

UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA



CARRERA DE COMPUTACION

Tema:

PROYECTO INTEGRADOR DE SABERES

Integrantes:

- Renny Steven Poma Medina
- Kevin Ramiro Cabrera Cabrera
- Gilmar Alexander Jiménez Medina

Docentes:

Mgtr. Ángel Eduardo Encalada Encalada

Mgtr. Jorge Afranio López Vargas

Asignaturas:

Prácticum 1.2

Base de Datos Avanzada

Programación Avanzada

Fecha de entrega:

31 de julio 2022

Periodo académico:

Abril 2022 - Agosto 2022

- **Introducción**

Se desarrollara un proyecto integrador que se llevara a cabo con dos asignaturas las cuales son base de datos y programación, gracias a los conocimientos adquiridos en cada una de estas materias es que elaboraremos este proyecto donde se lo realizara de manera grupal, donde se realizó varios avances hasta llegar a la fase final, también se aprendió a trabajar en grupo sabiendo así que es una manera muy importante, al respecto básicamente hablaremos de lo que trata nuestro proyecto, donde primeramente analizaremos un csv sobre el Catastro turístico de establecimientos a nivel nacional donde tenemos una fuente del ministerio de turismo donde tenemos una aproximación de 22209 registros donde a partir de aquí se realizó como primer avance una lluvia de ideas sobre el análisis que nos gustaría realizar con los datos relacionados al catastro turístico de establecimientos a nivel nacional, y posteriormente seleccionar los datos complementarios que agregaremos al conjunto de datos base describirlos y cuál es su propósito, también las fuentes de donde se obtuvieron esos datos posteriormente con base al estudio de los datos base, e identificación de los datos complementarios comenzamos a realizar lo que es el diseño conceptual considerando los datos base y los complementarios, también implementamos el diseño lógico relacional , generar el script DDL para la implementación de la base de datos en MySQL luego la preparación de datos para cargarlos a las tablas del esquema relacional ya completos y por ultimo implementación a la base de datos MySQL y cargar los datos. Generar un solo script SQL para la implementación y carga de la base de datos.

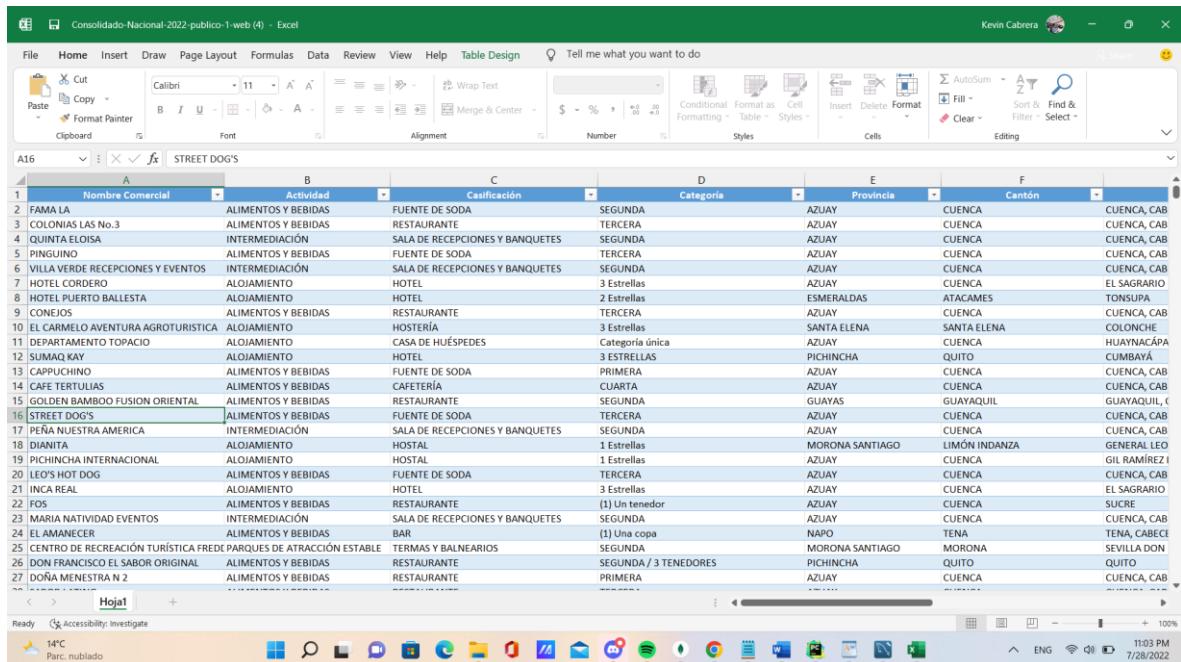
En parte de programación trabajaremos con una herramienta muy importante porque permite que los programadores escriban aplicaciones rápidamente en Java, Scala, Python, R y SQL para que puedan acceder a ellas los desarrolladores, los científicos de datos y los usuarios avanzados de las áreas de negocio con experiencia en estadísticas, puesto que esta herramienta era muy pesada gracias al Ingeniero nos facilitó una herramienta en la nube donde se podía trabajar con normalidad Zeppelin que era gitpod esta herramienta estaba relacionada con el GitHub, pues en la misma se tuvimos que ejecutar varios comando para que en el mismo se instale java y Scala, en concreto aquí pudimos colocar nuestro script de SQL e incluso nuestro csv donde en la misma herramienta se puede limpiar los datos se le puede hacer una lectura y almacenamiento de los datos también se puede mostrar las columnas cambiarlas de nombre borrar los espacios en blancos básicamente aquí generamos consultas y se puede visualizar lo solicitado con estadísticas

• Planificación del trabajo

Para la planificación de este proyecto se ha planteado como objetivo el poder dar a conocer a las personas las estadísticas de nuestro país en cuanto a los establecimientos o comerciales que lo conforman, según los datos facilitados por nuestros docentes de las materias de “Programación Avanzada” y “Base de Datos Avanzada”, materias las cuales con los conocimientos adquiridos de las mismas se trata de obtener datos limpios y concisos para su respectivo uso y manejo, permitiéndonos tener información de una manera más legible y óptima para su respectivo análisis.

• Fuentes de datos

Todos los datos con los que se trabajó fueron facilitados por el docente, datos que se encuentran en el siguiente enlace: <https://servicios.turismo.gob.ec/portfolio/catastro-turistico-nacional>.



The screenshot shows a Microsoft Excel spreadsheet titled "Consolidado-Nacional-2022-publico-1-web (4) - Excel". The table contains data from a CSV file with the following columns: Nombre Comercial, Actividad, Clasificación, Categoría, Provincia, and Cantón. The data includes various establishment names, their activities (like ALIMENTOS Y BEBIDAS, RESTAURANTE, HOTEL, etc.), their classification (like FUENTE DE SODA, SALA DE RECEPCIONES Y BANQUETES, etc.), categories (like SEGUNDA, TERCERA, etc.), provinces (like CUENCA, AZUAY, etc.), and cantons (like CUENCA, CAB, etc.). The table has 27 rows of data, starting from row 1 and ending at row 27. The Excel interface shows various toolbars and a ribbon at the top, and a taskbar at the bottom with icons for different applications like File Explorer, Edge, and Mail.

	A	B	C	D	E	F
1	Nombre Comercial	Actividad	Clasificación	Categoría	Provincia	Cantón
2	FAMA LA	ALIMENTOS Y BEBIDAS	FUENTE DE SODA	SEGUNDA	AZUAY	CUENCA
3	COLONIAS LAS No.3	ALIMENTOS Y BEBIDAS	RESTAURANTE	TERCERA	AZUAY	CUENCA, CAB
4	QUINTA ELOISA	INTERMEDIACIÓN	SALA DE RECEPCIONES Y BANQUETES	SEGUNDA	AZUAY	CUENCA, CAB
5	PINGUINO	ALIMENTOS Y BEBIDAS	FUENTE DE SODA	TERCERA	AZUAY	CUENCA, CAB
6	VILLA VERDE RECEPCIONES Y EVENTOS	INTERMEDIACIÓN	SALA DE RECEPCIONES Y BANQUETES	SEGUNDA	AZUAY	CUENCA, CAB
7	HOTEL CORDERO	ALOJAMIENTO	HOTEL	3 Estrellas	AZUAY	CUENCA
8	HOTEL PUERTO BALLESTA	ALOJAMIENTO	HOTEL	2 Estrellas	ESMERALDAS	ATACAMES
9	CONEJOS	ALIMENTOS Y BEBIDAS	RESTAURANTE	TERCERA	AZUAY	CUENCA, CAB
10	EL CARMELO AVENTURA AGROTURISTICA	ALOJAMIENTO	HOTERÍA	3 Estrellas	SANTA ELENA	SANTA ELENA
11	DEPARTAMENTO TOPACIO	ALOJAMIENTO	CASA DE HUÉSPEDES	Categoría única	AZUAY	COLONCHE
12	SUMAQ KAY	ALOJAMIENTO	HOTEL	3 ESTRELLAS	PICHINCHA	QUITO
13	CAPPUCHINO	ALIMENTOS Y BEBIDAS	FUENTE DE SODA	PRIMERA	AZUAY	CUMBAYÁ
14	CAFE TERTULIAS	ALIMENTOS Y BEBIDAS	CAFETERÍA	CUARTA	AZUAY	CUENCA, CAB
15	GOLDEN BAMBOO FUSION ORIENTAL	ALIMENTOS Y BEBIDAS	RESTAURANTE	SEGUNDA	GUAYAS	GUAYAQUIL, C
16	STREET DOG'S	ALIMENTOS Y BEBIDAS	FUENTE DE SODA	TERCERA	AZUAY	GUAYAQUIL
17	PEÑA NUESTRA AMERICA	INTERMEDIACIÓN	SALA DE RECEPCIONES Y BANQUETES	SEGUNDA	AZUAY	CUENCA, CAB
18	DIANITA	ALOJAMIENTO	HOTEL	1 Estrella	MORONA SANTIAGO	LIMÓN INDANZA
19	PICHINCHA INTERNACIONAL	ALOJAMIENTO	HOTEL	1 Estrellas	AZUAY	GENERAL LEO
20	LEO'S HOT DOG	ALIMENTOS Y BEBIDAS	FUENTE DE SODA	TERCERA	AZUAY	GIL RAMÍREZ I
21	INCA REAL	ALOJAMIENTO	HOTEL	3 Estrellas	AZUAY	CUENCA, CAB
22	FOS	ALIMENTOS Y BEBIDAS	RESTAURANTE	(1) Un tenedor	AZUAY	EL SAGRARIO
23	MARIA NATIVIDAD EVENTOS	INTERMEDIACIÓN	SALA DE RECEPCIONES Y BANQUETES	SEGUNDA	AZUAY	SUCRE
24	EL AMANECER	ALIMENTOS Y BEBIDAS	BAR	(1) Una copa	NAPO	CUENCA, CAB
25	CENTRO DE RECREACIÓN TURÍSTICA FREDE PARQUES DE ATRACCIÓN ESTABLE	ALIMENTOS Y BEBIDAS	TERMAS Y BALNEARIOS	SEGUNDA	MORONA SANTIAGO	TENA, CABECE
26	DON FRANCISCO EL SABOR ORIGINAL	ALIMENTOS Y BEBIDAS	RESTAURANTE	SEGUNDA / 3 TENEDORES	PICHINCHA	SEVILLA DON
27	DOÑA MENESTRA N 2	ALIMENTOS Y BEBIDAS	RESTAURANTE	PRIMERA	AZUAY	QUITO
						CUENCA, CAB

