark, una estructura de datos flexible y potente que facilitó el análisis y la manipulación de nuestros datos en las etapas posteriores del proyecto.

Una vez que el archivo CSV fue cargado en la estructura de DataFrame con éxito, el siguiente paso consistió en familiarizarnos con la estructura y el contenido de nuestros datos. Este entendimiento es crucial para poder realizar un análisis de datos efectivo y eficiente.

Para obtener una visión clara de la estructura del DataFrame, utilizamos el método .printSchema() sobre nuestra variable data. Este método imprime el esquema del DataFrame en un formato fácilmente legible, proporcionando información valiosa como los nombres de las columnas y los tipos de datos que contiene cada una de ellas.

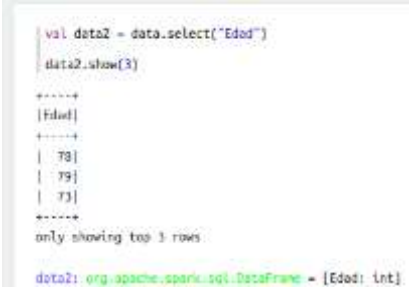
Aquí mostramos cómo se utilizó esta función para visualizar el esquema del DataFrame:

Las operaciones con Spark se realizan a través de abstracciones de datos llamadas RDD (Resilient Distributed Datasets) y DataFrame. Un DataFrame es una colección distribuida de datos organizados en columnas con nombre, similar a una tabla en una base de datos relacional, que permite a los desarrolladores realizar operaciones de análisis de datos de manera más fácil y eficiente.

En este proyecto, se utilizó una serie de métodos proporcionados por Spark y Scala para manipular y analizar los datos en nuestro DataFrame. Estos métodos permiten realizar una amplia gama de operaciones, desde la selección y filtrado de datos hasta la realización de cálculos estadísticos. En las siguientes secciones, se proporciona una descripción detallada de estos métodos, su uso en el proyecto y los resultados que proporcionan.

1. Método select():

El método select() se utiliza para seleccionar columnas específicas de un DataFrame. En este proyecto, este método se utilizó para seleccionar las columnas "numero_cuartos" y "numero_dormitorios" del DataFrame.



Como resultado obtenemos un nuevo DataFrame que contiene solo las columnas especificadas, en este ejemplo la columna “Edad” de nuestro csv, ademas el uso del metodo .show() para mostrar

los datos obtenidos como se observa en el ejemplo

3. Método count():

El método count() cuenta el número de filas en un DataFrame o en grupos específicos si se usa después de groupBy(). En este proyecto, se utilizó para calcular el número de registros para cada número único de las columnas que necesitemos

```
val data2 = data.select(count("Edad").alias("ConteoRegistros"))
data2.show()
```

```
+---------------+
|ConteoRegistros|
+---------------+
|          87399|
+---------------+
```

Con esto tenemos un entero que representa el número de registros en el DataFrame en este ejemplo edad tiene 87,399 registros, para este ejemplo aplicamos el metodo .alias() metodo bastante comun en nuestro proyecto este metodo lo usamos para renombrar columnas en el dataframe

4. Método groupBy():

El método groupBy() se utiliza para agrupar el DataFrame según los valores de una o varias columnas. Esta operación es similar a la operación SQL GROUP BY que agrupa los datos en base a una o varias columnas para permitir operaciones de agregación como sumar, contar, promediar, etc.

```
val data2 = data.select("Edad").groupBy("Edad").count()

data2.show(3)
```

```
+----+-----+
|Edad|count|
+----+-----+
|  31|  930|
|  85|  206|
|  65|  905|
+----+-----+
only showing top 3 rows
```

El resultado obtenido en el DataFrame data2 será una tabla que muestra cada valor único de edad encontrado en el DataFrame original y la cantidad de veces que se repite ese valor en el conjunto de datos.

5. Método filter():
El método filter() en Spark se utiliza para filtrar las filas del DataFrame basándose en una condición dada. Este método devuelve un nuevo DataFrame con solo las filas que cumplen con la condición especificada.

```
val data2 = data.filter(col("Edad") === 23)
  .groupBy("Area")
  .count()
data2.show()
```

```
+----+-----+
|Area|count|
+----+-----+
|   1| 1133|
|   2|  372|
+----+-----+
```

En este ejemplo, estamos filtrando el DataFrame para obtener solo las personas cuya edad es igual a 23. Luego, realizamos un groupBy según el campo "Área" y contamos cuántas personas cumplen con esta condición en cada área específica. El resultado es un resumen que muestra cuántas personas de 23 años hay en cada área.

6. Método where():
El método where() es esencialmente un alias para filter(). Se utiliza para filtrar las filas de un DataFrame en base a una condición determinada.

```
val data2 = data.where(col("Edad") === 23).groupBy("Area").count()
data2.show()
```

```
+----+-----+
|Area|count|
+----+-----+
|   1| 1133|
|   2|  372|
+----+-----+

data2: org.apache.spark.sql.DataFrame = [Area: int, count: bigint]
```

En este ejemplo, usamos el método where para filtrar el DataFrame y obtener solo las personas cuya edad es igual a 23. Luego, realizamos un groupBy según el campo "Área" y contamos cuántas personas cumplen con esta condición en cada área específica. El resultado es un resumen que muestra cuántas personas de 23 años hay en cada área.