Se trabaja con CSV que contiene el catastro turístico de establecimientos a nivel nacional en todas las actividades turísticas.

Para poder trabajar de mejor manera se realizó variables las cuales consideramos que serían interesantes de trabajar en el data set proporcionado por el docente, dicha variable consta de poder añadir al data set proporcionado, los sitios turísticos que tiene el Ecuador por cada provincia, y para ello se investigó y se pudo obtener un data set el cual nos facilitó esta tarea el cual se lo puede encontrar en el siguiente enlace: <http://ambar.utpl.edu.ec/it/dataset/lugares-turistico/resource/ba3c85d5-bbc9-4f3c-a059-949532ae685b>

Nombre de la tabla: dataset_lugares_publicos_lugares

Filtros:

- Resumir con tabla dinámica
- Quitar duplicados
- Convertir en rango
- Insertar segmentación de datos
- Exportar Actualizar
- Abrir en el explorador
- Propiedades
- Desvincular
- Botón de encabezado
- Primera columna
- Fila de totales
- Última columna
- Filas con bandas
- Columnas con bandas
- Columnas con bandas
- Estilos de tabla

Consultas y cone..

Consultas | Conexiones

2 consultas

Hojas

Se cargaron 22.209 filas. 7 errores.

dataset-lugares-publicos-lug...

Se cargaron 114 filas.

Este implemento fue realizado con éxito donde solo se tomó los datos de los sitios turísticos por provincia.

También se lo logro implementar una variable más la cual trata de poder añadirle los Parques Nacionales a la data set proporcionado y al no ser datos muy extensos se lo realizo de manera manual quedándonos de esta manera:

Nombre de la tabla: ParquesNacionales

Filtros:

- De texto/CSV
- Fuentes recientes
- Consultas y conexiones
- Actualizar todo
- Propiedades
- Obtener datos
- De la web
- De una tabla o rango
- Obtener y transformar datos

Tipos de datos

Consultas y conexiones

Consultas | Conexiones

2 consultas

Hojas

Se cargaron 22.209 filas. 7 errores.

dataset-lugares-publicos-lug...

Se cargaron 114 filas.

• Componente de Base de Datos:

1. Estudio de los datos base del Catastro turístico de establecimientos a nivel nacional.
2. Luego se realizó una lluvia de ideas sobre el análisis que nos gustaría realizar con los datos relacionados al catastro turístico de establecimientos a nivel nacional donde las variables que escogimos son las siguientes:

Variable 1: Cantidad de turistas por provincia que ingresar anualmente.

El propósito es dar a conocer cómo se lleva una estadística de los ingresos anualmente es así para que el estado lleve el registro de cada una de las provincias y poder repotenciarlas.

Variable 2: Lugares recomendados turísticos por cada provincia.

El propósito es que da a conocer a las personas que por cada provincia existen muchos lugares turísticos en donde pueden visitar y así no asistan al mismo lugar ya que tenemos una gran diversidad en nuestro país.

Variable 3: Parques nacionales del ecuador

El propósito en cuanto a los parques nacionales de Ecuador es el de brindar y hacer saber sobre los parques que sean más accesibles y reconfortantes dependiendo de la persona, haciendo que tenga una mejor experiencia.

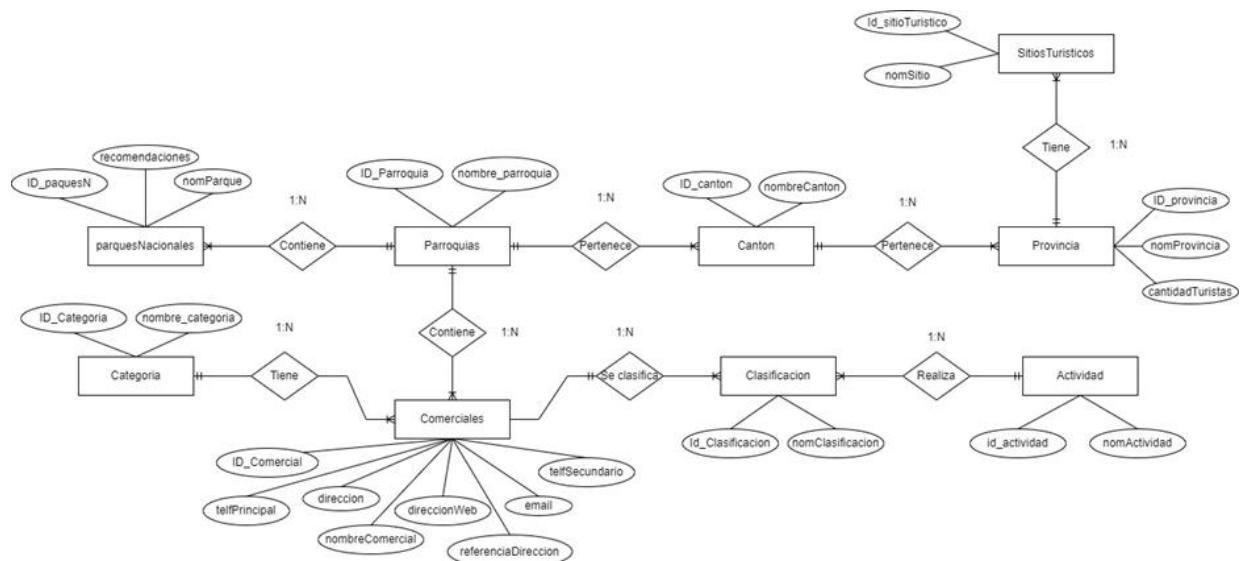
Variable 4: recomendaciones

El propósito de esta variable es dar a conocer si cada parque es recomendado o no, puesto que cada diferente turista tiene su punto de vista y hay veces que lugares terminan por no ser aceptados.

Variable 5: nombre de la actividad

El propósito es dar a conocer el nombre de la actividad para que así esta se relacione con cada uno de los comercios.

3. Luego realizamos el diseño conceptual de la base de datos (considerando los datos base y los datos complementarios).



DICCIONARIO DE DATOS

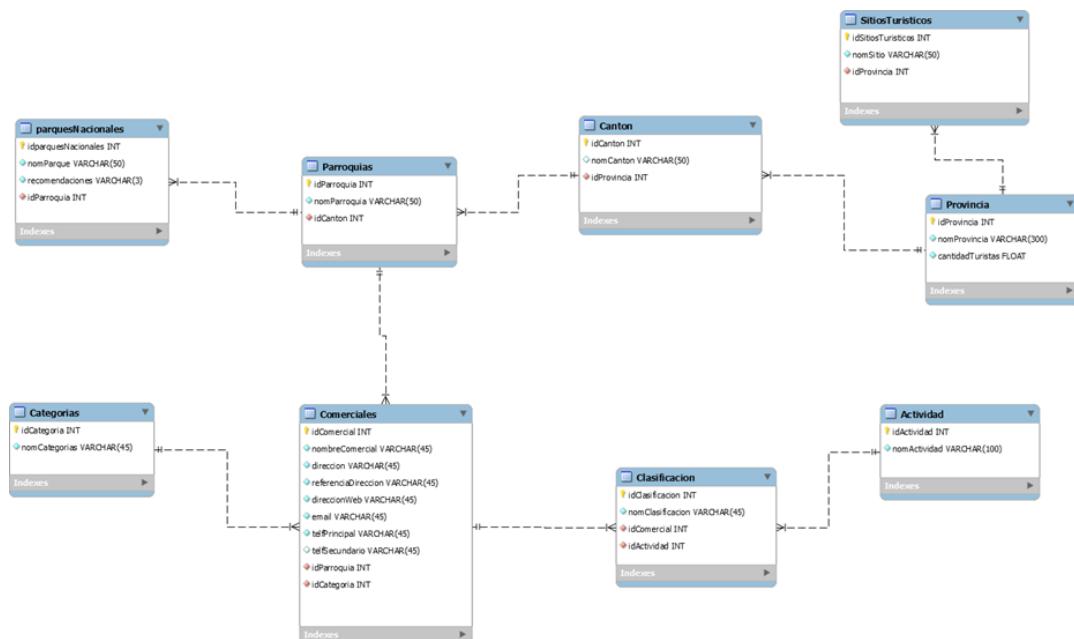
Entidad	Atributo	Dominio	Valor por defecto
Categorias	nomCategoria	Solo 3 valores posibles: “primero”, “segundo”, “tercero”	“primero”