8. Método mean(), min(), max():
Estos métodos se utilizan para calcular estadísticas básicas de las columnas de un DataFrame. En

este proyecto, se usaron para calcular la media, el mínimo y el máximo de la columna "numero_cuartos".

```
val columna = "gasto_super"

val dfSuper = data2.select(columna).where(col(columna).isNotNull)

val media = dfSuper.agg(functions.avg(functions.col(columna))).first().getDouble(0)
val stdev = dfSuper.agg(stddev(columna)).first().getDouble(0)

val mediana = dfSuper.stat.approxQuantile(columna, Array(0.5), 0.0)(0)
val mad = dfSuper.withColumn("deviation", abs(col(columna) - lit(mediana)))
  .stat.approxQuantile("deviation", Array(0.5), 0.0)(0)

val cuartiles = dfSuper.stat.approxQuantile(columna, Array(0.25, 0.75), 0.0)
val q1 = cuartiles(0)
val q3 = cuartiles(1)
val iqr = q3 - q1
val limiteInferior = q1 - 1.5 * iqr
val limiteSuperior = q3 + 1.5 * iqr

val superNoOutliers = dfSuper
  .where(col(columna) > limiteInferior && col(columna) < limiteSuperior)

println("SIN OUTLIERS")

superNoOutliers.summary().show
```

```
SIN OUTLIERS
+-------+-----------------+
|summary|      gasto_super|
+-------+-----------------+
|  count|              484|
|   mean|67.69834710743801|
| stddev|47.15596690402599|
|    min|              0.0|
|    25%|             30.0|
|    50%|             60.0|
|    75%|            100.0|
|    max|            200.0|
+-------+-----------------+
```

En este ejemplo, tenemos un DataFrame llamado datas que contiene datos relacionados con el gasto en supermercados (GastoSuper). Primero, filtramos el DataFrame para obtener solo las filas donde la columna GastoSuper no es nula y luego realizamos varias operaciones estadísticas en esta columna.

Calculamos la media y la desviación estándar utilizando las funciones avg y stddev, respectivamente. Luego, calculamos la mediana y la desviación absoluta mediana (MAD) utilizando la aproximación cuantil. También calculamos los cuartiles (Q1 y Q3) y el rango intercuartílico (IQR) para identificar outliers mediante el método Tukey.

Finalmente, filtramos el DataFrame original para obtener solo las filas que no son outliers y mostramos un resumen estadístico del DataFrame resultante utilizando el método summary.

El uso de las funciones mean, min y max nos permite obtener estadísticas básicas y útiles de la columna, como la media, el valor mínimo y el valor máximo. Estas estadísticas son fundamentales para comprender la distribución de los datos y detectar valores atípicos (outliers) que puedan afectar el análisis y los resultados. En este caso, hemos aplicado estas funciones en conjunto con otros métodos estadísticos para realizar un análisis más completo del gasto en supermercados y detectar outliers.

Consultas MySql forma 1

En nuestro proyecto, se realizan consultas SQL para extraer datos precisos de nuestra base de datos. Estos datos extraídos son esenciales para obtener información y análisis de los datos. Gracias a las potentes capacidades de Apache Spark y su interoperabilidad con SQL y otros lenguajes de programación, podemos ejecutar consultas SQL directamente en nuestra base de datos y trabajar con los resultados como si fueran DataFrames. Esto nos permite aprovechar las operaciones de transformación y acción de Spark para analizar y procesar datos de manera eficiente.

```
val query = """
    (SELECT tv.descripcion_tipoVivienda AS TipoVivienda, COUNT(*) AS TotalPersonas
    FROM tipo_vivienda tv
    JOIN vivienda v ON tv.id_tipoVivienda = v.TIPO_VIVIENDA_id_tipoVivienda
    JOIN hogar h ON v.id_vivienda = h.VIVIENDA_id_vivienda
    JOIN servicio_higienico sh ON h.SERVICIO_HIGIENICO_id_servicioHigiencio = sh.id_servicioHigiencio
    GROUP BY TipoVivienda, tv.descripcion_tipoVivienda) as qryData
"""

val tvPersonas = spark.read
    .format("jdbc")
    .option("url", "jdbc:mysql://localhost:3306/vivienda")
    .option("driver", "com.mysql.jdbc.Driver")
    .option("user", "root")
    .option("password", "12062003")
    .option("dbtable", query)
    .load()

tvPersonas.show
```

```
+--------------------+-------------+
|        TipoVivienda|TotalPersonas|
+--------------------+-------------+
|       Casa o villa |        15716|
|        Departamento|         8056|
|Cuartos en casa d...|          678|
|            Mediagua|          807|
|     Rancho, covacha|         1317|
|               Choza|            6|
+--------------------+-------------+
```

El código anterior establece una conexión con una base de datos MySQL y ejecuta una consulta SQL para obtener datos. Estos datos son luego leídos en un DataFrame de Spark para su posterior análisis y procesamiento.

1. La consulta SQL se guarda en la variable query. Esta consulta SQL selecciona el tipo de vivienda y el total de personas de varias tablas relacionadas en la base de datos.

2. Se utiliza el método read de la SparkSession para leer los datos.

3. Se utiliza el método format("jdbc") para especificar que los datos se deben leer utilizando el conector JDBC.

4. El método option se utiliza para configurar las propiedades de la conexión a la base de

datos, como la URL de la base de datos, el controlador JDBC, el nombre de usuario y la contraseña.

5. El método option("dbtable", query) se utiliza para especificar que los datos deben ser leídos ejecutando la consulta SQL especificada en la base de datos.

6. Finalmente, el método load se utiliza para cargar los datos y devolver un DataFrame.

Esta forma de trabajar es útil ya que permite realizar consultas SQL directamente en la base de datos y luego trabajar con los resultados como si fueran DataFrames en Spark. Esto combina las ventajas de SQL, un lenguaje conocido por su capacidad para manipular y consultar datos, con las capacidades de procesamiento de datos distribuidos de Spark. En otras palabras, nos permite hacer uso de la potencia de SQL para la extracción de datos y la potencia de Spark para el procesamiento y análisis de los mismos.

Consultas MySql forma 2

Esta consulta SQL se ejecuta en Apache Zeppelin. La consulta selecciona la descripción del tipo de vivienda y cuenta el número total de personas por tipo de vivienda. Este recuento se realiza uniendo varias tablas (tipo_vivienda, vivienda, hogar, servicio_higienico) en la base de datos usando la cláusula JOIN.

TipoVivienda	TotalPersonas
Casa o villa	15710
Departamento	8256
Cuartos en casa de inquilinato	678
Mediagua	807
Rancho, covacha	1317
Choza	6

El resultado de la consulta se guarda en la variable result mediante la directiva saveAs de Zeppelin. Esta característica permite guardar el resultado de una consulta para su uso en celdas posteriores de la misma nota o incluso en otras notas de Zeppelin.

Python y pandas

Uno de los beneficios clave de utilizar Apache Zeppelin es su capacidad para admitir múltiples lenguajes de programación en una sola nota. Esto nos permite integrar la potencia de procesamiento paralelo de Apache Spark con las capacidades de visualización de Python.

Pasar data frame a python:

1. z.put("dfpy", df): En este paso, almacenamos el DataFrame Spark df en la variable compartida Zeppelin dfpy. La función put() de Zeppelin nos permite guardar un objeto de

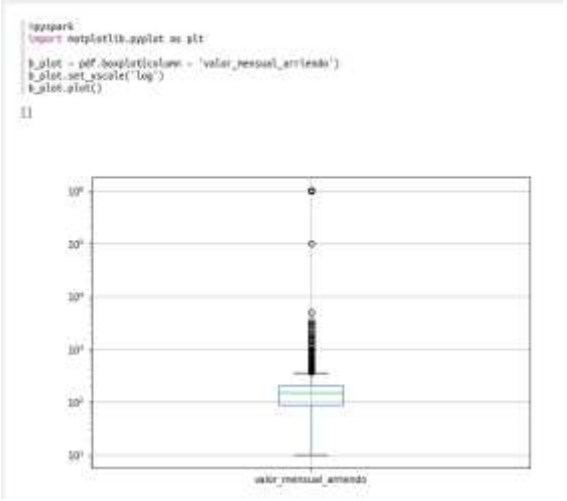
Spark en el espacio de nombres de Zeppelin, lo que nos permite compartir datos entre diferentes interpretes (como Scala y Python) en la misma nota.

2. %pyspark: Este es un comando mágico de Zeppelin que nos permite cambiar el interprete de la celda a PySpark.

3. df = DataFrame(z.get("dfpy"), sqlContext): Aquí, recuperamos el DataFrame almacenado anteriormente usando la función get() de Zeppelin y lo convertimos en un DataFrame PySpark. El argumento de z.get("dfpy") es el nombre de la variable que almacenamos en el paso 1.

4. import pandas as pd: Aquí, importamos la biblioteca pandas, que es una de las bibliotecas de análisis y manipulación de datos más populares en Python.

5. pdf = df.toPandas(): En este último paso, convertimos nuestro DataFrame de PySpark a un DataFrame de pandas para poder utilizar las capacidades de visualización de pandas. toPandas() es un método de PySpark DataFrame que convierte el DataFrame de PySpark en un DataFrame de pandas.



En este fragmento de código, se utiliza la biblioteca de visualización matplotlib de Python para crear un gráfico de caja, una herramienta común de visualización de datos que proporciona una representación gráfica de la distribución de los datos. Aquí es cómo funciona:

1. import matplotlib.pyplot as plt: Esta línea importa el módulo pyplot de la biblioteca matplotlib, que proporciona funciones para hacer que la creación de gráficos sea tan sencilla como posible.

2. b_plot = pdf.boxplot(column = 'valor_mensual_arriendo'): Aquí, el método boxplot() de pandas se utiliza para crear un gráfico de caja para la columna 'valor_mensual_arriendo'.

El gráfico de caja se guarda en la variable b_plot.

3. b_plot.set_yscale('log'): Esta línea ajusta la escala del eje y al formato logarítmico. Esto se hace a menudo cuando se trabaja con datos que varían en varios órdenes de magnitud, ya que el formato logarítmico puede hacer que estos datos sean más fáciles de interpretar visualmente.

4. b_plot.plot(): Finalmente, se llama al método plot() para dibujar el gráfico de caja.

El gráfico de caja resultante proporciona una visualización clara de la distribución de los datos en 'valor_mensual_arriendo', mostrando cosas como la mediana, los cuartiles y los posibles valores atípicos. Al utilizar una escala logarítmica para el eje y, se puede manejar una amplia gama de valores de arrendamiento mensuales, facilitando la visualización e interpretación de los datos.

Scala y python a Angular

En esta etapa, convertimos un DataFrame en Spark a formato JSON, que es una representación de los datos en formato de texto, fácilmente interpretable por humanos y máquinas. En este proceso, transformamos cada fila del DataFrame a un objeto JSON.

Pasar JSON a Angular:

Una vez convertido el DataFrame a JSON, pasamos este JSON a una visualización de Angular en Zeppelin. En este contexto, Angular es una plataforma para desarrollar aplicaciones web y "Zeppelin" es una herramienta que permite crear cuadernos interactivos con soporte para múltiples lenguajes de programación.

Ver el JSON en Angular:

Luego, creamos un campo de entrada en Angular para visualizar el JSON.

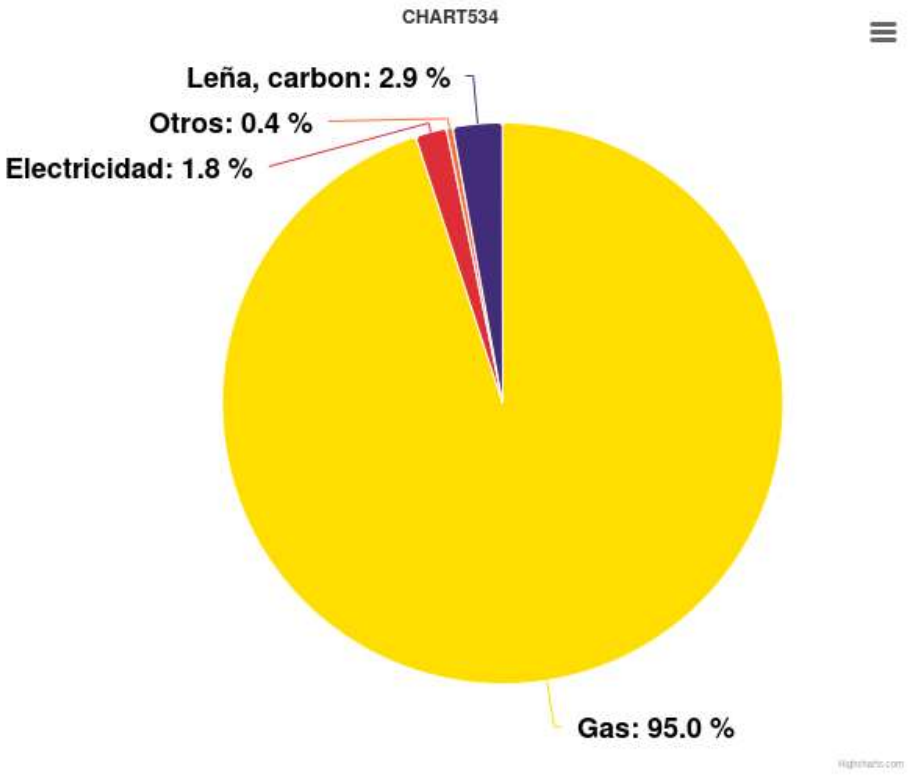
Limpiar el JSON en Angular:

A continuación, se limpia el JSON para eliminar los caracteres innecesarios y hacerlo más legible. Este proceso consiste en reemplazar ciertos caracteres en el JSON.