Clasificación	nomClasificacion	Solo 6 valores posibles: “fuente de soda “,” restaurante”, “sala de recepciones y banquetes”, “Hostería”, “hotel”, “termas y balnearios”	“restaurante “
Actividad	nomActividad	Solo 6 valores posibles: “alimentos y bebidas” “Intermediación”, “alojamiento”, “Parques de atracción estable “	“Alimentos y bebidas”
ParquesNacionales	recomendaciones	Dos valores: “si”, “no”	“si”
Provincias	cantidadTuristas	Debe ser contado por año	

Otras restricciones de integridad:

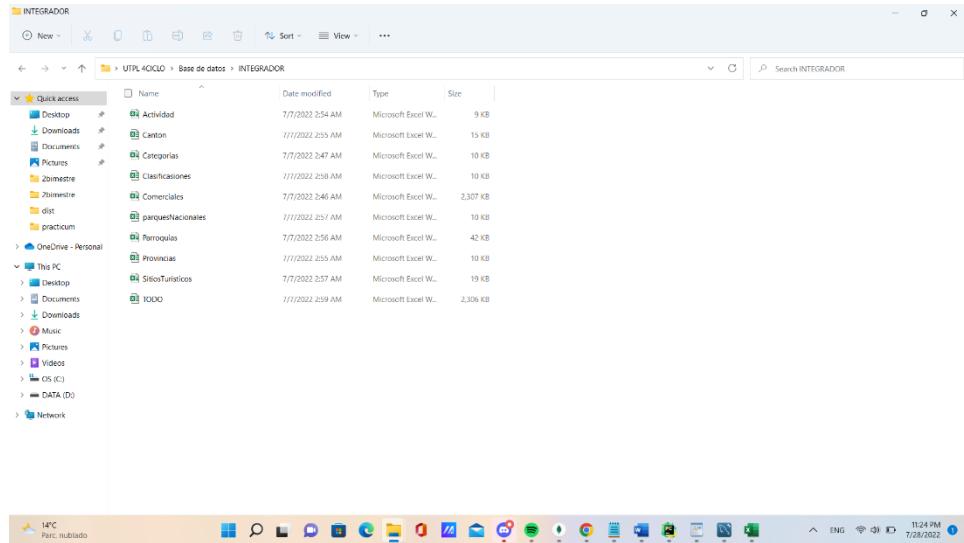
- Un comercial debería tener al menos una parroquia
- Un parque nacional debería tener al menos una provincia
- Cada provincia debe tener al menos un sitio turístico
- La provincia contiene a cantones y provincias
- Cada comercio debe tener una categoría y una actividad

4. Realizamos el Diseño lógico relacional gracias mySQL



5. Preparación de los datos para cargarlos a las tablas del esquema relacional obtenido en el punto anterior:

- Se deben obtener los datos complementarios e integrarlos a los datos base
Pueden usar Excel para este preprocesamiento.



- Aquí se tiene una hoja Excel por cada tabla de la base de datos

	A	B	C
	idSitoTuristico	idProvincia	nomMuseo
1	01	01	PARQUE NACIONAL CAJAS LLAVIUCU
2	02	01	CATEDRAL DE LA INMACULADA CONCEPCIÓN
3	03	01	PLAZA DE FLORES
4	04	01	MUSEO PUMAPUNGU
5	05	01	ARTESANÍAS EN PAÑA TOQUILLA CHORDELEG
6	06	01	ORFEBRERIA EN CHORDELEG
7	07	04	CAMPO DE BOLIVIA
8	08	01	CLEMENTERIO MUNICIPAL JOSÉ MARÍA AZELE FRANCO
9	09	04	GRUTA DE LA PAZ
10	10	04	IGLESIA DE LA VIRGEN DE LA CARIDAD
11	11	01	LAGUNA LA ENCARNACIÓN
12	12	01	MUSEO ARQUEOLÓGICO, DE ARTE MODERNO Y CONTEMPORÁNEO "GERMAN BASTIAMS VACA"
13	13	04	SAN JUAN Bautista LA CUMBRE DE LA PURIFICACIÓN
14	14	05	COLUMNA DE ISAMANI
15	15	05	HACIENDA SAN AGUSTÍN DEL CALLO
16	16	06	CASCADA YAGUAROCOCIA (PALATANZA)
17	17	06	NARIZ DEL DIABLO - ESTACIÓN DE SIBAMBE
18	18	06	LAGUNA MAGDALENA DE ATILLO
19	19	06	CASCADA CHI CHIPAY
20	20	06	LAGUNA DE LOS COCHES PUNAY
21	21	06	LAGUNA DE COSTA
22	22	06	MUSEO DE LA CIUDAD
23	23	06	NEVADO EL ALTAR
24	24	06	AGUAS TERMALES AGUALLANCHI
25	25	07	CAFÉ ZARUMA
26	26	07	FIESTA DE SEMANA SANTA EN EL CERRO EL CALVARIO

Aquí básicamente utilizamos la herramienta de Excel por muchas funcionalidades que nos ofrece la misma para así realizar un excelente trabajo referente a nuestro proyecto.

Una de las funciones muy importantes que nos ayudó a suprimir valores que se repiten es la función VLOOKUP, que significa "búsqueda vertical", puede ayudarlo a encontrar rápidamente los datos asociados con un determinado valor que ingrese. Por ejemplo, puede tener una tabla que contenga productos con ID y precios únicos. VLOOKUP puede mostrarle el precio de un determinado producto si ingresa su ID

A	B	C	
1	IdCanton	nomCanton	idProvincia
2	01	CUENCA	01
3	02	GIRÓN	01
4	03	GUAYACO	01
5	04	NARIÑO	01
6	05	PAUTE	01
7	06	PUCARA	01
8	07	SAN FERNANDO	01
9	08	SANTA ISABEL	01
10	09	SIGSIG	01
11	10	ÓÑA	01
12	11	CHORDELEG	01
13	12	EL PAN	01
14	13	SEVILLA DE ORO	01
15	14	GUACHAPALA	01
16	15	CAMILO PONCE ENRIQUEZ	01
17	16	GUARANDA	02
18	17	CHILLANES	02
19	18	CHIMBO	02
20	19	CHOCOLIA	02
21	20	SAN MIGUEL	02
22	21	CALUMA	02
23	22	LAS NAVES	02
24	23	AOZOGUE	03
25	24	BIBIÁN	03
26	25	CASAR	03
27	26	LA TRONCAL	03

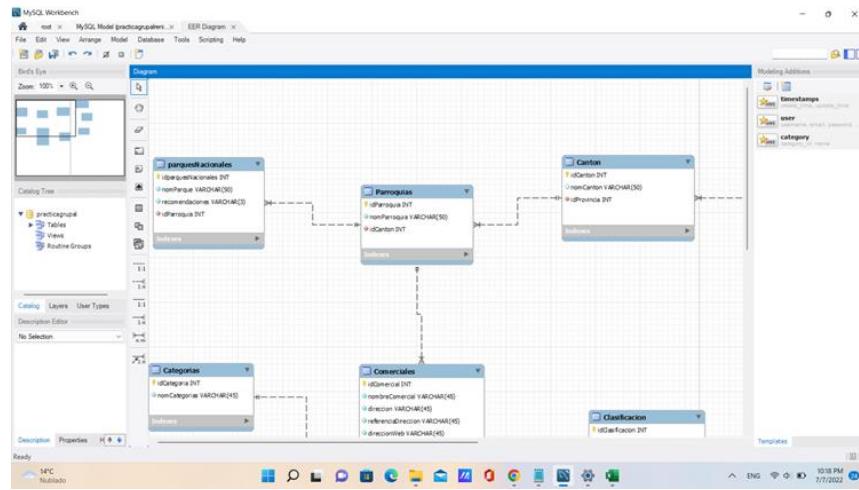
Buscar y remplazar: Resulta una herramienta ideal para estructurar claramente los valores, especialmente si son numerosos. Sin embargo, cuando los archivos incluyen muchos datos, pueden volverse un poco confusos. En este caso, si necesitas encontrar determinadas combinaciones de letras o números y sustituirlas, puedes utilizar las funciones de buscar y reemplazar en Excel, que te permiten hacerlo con solo un clic del ratón.

D	E	F
ReferenciaDireccion	direccionWeb	telfPrincipal
2	null	2843739
3	null	2804790
4	null	72829459
5	null	2810586
6	null	983123255
7	JUNTO A LA IGLESIA DE SAN ALFONSO O DIAGONAL AL DISPENSARIO CENTRAL DEL IEES	null
8	HOTEL PUERTO BALLESTA	www.puertoballesta.com
9	null	2800410
10	null	042786693
11	DETРАS DE LA IGLESIA DEL VERGEL, A UNA CUADRA DE LA ESCUELA CARLOS ZAMBRANO	null
12	PISO: SN. SECTOR: FRENTE AL SCALA SHOPPING	www.sumaqkay.com
13	null	2843782
14	null	991610014
15	null	43069305
16	null	72834318
17	null	72801656
18	null	072770122
19	A MEDIA CUADRA DE LA CLINICA BOLIVAR	null
20	null	072833695
21	null	2880591
22	null	072823636
23	null	null
24	null	72879342
25	null	02311047
26	A UNA CUADRA DEL C.C. HNO MIGUEL	null
27	null	994757103

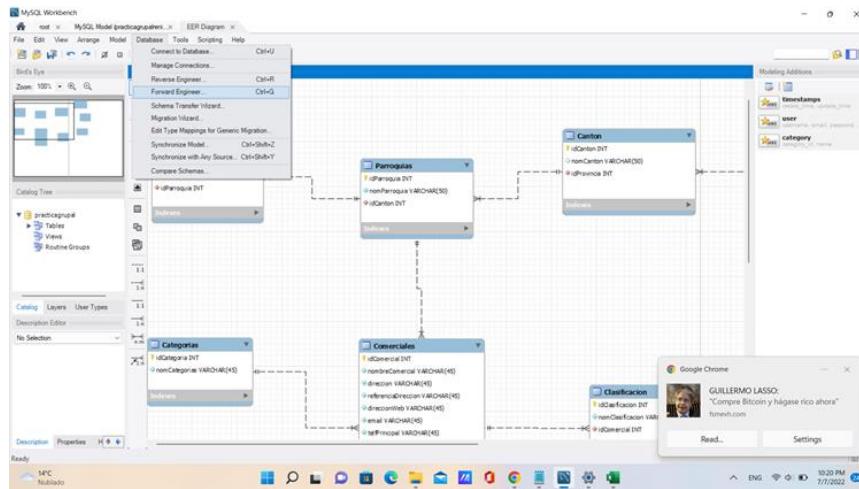
Como se la realiza. - La función Buscar es la herramienta ideal para ello. Para abrirla, selecciona la pestaña de “Inicio” del menú y, después, abre el menú desplegable de “Buscar y seleccionar”. Pulsa “Buscar” para abrir la ventana, donde encontrarás todas las opciones.