Generación de gráficos en Angular:

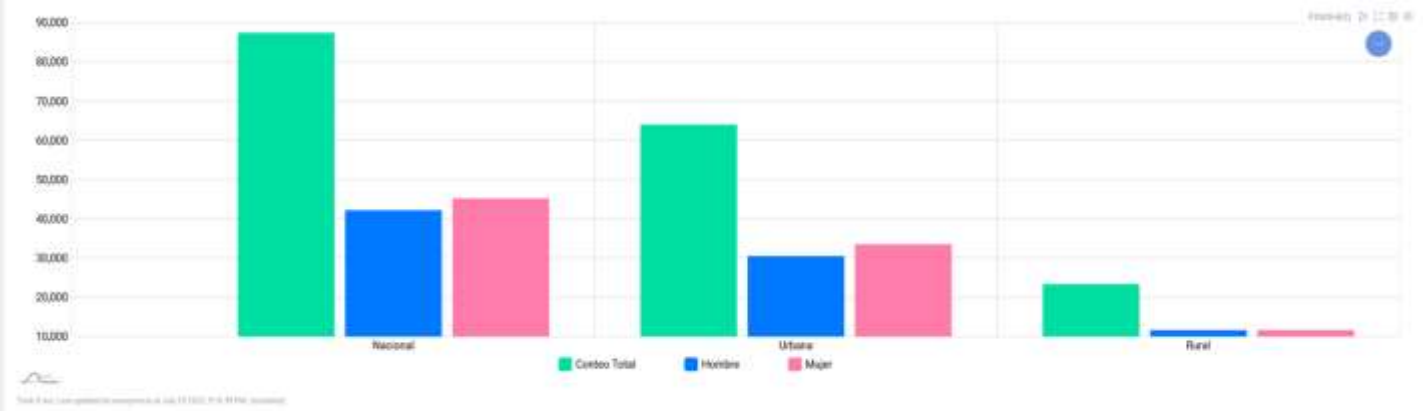
Una vez que los datos se han convertido a JSON y se han pasado a Angular, pueden ser utilizados para crear visualizaciones con bibliotecas de gráficos como amCharts y Highcharts. Estas bibliotecas ofrecen una variedad de gráficos interactivos y atractivos que pueden ser utilizados en informes y presentaciones para mostrar de forma efectiva los resultados del análisis de los datos.

Ejemplo:



Resultados obtenidos y visualizaciones con su análisis e interpretación

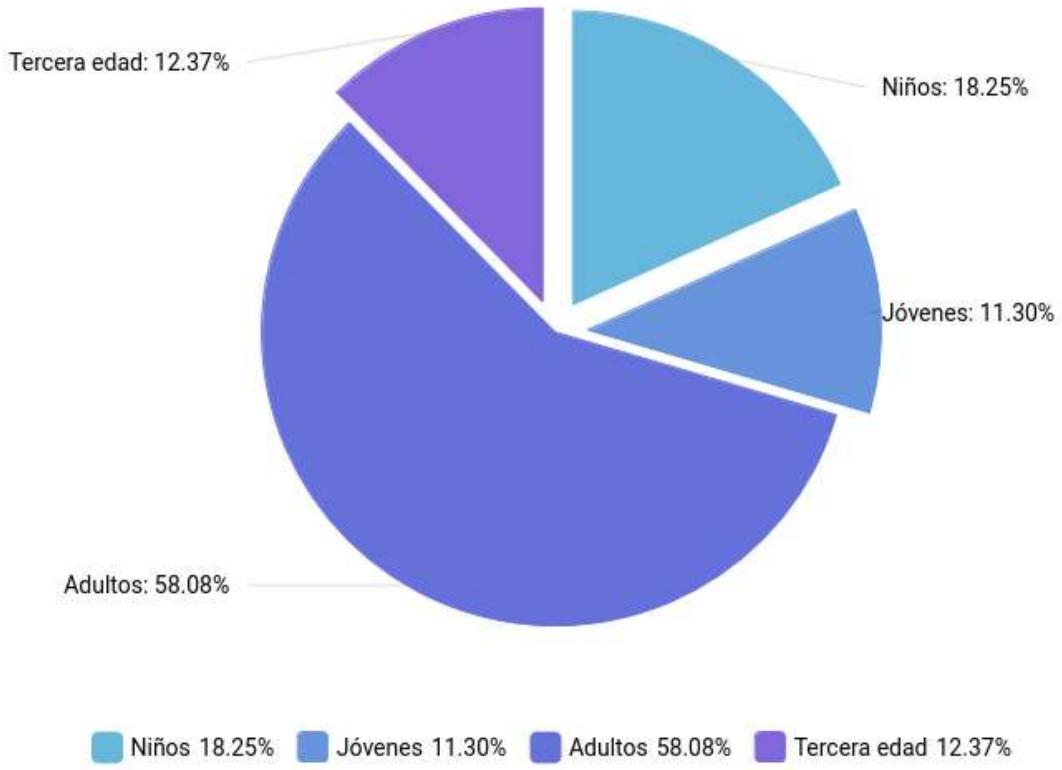
Numero de personas encuestadas separadas por áreas y por genero



A nivel nacional, la población total es de 87,399 individuos, divididos en 42,185 hombres y 45,214 mujeres. Esto sugiere una ligera predominancia femenina en la población total.
Cuando desglosamos los datos por área geográfica, vemos que en las áreas urbanas hay 64,037 individuos, de los cuales 30,520 son hombres y 33,517 son mujeres. Nuevamente, hay más mujeres que hombres en las áreas urbanas.

En contraste, en las áreas rurales, la población total es de 23,362 individuos, con una distribución casi igual entre hombres (11,665) y mujeres (11,697).