Carga de los datos

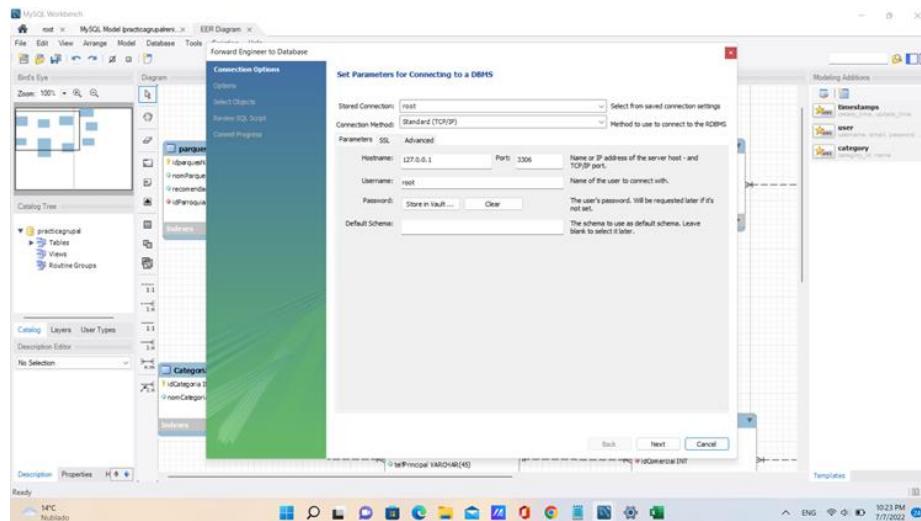
1. Tenemos que realizar el modelo lógico gracias a la herramienta MySQL Workbench, donde aquí mismo nos brinda la creación del script con todas sus tablas y relaciones.



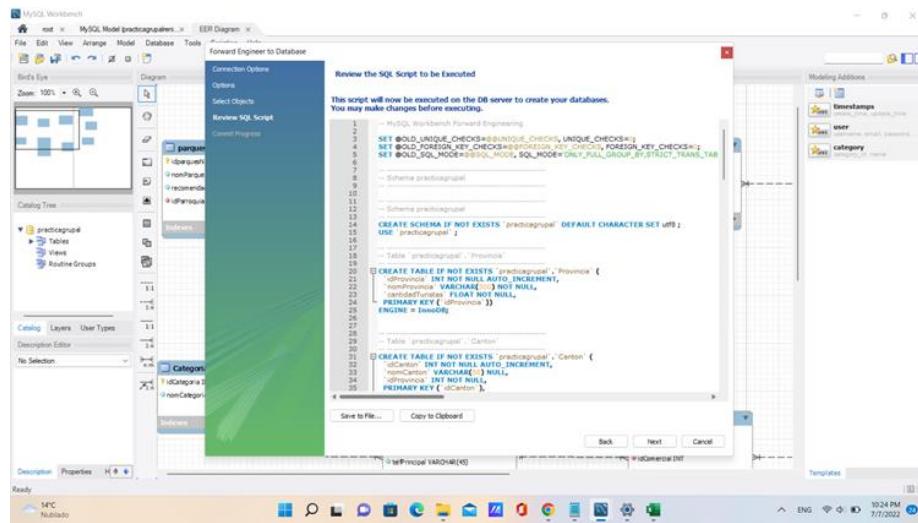
2. Luego una vez terminado con el modelo lógico procedemos a generarla nos vamos a la parte superior donde nos dice database y elegimos la opción Forward Enginner



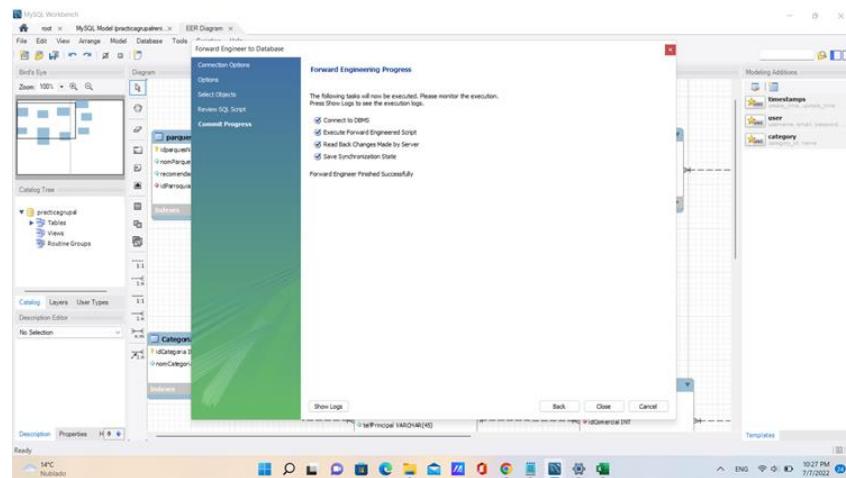
3. Luego nos aparecerá una ventana donde nos conectará directo al root



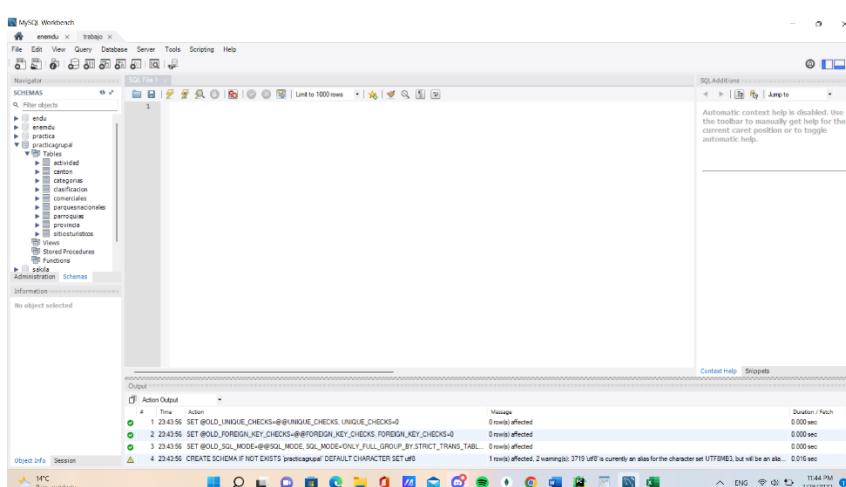
4. Luego colocamos next hasta la ventana donde podremos observar nuestro script que se ha generado a base del diseño lógico



5. Colocamos next una vez más y verificamos que este correctamente las opciones que nos salen en pantalla es así que podremos decir que esta correctamente generado y seleccionamos close.



6. Así verificamos que en nuestro usuario se hayan creado correctamente las tablas hacia el lado izquierdo



7. Aquí posteriormente Se realizo una función concatenar las celdas para generar los insert que nos permita realizar la carga de los datos de cada tabla, en lo cual nos ayuda obtener la estructura para cargar a la base de datos.

The screenshot shows an Excel spreadsheet with data in columns A, B, and C. The formula bar displays the following SQL query:

```
INSERT INTO `Canton` ('idCanton', 'nomCanton', 'idProvincia') VALUES
```

The data in the table includes:

	A	B	C
1	A	CANTON	1
2	2	CUENCA	1
3	3	BAJAS	1
4	4	GUALACEO	1
5	5	NASON	1
6	6	PAN	1
7	7	PUCARA	1
8	8	SAN FERNANDO	1
9	9	SANTA ISABEL	1
10	10	SIGIS	1
11	11	ORIA	1
12	12	CHONDELEG	1
13	13	PAN	1
14	14	SEVILLA DE ORO	1
15	15	GUACHAPALA	1
16	16	CARLOS ALFONSE ENRIQUEZ	2
17	17	GUARANAO	2
18	18	CHILLANES	2
19	19	CHIRIBON	2
20	20	ECHENANDA	2
21	21	SAN MIGUEL	2
22	22	LAS NAVES	2
23	23	ADOGUES	3
24	24	EL PAN	3
25	25	CARABA	3
26	26	LA TRONCAL	3
27	27	EL TAMBO	3
28	28	EL TAMB	3
29	29	TUSCAL	3
30	30	TULCAN	4
31	31	BOQUILLAS	4

8. Luego proseguimos a insertar cada uno de los datos que corresponden a cada tabla con sus respectivos atributos.

The screenshot shows the MySQL Workbench interface with an open SQL file. The code in the editor is:

```
INSERT INTO `canton` ('idCanton', 'nomCanton', 'idProvincia') VALUES
```

The code continues with a large list of values, each starting with a number from 1 to 31, followed by a name and a value of 1 or 2.

The screenshot shows the MySQL Workbench interface with an open SQL file. The code in the editor is:

```
INSERT INTO `parroquia` ('idParroquia', 'nomParroquia', 'idCanton') VALUES
```

The code continues with a large list of values, each starting with a number from 1 to 343, followed by a name and a value of 1 or 3.

The screenshot shows the MySQL Workbench interface. The top menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, Help, and EER Diagrams. The left sidebar displays the database structure under the Schemas tab, showing tables like 'practicas' and 'practicasgrupal'. The main area is divided into two panes: 'EER Diagram' on the left and 'SQL Editor' on the right. The SQL Editor contains the following SQL code:

```
1088
1089 INSERT INTO `categorias` ('idCategoria', 'nomCategorias') VALUES
1090 ('1', 'SEGUNDA');
1091 ('2', 'TERCERA');
1092 ('3', 'UNICA');
1093 ('4', 'PRIMERA');
1094 ('5', 'QUATRA');
1095 ('6', 'LOCALES');
1096 ('7', 'AGENCIA DE VIAJES INTERNACIONAL');
1097 ('8', 'AGENCIA DE VIAJES DUAL');
1098 ('9', 'SERVICIO TRANSP.TERRAS.TURIS.');
1099 ('10', 'GARAJONAY TURISTICO');
1100 ('11', 'AGENCIAS A CARGO');
1101 ('12', 'AGENCIA DE VIAJES MAYORISTA');
1102 ('13', 'LAN.TOUR DISABIL');
1103 ('14', 'LANE_PASAJER');
1104 ('15', 'AGENCIAS DE VENTA');
1105 ('16', 'NOTORIAS');
1106 ('17', 'SERVICIO NACIONAL AIREO');
1107 ('18', 'ENTER_OPEN');
1108 ('19', 'AGENCIAS NO OPERANTE EN EL PAIS QUE TIENE OFICINAS DE REPRESENTACION O INFORMACION');
1109 ('20', 'NOTO_Velero');
1110 ('21', 'SERVICIO AVIONETAS HELICOPTEROS');
1111 ('22', 'FLUVIAL');
1112 ('23', 'FERIA');
1113 ('24', 'INTERNACIONAL NO OPERANTE EN EL PAIS QUE TIENE OFICINAS DE VENTA');
1114 ('25', 'FUNDICULARES O TELEFERICOS POR CORDILLAS');
```

9. Luego realizamos una consulta para verificar que se hayan ingresado correctamente los datos en que este caso con la tabla provincias.

The screenshot shows the MySQL Workbench interface. In the left sidebar, under the 'practicagrupal' schema, the 'provincias' table is selected. The main pane displays a result grid for the query: 'SELECT * FROM practicagrupal.provincia;'. The grid has columns: idProvincia, nombreProvincia, and cantidadEstas. The data includes 26 rows of provincial information. The right side of the interface shows the SQL editor with the same query and some help text.

	idProvincia	nombreProvincia	cantidadEstas
1	AZUAY	42.658	
2	BOLÍVAR	22.333	
3	CAÑAR	1.771	
4	CARCHÉ	9.083	
5	COTOPAXI	21.879	
6	CHIMBORAZO	28.486	
7	EL ORO	24.091	
8	ESMERALDAS	35.316	
9	GUAYAS	86.774	
10	IDA VACA	36.744	
11	LOJA	24.327	
12	LOS RÍOS	3.82	
13	MANABÍ	140.488	
14	MORENO SAMBOANGA	10.953	
15	NAPO	54.456	
16	PASTAZA	31.113	

• Componente de Programación Avanzada

Para la asignatura de programación avanzada la herramienta principal que se utilizó para el desarrollo del proyecto es apache zeppelin y también se utilizó librerías y dependencias.

Apache Zeppelin

Es un notebook basado en una aplicación web, que sirve para hacer documentos, que sirven para hacer análisis de datos de manera interactiva, ejecutar código, mostrar visualizaciones, etc. En particular, Zeppelin tiene soporte nativo para Spark (ya sea localmente o conectándose a un clúster).

Si bien Zeppelin no es necesario para el trabajo con Spark (de hecho, a la hora de programar un pipeline de Spark es mejor pensarla como una aplicación autocontenido), la verdad es que a la hora de hacer análisis exploratorio de datos, o escribir prototipos que sean fáciles de probar (quick and dirty), un notebook (no solo Zeppelin, también Jupyter también se puede configurar para trabajar con Scala y Spark) es una herramienta muy útil.

The screenshot shows the Zeppelin web interface. The top navigation bar includes 'Zeppelin', 'Notebook', 'Job', 'Search', and 'anonymous'. The main content area features a large 'Welcome to Zeppelin!' heading, followed by a brief introduction: 'Zeppelin is web-based notebook that enables interactive data analytics. You can make beautiful data-driven, interactive, collaborative document with SQL, code and even more!'. On the left, there's a 'Notebook' sidebar with options like 'Import note', 'Create new note', and a 'Filter' dropdown. On the right, there's a 'Help' section with links to documentation, a 'Community' section encouraging contributions, and links to mailing lists, issues tracking, and GitHub. A large decorative graphic of a hot air balloon is positioned on the right side of the main content area.

¿Qué es Apache Spark?

Spark es una librería, escrita originalmente en Scala (un lenguaje muy elegante de programación funcional con diseño orientado a objetos, les recomiendo revisarlo si no lo hicieron aún), cuyo objetivo es servir de motor para hacer análisis y procesamiento de datos a gran escala mediante cómputo distribuido. Tiene la gran ventaja de poder utilizarse con diversos lenguajes (Scala, Python, R, Java y SQL) y correr sobre diversos sistemas (e.g., EC2, Hadoop YARN, Mesos, Kubernetes) y poder obtener datos de diversas fuentes (e.g. HDFS, Cassandra, HBase, Hive).

Comenzamos revisando el DataFrame del Catastro en lo cual lo podemos encontrar en el siguiente enlace:

<https://github.com/alexander-jimenez11/Proyecto-Integrador/blob/main/Programacion%20avanzada/Consolidado-Nacional-2022-publico-1-web.xlsx>.

Primero realizamos la carga y limpieza de los datos:

1. Lectura y almacenamiento del archivo Excel de la página del ministerio del turismo en un DataFrame

```
val df = spark.read
  .format("com.crealytics.spark.excel")
  .option("header", "true")
  .option("inferSchema", "true")
  .load("https://servicios.turismo.eob.ec/descargas/Catastro-turistico/Consolidado-Nacional-2022-publico-1-web.xlsx")

df: org.apache.spark.sql.DataFrame = [Nombre Comercial: string, Actividad: string ... 11 more fields]
```

2. Se muestra las columnas que contiene el DataFrame

```
df.printSchema()

root
|-- Nombre Comercial: string (nullable = true)
|-- Actividad: string (nullable = true)
|-- Clasificación: string (nullable = true)
|-- Categoría: string (nullable = true)
|-- Provincia: string (nullable = true)
|-- Cantón: string (nullable = true)
|-- Parroquia: string (nullable = true)
|-- Referencia de Dirección: string (nullable = true)
|-- Dirección: string (nullable = true)
|-- Teléfono Principal: string (nullable = true)
|-- Teléfono Secundario: string (nullable = true)
|-- Correo Electrónico: string (nullable = true)
|-- Dirección Web: string (nullable = true)
```

3. Información del DataFrame

Nombre Comercial	Actividad	Clasificación	Categoría	Provincia	Cantón	Parroquia	Referencia de Dirección	Dirección Web
FAMA LA	ALIMENTOS Y BEBIDAS	FUENTE DE SODA	SEGUNDA	AZUAY	CUENCA	CUENCA, CABECERA CANTONAL Y CAPITAL PROVINCIAL	null	BO
COLONIAS LAS No.3	ALIMENTOS Y BEBIDAS	RESTAURANTE	TERCERA	AZUAY	CUENCA	CUENCA, CABECERA CANTONAL Y CAPITAL PROVINCIAL	null	AV. HU ME
QUINTA ELOISA	INTERMEDIACIÓN	SALA DE RECEPCIONES Y BANQUETES	SEGUNDA	AZUAY	CUENCA	CUENCA, CABECERA CANTONAL Y CAPITAL PROVINCIAL	null	JO PE FE AR
PINGUINO	ALIMENTOS Y BEBIDAS	FUENTE DE SODA	TERCERA	AZUAY	CUENCA	CUENCA, CABECERA CANTONAL Y CAPITAL PROVINCIAL	null	RE 87

4. Se renombra todas las columnas para no tener problemas con las palabras mayúsculas y signos ortográfico auxiliar diacrítico.

```
val df1 = df
    .withColumnRenamed("Nombre Comercial", "nombre")
    .withColumnRenamed("Actividad", "clasificacion")
    .withColumnRenamed("Casificación", "subclasificacion")
    .withColumnRenamed("Categoría", "categoria")
    .withColumnRenamed("Cantón", "canton")
    .withColumnRenamed("Dirección", "direccion")
    .withColumnRenamed("Referencia de Dirección", "dir_referencia")
    .withColumnRenamed("Teléfono Principal", "telf_principal")
    .withColumnRenamed("Teléfono Secundario", "telf_secundario")
    .withColumnRenamed("Correo Electrónico", "email")
    .withColumnRenamed("Dirección Web", "web")
```

```
df1: org.apache.spark.sql.DataFrame = [nombre: string, clasificacion: string ... 11 more fields]
```

5. Visualizamos los nombres de las columnas renombradas

```
df1.printSchema()
root
|-- nombre: string (nullable = true)
|-- clasificacion: string (nullable = true)
|-- subclasificacion: string (nullable = true)
|-- categoria: string (nullable = true)
|-- Provincia: string (nullable = true)
|-- canton: string (nullable = true)
|-- Parroquia: string (nullable = true)
|-- dir_referencia: string (nullable = true)
|-- direccion: string (nullable = true)
|-- telf_principal: string (nullable = true)
|-- telf_secundario: string (nullable = true)
|-- email: string (nullable = true)
|-- web: string (nullable = true)
```