Edad de las personas encuestadas



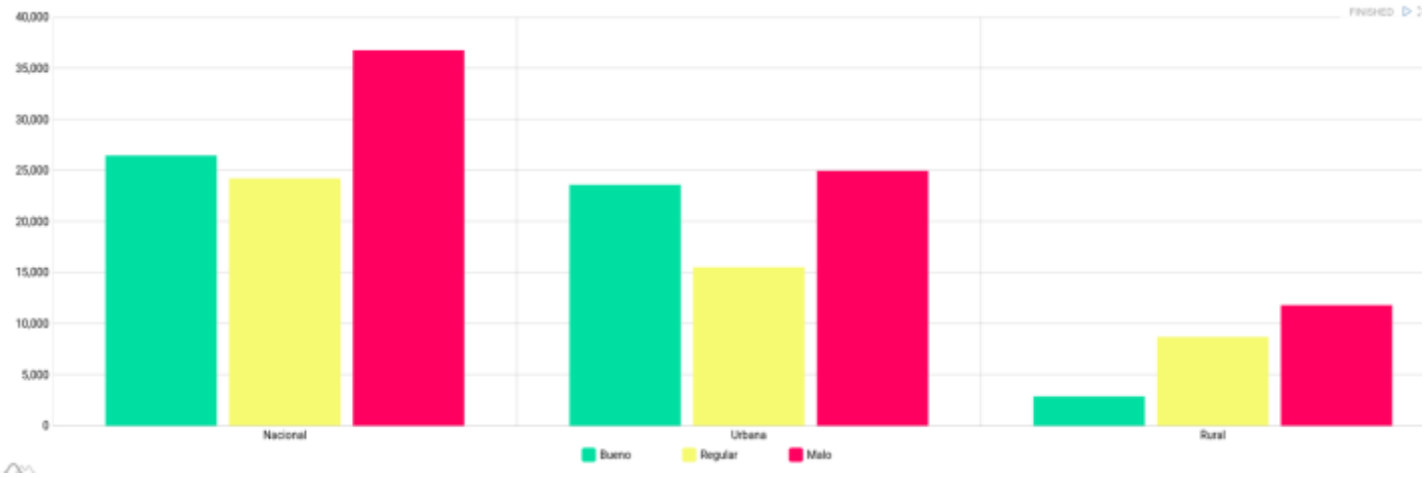
A nivel nacional, la población total es de 87,399 individuos. Estos se dividen en 15,953 niños, 9,874 jóvenes, 50,761 adultos y 10,811 personas de tercera edad.
La mayoría de la población se encuentra en la categoría de adultos con un total de 50,761, lo que indica que la población de este país está en su mayoría en su etapa productiva. Esto podría tener implicaciones en la economía del país, ya que más personas están en una etapa de su vida donde pueden trabajar y contribuir a la economía.

Los niños constituyen el segundo grupo más grande con 15,953, lo que puede indicar una alta tasa de natalidad en el país o un buen estado de salud que permite una alta supervivencia infantil.

Los jóvenes, que probablemente estén en su etapa de educación o inicio de su vida laboral, son 9,874. Este grupo de edad podría representar a los futuros trabajadores y líderes del país.

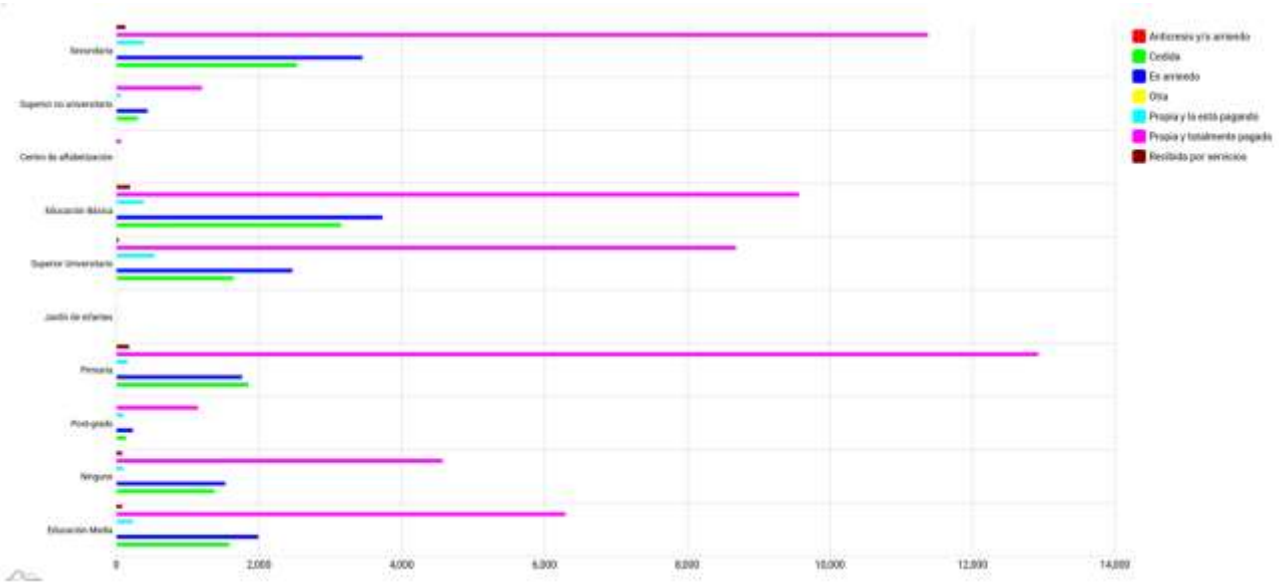
Finalmente, las personas de tercera edad representan a 10,811 individuos. Esta población es crítica ya que podrían requerir más atención médica y servicios de apoyo.

Estado de las viviendas en las diferentes areas



A nivel nacional, se registran 26,447 viviendas en estado "Bueno", 36,745 viviendas en estado "Malo", y 24,207 viviendas en estado "Regular". Esto sugiere que el país enfrenta desafíos en la calidad de la vivienda, ya que la mayor cantidad de las mismas cae en la categoría de "Malo".
Al desglosar los datos por áreas urbanas y rurales, se pueden observar diferencias notables. En las áreas urbanas, hay 23,576 viviendas en estado "Bueno", 24,944 en estado "Malo", y 15,517 en estado "Regular". Aunque la cantidad de viviendas "Buenas" y "Malas" es similar, es preocupante que la cantidad de viviendas "Malas" sea tan alta en las áreas urbanas, donde se espera una infraestructura de vivienda de mayor calidad.