6. Se hace la eliminación de los espacios en el inicio y fin de los strings.

```
val df2 = df1
    .withColumn("nombre", trim($"nombre"))
    .withColumn("clasificacion", trim($"clasificacion"))
    .withColumn("subclasificacion", trim($"subclasificacion"))
    .withColumn("categoria", trim($"categoria"))
    .withColumn("provincia", trim($"Provincia"))
    .withColumn("canton", trim($"canton"))
    .withColumn("parroquia", trim($"Parroquia"))
    .withColumn("direccion", trim($"direccion"))
    .withColumn("dir_referencia", trim($"dir_referencia"))
    .withColumn("telf_principal", trim($"telf_principal"))
    .withColumn("telf_secundario", trim($"telf_secundario"))
    .withColumn("email", trim($"email"))
    .withColumn("web", trim($"web"))
```

```
df2: org.apache.spark.sql.DataFrame = [nombre: string, clasificacion: string ... 11 more fields]
```

7. Se renombra la columna categorías.

```

val df3 = df2.withColumn("categoria",
    when(upper($"categoria") === "(5) CINCO TENEDORES", "5 TENEDORES")
    .when(upper($"categoria") === "LUJO / 5 TENEDORES", "5 TENEDORES")
    .when(upper($"categoria") === "(4) CUATRO TENEDORES", "4 TENEDORES")
    .when(upper($"categoria") === "PRIMERA / 4 TENEDORES", "4 TENEDORES")
    .when(upper($"categoria") === "PRIMERA" && $"subclasificacion" === "RESTAURANTE", "4 TENEDORES")
    .when(upper($"categoria") === "(3) TRES TENEDORES", "3 TENEDORES")
    .when(upper($"categoria") === "SEGUNDA / 3 TENEDORES", "3 TENEDORES")
    .when(upper($"categoria") === "SEGUNDA" && $"subclasificacion" === "RESTAURANTE", "3 TENEDORES")
    .when(upper($"categoria") === "(2) DOS TENEDORES", "2 TENEDORES")
    .when(upper($"categoria") === "TERCERA / 2 TENEDORES", "2 TENEDORES")
    .when(upper($"categoria") === "TERCERA" && $"subclasificacion" === "RESTAURANTE", "2 TENEDORES")
    .when(upper($"categoria") === "(1) UN TENEDOR", "1 TENEDOR")
    .when(upper($"categoria") === "CUARTA / 1 TENEDOR", "1 TENEDOR")
    .when(upper($"categoria") === "CUARTA" && $"subclasificacion" === "RESTAURANTE", "1 TENEDOR")
    .when(upper($"categoria") === "(3) TRES COPAS", "3 COPAS")
    .when(upper($"categoria") === "PRIMERA / 3 COPAS", "3 COPAS")
    .when(upper($"categoria") === "PRIMERA" && $"subclasificacion" === "BAR" || $"subclasificacion" === "DISCOTECA", "3 COPAS")
    .when(upper($"categoria") === "(2) DOS COPAS", "2 COPAS")
    .when(upper($"categoria") === "SEGUNDA / 2 COPAS", "2 COPAS")
    .when(upper($"categoria") === "SEGUNDA" && $"subclasificacion" === "BAR" || $"subclasificacion" === "DISCOTECA", "2 COPAS")
    .when(upper($"categoria") === "(1) UNA COPA", "1 COPA")
    .when(upper($"categoria") === "TERCERA / 1 COPA", "1 COPA")
    .when(upper($"categoria") === "TERCERA" && $"subclasificacion" === "BAR" || $"subclasificacion" === "DISCOTECA", "1 COPA")
    .when(upper($"categoria") === "CUARTA" && $"subclasificacion" === "CAFETERÍA", "4 TAZAS")
    .when(upper($"categoria") === "TERCERA" && $"subclasificacion" === "CAFETERÍA", "3 TAZAS")
    .when(upper($"categoria") === "(2) DOS TAZAS", "2 TAZAS")
    .when(upper($"categoria") === "LUJO / 2 TAZAS", "2 TAZAS")
    .when(upper($"categoria") === "SEGUNDA" && $"subclasificacion" === "CAFETERÍA", "2 TAZAS")
    .when(upper($"categoria") === "(1) UNA TAZA", "1 TAZA")
    .when(upper($"categoria") === "(1) UNA TAZA", "1 TAZA")
    .when(upper($"categoria") === "PRIMERA / 1 TAZA", "1 TAZA")
    .when(upper($"categoria") === "PRIMERA" && $"subclasificacion" === "CAFETERÍA", "1 TAZA")
    .when(upper($"categoria") === "LUJO" && $"clasificacion" === "ALOJAMIENTO", "5 ESTRELLAS")
    .when(upper($"categoria") === "CUARTA" && $"clasificacion" === "ALOJAMIENTO", "4 ESTRELLAS")
    .when(upper($"categoria") === "TERCERA" && $"clasificacion" === "ALOJAMIENTO", "3 ESTRELLAS")
    .when(upper($"categoria") === "SEGUNDA" && $"clasificacion" === "ALOJAMIENTO", "2 ESTRELLAS")
    .when(upper($"categoria") === "PRIMERA" && $"clasificacion" === "ALOJAMIENTO", "1 ESTRELLAS")
    .otherwise(upper($"categoria"))
)
)
: = [nombre: string, clasificacion: string ... 11 more fields]
df3: org.apache.spark.sql.DataFrame = [nombre: string, clasificacion: string ... 11 more fields]

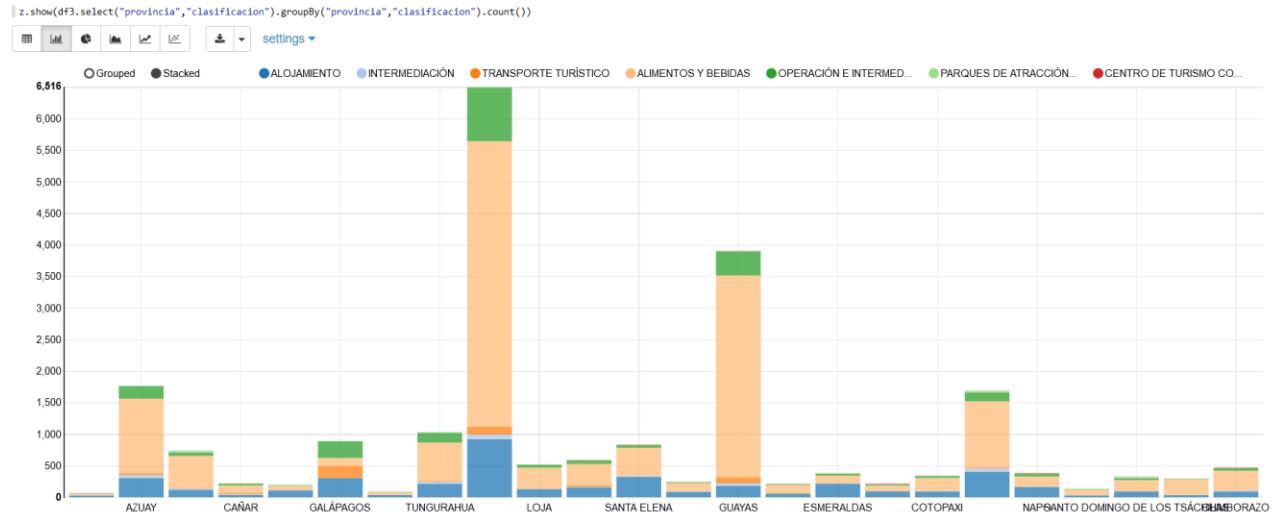
```

8. Se muestran los datos ya limpios.

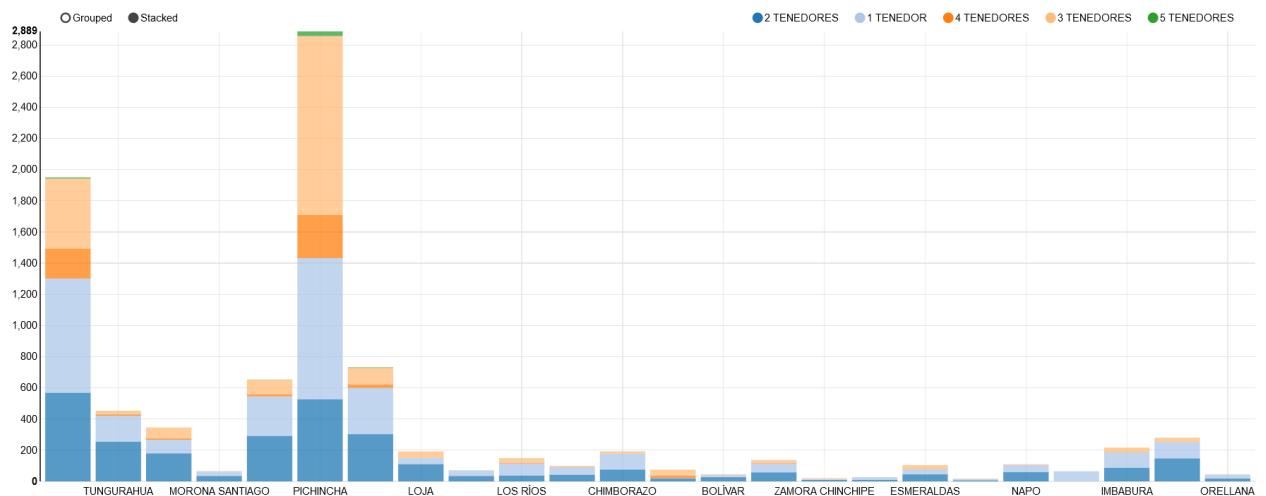
z.show(df3)											
nombre	clasificacion	subclasificacion	categoria	provincia	canton	parroquia	dir_referencia	direccion	telf_princ		
FAMA LA	ALIMENTOS Y BEBIDAS	FUENTE DE SODA	SEGUNDA	AZUAY	CUENCA	CUENCA, CABECERA CANTONAL Y CAPITAL PROVINCIAL.	null	BOLIVAR 12-86	2843739	^	
COLONIAS LAS No.3	ALIMENTOS Y BEBIDAS	RESTAURANTE	2 TENEDORES	AZUAY	CUENCA	CUENCA, CABECERA CANTONAL Y CAPITAL PROVINCIAL.	null	AV. ESPAÑA Y AV. HURTADO DE MENDOZA	2804790		
QUINTA ELOISA	INTERMEDIACIÓN	SALA DE RECEPCIONES Y BANQUETES	SEGUNDA	AZUAY	CUENCA	CUENCA, CABECERA CANTONAL Y CAPITAL PROVINCIAL.	null	JOSE RAFAEL PEÑAHERRERA Y FERNANDO DE ARAGON	72829459		
PINGUINO	ALIMENTOS Y BEBIDAS	FUENTE DE SODA	TERCERA	AZUAY	CUENCA	CUENCA, CABECERA CANTONAL Y CAPITAL PROVINCIAL.	null	REMIGIO CRESPO 6 87	2810586		
VILLA VERDE RECEPCIONES Y	INTERMEDIACIÓN	SALA DE RECEPCIONES Y	SEGUNDA	AZUAY	CUENCA	CUENCA, CABECERA	null	AV. DE LAS AMERICAS 9-263 Y	983123255	>	

Analisis de datos usando como fuente archivo original

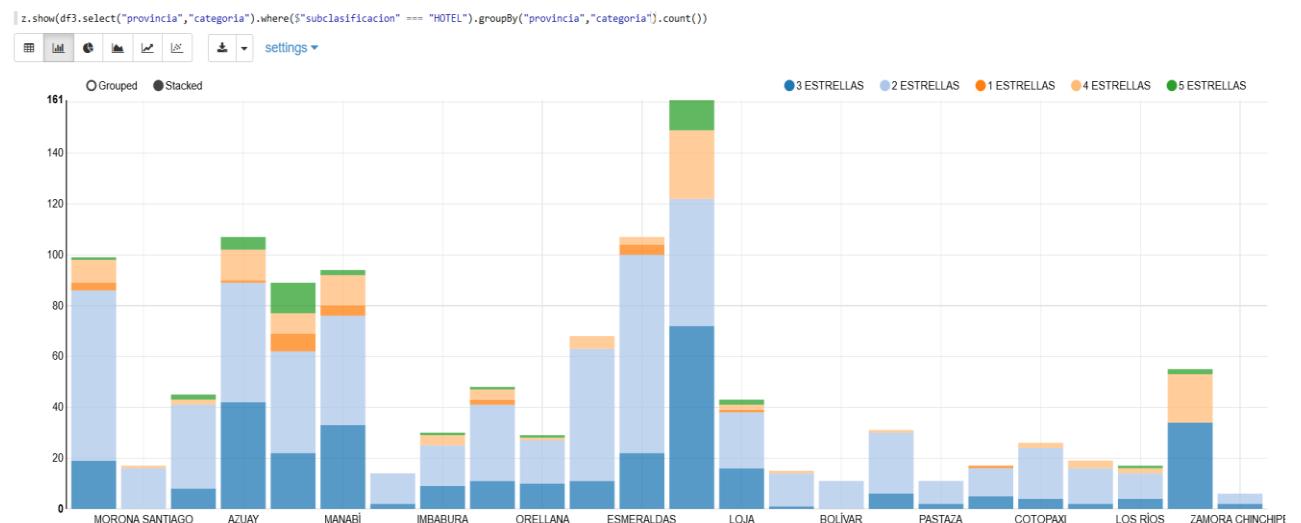
1. Cantidad de establecimientos turísticos por provincia y sector turístico



2. Cantidad de restaurantes por provincia y categoría.

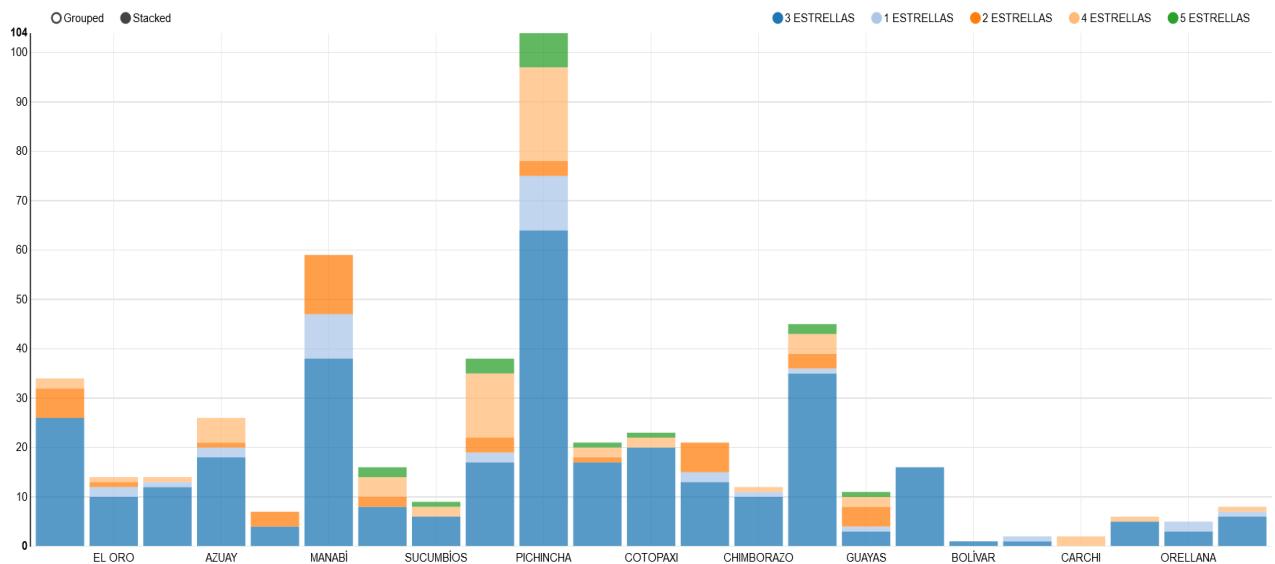


3. Cantidad de hoteles por provincia y categoría.



4. Cantidad de hosterías por provincia y categoría.

```
z.show(df3.select("provincia","categoria").where($"subclasificacion" === "HOSTERÍA").groupBy("provincia","categoria").count())
```



Análisis de formulario

```

("AZUAY", "Azuay"),
("CAÑAR", "Cañar"),
("BOLIVAR", "Bolívar"),
("GARCI", "Garchi"),
("ESMERALDAS", "Esmeraldas"),
("GUAYAS", "Guayas"),
("LOJA", "Loja")
)

val valorDefecto = valoresCombo(0)
z.put("provincia", z.select("Provincia", valoresCombo, valorDefecto))
z.run("paragraph_1659037806131_696869440")

```

Provincia

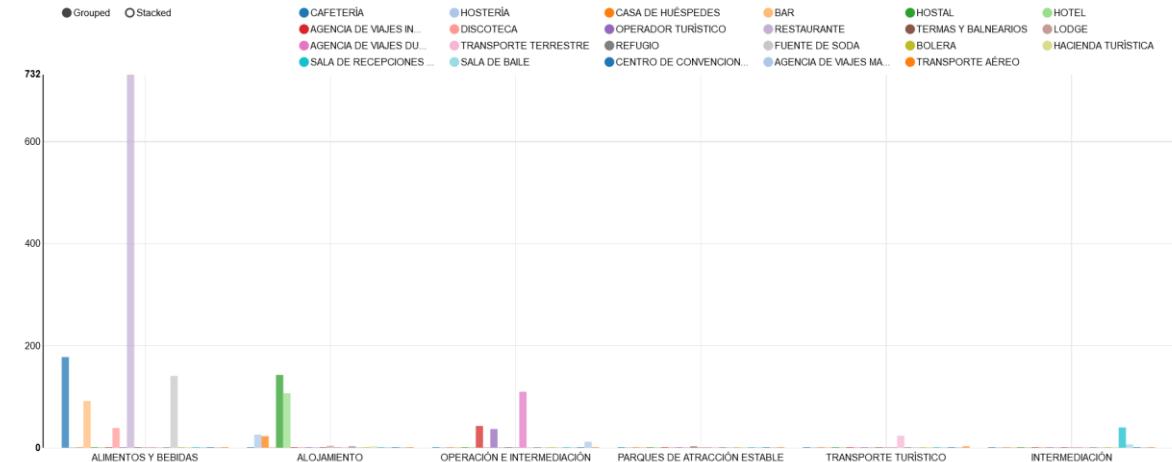
Azuay ▾

```
valoresCombo: Seq[(String, String)] = List((AZUAY,Azuay), (CAÑAR,Cañar), (BOLIVAR,Bolívar), (CARCHI,Carchi), (ESMERALDAS,Esmeraldas), (GUAYAS,Guayas), (LOJA,Loja))
valorDefecto: (String, String) = (AZUAY,Azuay)
```

Took 0 sec. Last updated by anonymous at July 31 2022, 10:12:18 PM.