En las áreas rurales, la situación es aún más desafiante. Se registran 2,871 viviendas en estado "Bueno", 11,801 en estado "Malo", y 8,690 en estado "Regular". Esto significa que la mayoría de las viviendas en las áreas rurales están en estado "Malo" o "Regular", lo que indica una disparidad en la calidad de la vivienda entre las áreas urbanas y rurales.



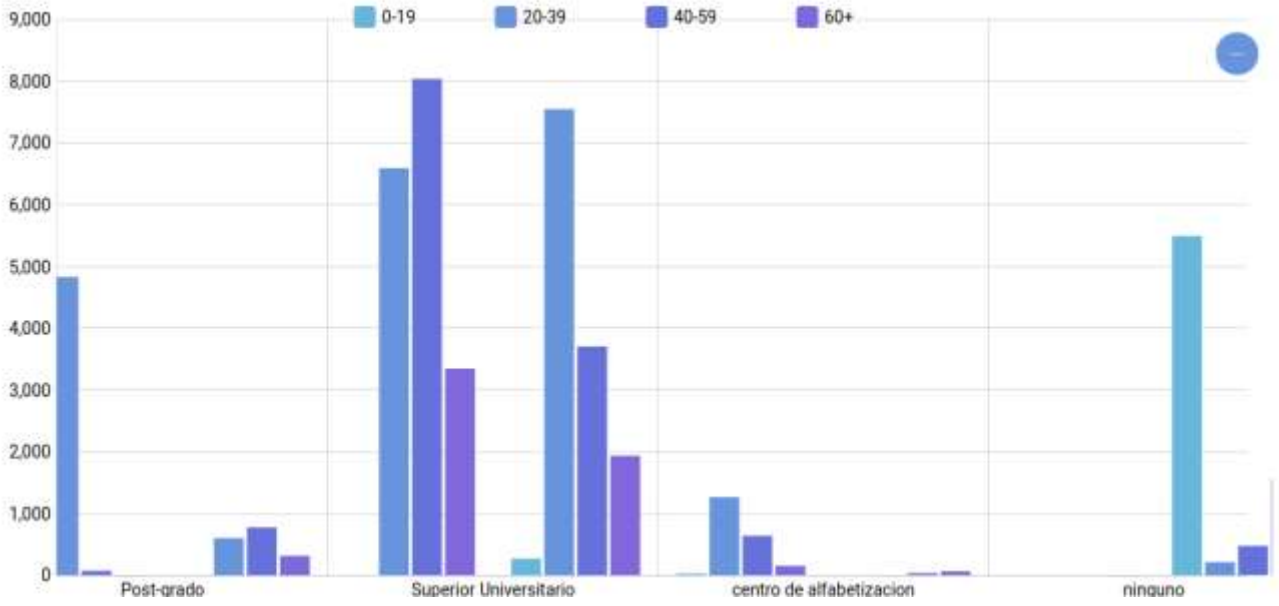
Este conjunto de datos muestra la relación entre el nivel de instrucción y la situación de la vivienda en Ecuador. A nivel general, se puede observar que, independientemente del nivel de instrucción, la mayoría de las personas tienen viviendas que están completamente pagadas. Sin embargo, hay diferencias significativas entre los diferentes niveles de instrucción.
En el caso de las personas con educación media, la mayoría posee su vivienda y la ha pagado completamente, seguido de las personas que tienen su vivienda en arriendo. Un pequeño porcentaje posee una vivienda que aún está pagando.

Las personas sin instrucción presentan una tendencia similar, aunque la cantidad de personas que poseen una vivienda propia y totalmente pagada es ligeramente menor en comparación con aquellos con educación media. También hay un número considerable de personas en esta categoría que viven en una vivienda cedida.

Para aquellos con postgrado, una gran mayoría posee una vivienda propia y totalmente pagada, seguido por las que están en arriendo y las que aún están pagando por su vivienda. El número de personas en esta categoría que viven en una vivienda cedida o recibida por servicios es relativamente bajo.

Los individuos con educación primaria muestran la mayor cantidad de viviendas propias y totalmente pagadas, seguido de las que están en arriendo y las cedidas.

El número de personas con una educación superior universitaria que poseen su vivienda y la han pagado completamente también es notablemente alto, seguido por las personas que están en arriendo y las que tienen su vivienda cedida.

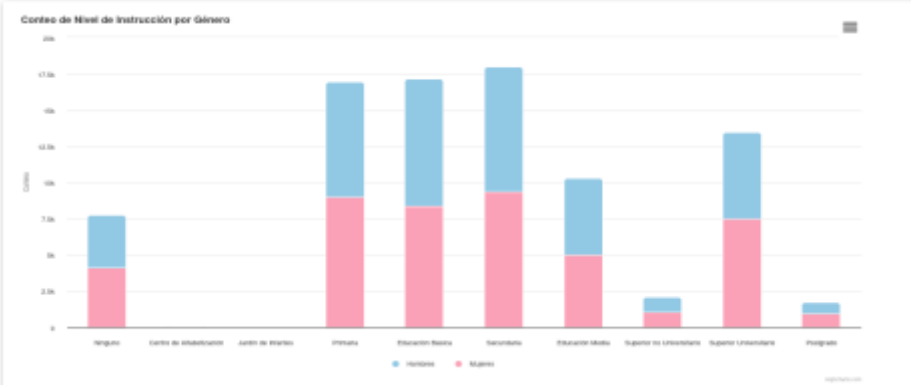


Rangos de edad por niveles de intstruccion

El análisis realizado muestra una distribución detallada de la población en diferentes rangos de edad en función de su nivel de instrucción. El objetivo principal es comprender cómo la educación varía entre distintos grupos de edad y detectar patrones significativos que puedan ayudar a informar políticas y decisiones relacionadas con la educación.

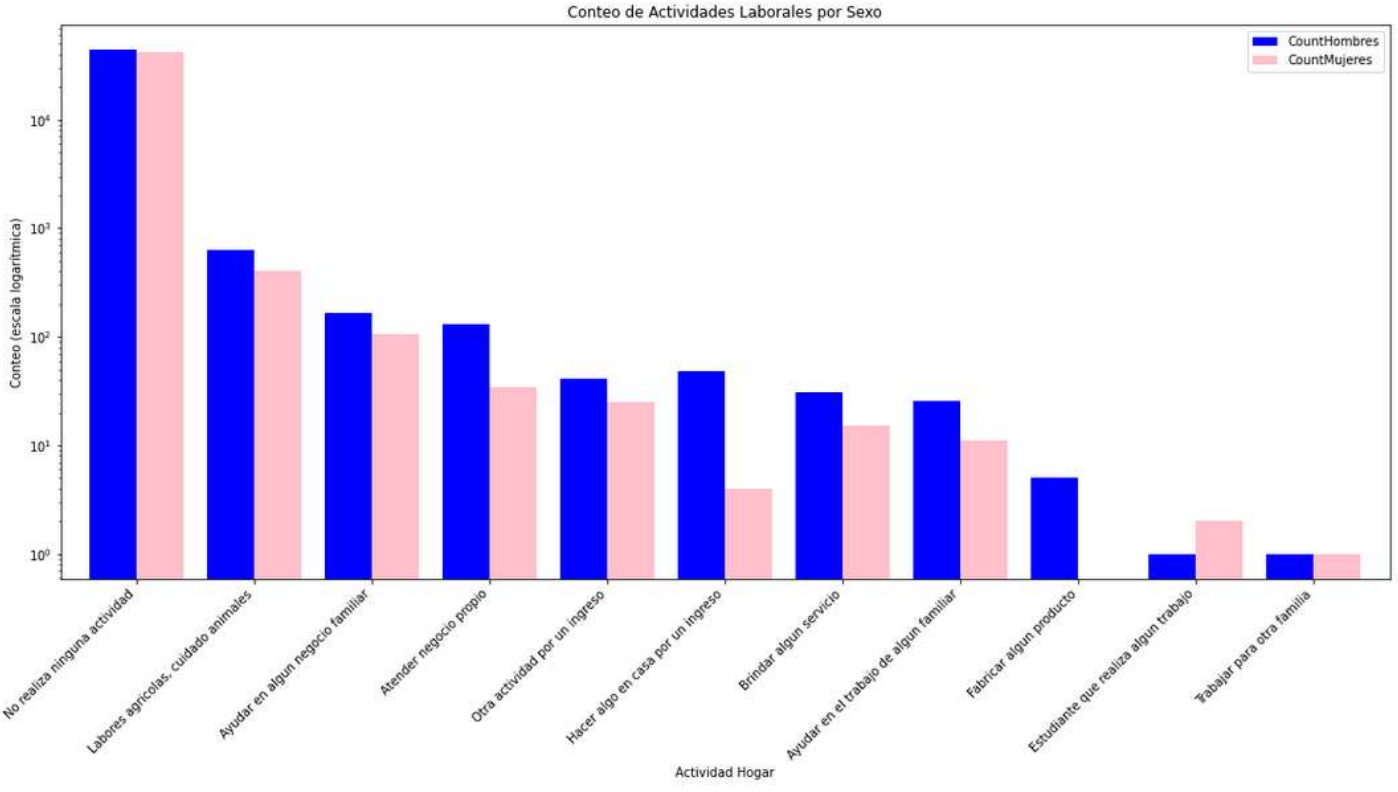
Para llevar a cabo el análisis, se recopilaron datos demográficos y educativos de una muestra representativa de la población. Los datos se agruparon en rangos de edad específicos, como "0-19 años", "20-39 años", "40-59 años" y "más de 60 años". Además, se clasificó el nivel de instrucción en diferentes categorías, como "sin educación", "educación primaria", "educación secundaria", "educación terciaria" y "educación superior".

Nivel de instrucción por genero



Hemos realizado un analisis enfocado en obtener la cantidad de personas en diferentes niveles de instrucción, agrupadas por sexo: Hombre y Mujer
Este análisis proporciona información valiosa sobre el nivel de educación de las personas según su género. Esto puede ayudar a comprender mejor la distribución de la educación en la población y detectar posibles desigualdades educativas entre hombres y mujeres.
Se muestra la cantidad de personas según su sexo (hombres o mujeres) y el nivel de instrucción que han alcanzado. Entonces, cada número representa la cantidad de personas en cada categoría de nivel de instrucción para hombres y mujeres.
Observaciones importantes a mencionar:
Los datos muestran que tanto hombres como mujeres tienen una cantidad significativa de personas con educación "Superior Universitario". La cifra es de 5982 hombres y 7473 mujeres. Esto indica que un número considerable de personas de ambos géneros ha obtenido títulos universitarios
Se observa que tanto hombres como mujeres tienen una participación más baja en el nivel de instrucción "Postgrado". La cifra es de 781 hombres y 910 mujeres. Esto sugiere que una cantidad menor de personas de ambos géneros ha continuado su educación después de obtener un título universitario.
Los datos muestran que hay una cantidad considerable de personas que no han completado su educación primaria, ya que tanto hombres como mujeres tienen una cantidad significativa de personas en el nivel "Ninguno" (3621 hombres y 4116 mujeres).

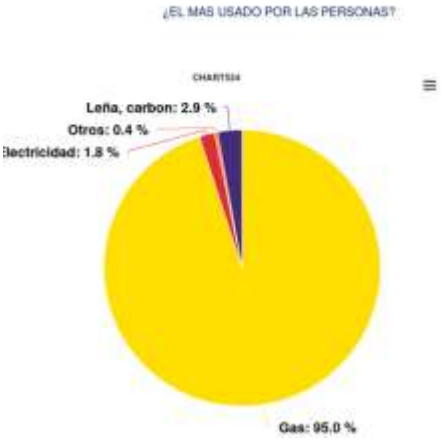
Conteo de actividad hogar por sexo



El análisis representa la distribución de género en diferentes actividades realizadas en el hogar.

Los datos presentados muestran la cantidad de hombres y mujeres involucrados en diversas tareas y actividades domésticas, lo que proporciona una visión detallada de cómo se distribuye la participación de género en el ámbito del hogar.
Según el análisis presentado, podemos observar que la participación de género en las actividades domésticas muestra una tendencia significativa hacia una mayor tasa de empleo en la población masculina. En comparación con las mujeres, el sexo masculino parece estar más involucrado en diversas tareas y actividades realizadas en el hogar.

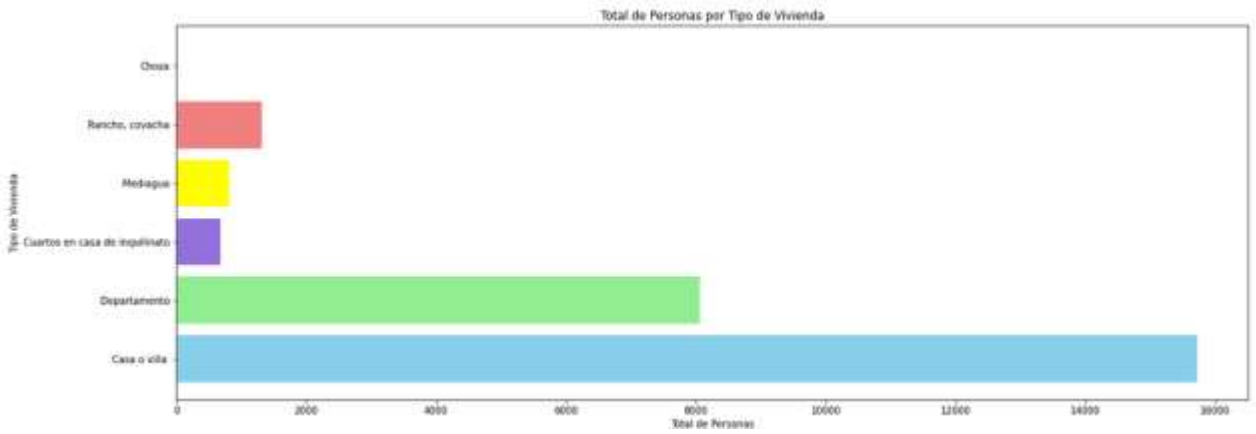
Materiales de cocina



Este análisis representa el consumo de diferentes fuentes de energía en dos áreas distintas: Urbana y Rural. Los datos están presentados en una gráfica de barras, donde cada barra representa una fuente de energía específica (Electricidad, Gas, Leña/Carbón y Otros) y su respectiva cantidad de consumo.
Observamos que en el área Urbana, el Gas es la fuente de energía más consumida, seguido por la Electricidad y la Leña/Carbón. Los "Otros" tipos de energía tienen una participación mínima en el consumo total.

Forma de obtencion de agua

La consulta permite identificar cual es la manera más comun de obtencion de agua en los hogares encuestados.
Esto es crucial para comprender el nivel de acceso de la población a servicios básicos, como baños, duchas o letrinas, lo que a su vez afecta la calidad de vida y la salud de las personas.



Total de personas por tipo de vivieda

1. La mayoría de las personas (15,716) viven en una "Casa o villa". Este tipo de vivienda es típicamente una residencia unifamiliar y parece ser la opción de vivienda más popular.
2. El siguiente tipo de vivienda más común es el "Departamento", con 8,056 personas. Los departamentos suelen estar ubicados en edificios residenciales y son especialmente comunes en las zonas urbanas.
3. El tercer tipo de vivienda más común es la "Rancho, covacha" con 1,317 personas. Este tipo de vivienda podría ser característico de las zonas rurales y podría reflejar condiciones de vida más precarias.
4. 807 personas viven en "Mediaguas", que son viviendas de emergencia, usualmente utilizadas tras desastres naturales.
5. 678 personas viven en "Cuartos en casa de vecinos", lo que podría sugerir un alto grado de hacinamiento en algunas áreas.
6. Solo 6 personas viven en "Choza". Esta cifra es bastante baja y puede reflejar un tipo de vivienda menos común, posiblemente asociado a niveles extremos de pobreza.

Conclusiones

La fusión de habilidades de programación, análisis de datos y el uso de una base de datos relevante es fundamental para maximizar el valor de los datos masivos disponibles en el conjunto de datos del INEC Ecuador. A través de herramientas y lenguajes de programación como Scala, Spark, Python, Angular y Pandas, somos capaces de manejar los datos de manera efectiva para extraer, transformar y cargar información, realizar análisis exhaustivos y presentar los resultados de manera visual. Este enfoque transversal nos permite no solo interpretar patrones y tendencias en los datos, sino también descubrir relaciones significativas entre diversas variables, como el tipo de vivienda y la cantidad de personas que habitan en ellas o la correlación entre el nivel de educación y el tipo de vivienda.

Este proyecto ha subrayado la relevancia de implementar técnicas de análisis de datos en la formulación de políticas públicas y en la planificación urbana. Los datos suministrados por INEC Ecuador constituyen una abundante fuente de información acerca de las condiciones de vida de los habitantes del país, y su análisis puede contribuir a detectar áreas que requieren atención y a priorizar recursos. Por ejemplo, nuestro análisis de los tipos de vivienda y la población que reside en cada una de ellas puede proporcionar orientación a los encargados de tomar decisiones sobre dónde es más necesario construir más viviendas o dónde concentrar los esfuerzos para mejorar las condiciones habitacionales. Además, al cotejar el nivel educativo y el tipo de propiedad de la vivienda, podemos obtener una visión más clara de la relación entre la educación y la seguridad en el hogar. En definitiva, estos hallazgos pueden ser utilizados para desarrollar estrategias de vivienda más justas y eficaces que se ajusten a las variadas necesidades de los habitantes de Ecuador.

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 Laboraly 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). ResultadosProvinciales: Loja. Recuperado de: https://www.ecuadorencifras.gob.ec/wp-content/descargas/Manu-lateral/Resultados-provinciales/loja.pdf

Apache Spark Developers. (n.d.). Apache Spark Documentation. Disponible en: https://spark.apache.org/docs/latest/

Apache Zeppelin. (n.d.). Apache Zeppelin Documentation 0.8.0. Disponible en: https://zeppelin.apache.org/docs/0.8.0/