1. Visualizamos la columna clasificación y subclasificación en la grafica de barras con sus respectivos números.

[z.show(df3.select("clasificacion", "subclasiacion").where("provincia == z.get("provincia")).groupBy("clasificacion", "subclasiacion").count())]



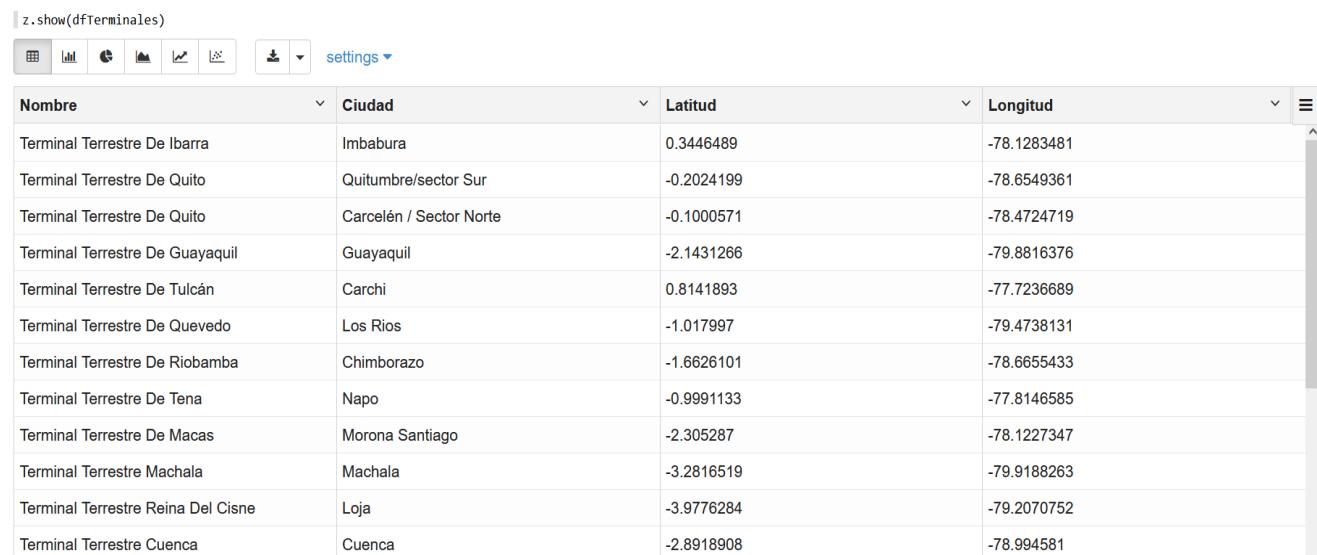
Análisis con datos de 2 dataframes

1. Lectura y almacenamiento del archivo csv en DataFrame

```
val dfTerminales = spark.read  
  .format("csv")  
  .option("inferSchema", "true")  
  .option("header", "true")  
  .option("delimiter", ",")  
  .load("/workspace/zeppelin-paavanzada/terminalesterrestres-csv.csv")
```

```
dfTerminales: org.apache.spark.sql.DataFrame = [Nombre: string, Ciudad: string ... 2 more fields]
```

2. Se muestra la información de la data de terminales terrestres.



Nombre	Ciudad	Latitud	Longitud
Terminal Terrestre De Ibarra	Imbabura	0.3446489	-78.1283481
Terminal Terrestre De Quito	Quitumbre/sector Sur	-0.2024199	-78.6549361
Terminal Terrestre De Quito	Carcelén / Sector Norte	-0.1000571	-78.4724719
Terminal Terrestre De Guayaquil	Guayaquil	-2.1431266	-79.8816376
Terminal Terrestre De Tulcán	Carchi	0.8141893	-77.7236689
Terminal Terrestre De Quevedo	Los Ríos	-1.017997	-79.4738131
Terminal Terrestre De Riobamba	Chimborazo	-1.6626101	-78.6655433
Terminal Terrestre De Tena	Napo	-0.9991133	-77.8146585
Terminal Terrestre De Macas	Morona Santiago	-2.305287	-78.1227347
Terminal Terrestre Machala	Machala	-3.2816519	-79.9188263
Terminal Terrestre Reina Del Cisne	Loja	-3.9776284	-79.2070752
Terminal Terrestre Cuenca	Cuenca	-2.8918908	-78.994581

Limpieza de datos del DataFrame de terminales

1. Se renombra las columnas nombre y ciudad.

```
val dfTerminales1 = dfTerminales  
  .withColumn("Nombre",  
    when($"nombre" === "Terminal Terrestre De Quito" && $"Ciudad" === "Quitumbre/sector Sur", "Terminal Terrestre Quitumbe / Sector Sur")  
    .when($"nombre" === "Terminal Terrestre De Quito" && $"Ciudad" === "Carcelén / Sector Norte", "Terminal Terrestre Carcelén / Sector Norte")  
    .otherwise($"Nombre"))  
  .withColumn("Ciudad",  
    when($"Ciudad" === "Quitumbre/sector Sur", "Quito")  
    .when($"Ciudad" === "Carcelén / Sector Norte", "Quito")  
    .otherwise($"Ciudad"))  
  .withColumnRenamed("Nombre", "nombre_terminal")
```

```
dfTerminales1: org.apache.spark.sql.DataFrame = [nombre_terminal: string, Ciudad: string ... 2 more fields]
```

2. Visualizamos los nombres de las columnas renombradas

```
dfTerminales1.printSchema()
```

```
root  
|-- nombre_terminal: string (nullable = true)  
|-- Ciudad: string (nullable = true)  
|-- Latitud: string (nullable = true)  
|-- Longitud: string (nullable = true)
```

3. Se muestran los datos ya limpios.

z.show(dfTerminales1)

The screenshot shows a Jupyter Notebook cell with the command `z.show(dfTerminales1)`. The output is a table titled "settings" with columns: nombre_terminal, Ciudad, Latitud, and Longitud. The data includes various terminals like Terminal Terrestre De Riobamba, Chimborazo, -1.6626101, -78.6655433, etc.

nombre_terminal	Ciudad	Latitud	Longitud
Terminal Terrestre De Riobamba	Chimborazo	-1.6626101	-78.6655433
Terminal Terrestre De Tena	Napo	-0.9991133	-77.8146585
Terminal Terrestre De Macas	Morona Santiago	-2.305287	-78.1227347
Terminal Terrestre Machala	Machala	-3.2816519	-79.9188263
Terminal Terrestre Reina Del Cisne	Loja	-3.9776284	-79.2070752
Terminal Terrestre Cuenca	Cuenca	-2.8918908	-78.994581
Terminal Terrestre Zamora	Zamora	-4.0674939	-78.954945
Terminal Terrestre Latacunga	Latacunga	-0.8717535	-77.8131405
Terminal Terrestre De Ambato	Ambato	-1.2361134	-78.6186811
Terminal Terrestre De Manta	Manta	-0.9606632	-80.692665
Terminal Terrestre De Santo Domingo	Santo Domingo	-0.2549306	-79.1635315
Terminal Terrestre De Nueva Loja	Nueva Loja	0.0916948	-76.8884608
Terminal Terrestre De Azogues	Azogues	-2.7277288	-78.8431156
Terminal Terrestre De Babahoyo	Babahoyo	-1.8117528	-79.5462591
Terminal Terrestre De Macas	Macas	-2.305287	-78.1227347

4. Se muestra el mapa con los terminales terrestres del Ecuador.



5. Aquí podemos visualizar el número de alojamientos por cantón que tiene el terminal terrestre.

z.show(dfTerminales1.join(df3, \$"cantón" === upper(\$"Ciudad"), "inner") .select("cantón","nombre_terminal") .where(\$"clasificación" === "ALOJAMIENTO") .groupby("cantón","nombre_terminal") .count() .withColumnRenamed("count", "Número de alojamientos por cantón que tiene terminal terrestre"))

The screenshot shows a Jupyter Notebook cell with the command `z.show(dfTerminales1.join(df3, $"cantón" === upper($"Ciudad"), "inner") .select("cantón","nombre_terminal") .where($"clasificación" === "ALOJAMIENTO") .groupby("cantón","nombre_terminal") .count() .withColumnRenamed("count", "Número de alojamientos por cantón que tiene terminal terrestre"))`. The output is a table with columns: canton, nombre_terminal, and Número de alojamientos por cantón que tiene terminal terrestre. The data includes GUAYAQUIL, AZOGUES, MANTA, AMBATO, QUITO, CUENCA, QUITO, ZAMORA, MACHALA, LOJA, SANTO DOMINGO, LATACUNGA, ESMERALDAS, and BABAHoyo.

canton	nombre_terminal	Número de alojamientos por cantón que tiene terminal terrestre
GUAYAQUIL	Terminal Terrestre De Guayaquil	122
AZOGUES	Terminal Terrestre De Azogues	9
MANTA	Terminal Terrestre De Manta	106
AMBATO	Terminal Terrestre De Ambato	55
QUITO	Terminal Terrestre Carcelén / Sector Norte	720
CUENCA	Terminal Terrestre Cuenca	257
QUITO	Terminal Terrestre Quito / Sector Sur	720
ZAMORA	Terminal Terrestre Zamora	20
MACHALA	Terminal Terrestre Machala	55
LOJA	Terminal Terrestre Reina Del Cisne	75
SANTO DOMINGO	Terminal Terrestre De Santo Domingo	92
LATACUNGA	Terminal Terrestre Latacunga	49
ESMERALDAS	Terminal Terrestre De Esmeraldas	43
BABAHoyo	Terminal Terrestre De Babahoyo	8

6. Visualizamos en un grafico de barras los alojamientos por cantón que tiene el terminal terrestre.



DataFrame Aeropuertos

1. Lectura y almacenamiento del archivo csv en DataFrame

```
val dfAeropuertos = spark.read
  .format("csv")
  .option("inferSchema", "true")
  .option("header", "true")
  .option("delimiter", ",")
  .load("/workspace/zeppelin-paavanzada/aeropuertos-csv.csv")
```

dfAeropuertos: org.apache.spark.sql.DataFrame = [Aeropuerto: string, Ciudad: string ... 2 more fields]

2. Se muestra la información de la data de terminales terrestres.

Aeropuerto	Ciudad	latitud	longitud
Aeropuerto Jose Maria Velasco Ibarra	Macara	-4.3807913	-79.9436849
Aeropuerto Camilo Ponce Enríquez	Catamayo	-3.996246	-79.3741991
Aeropuerto Chachoaña	Ambato	-1.213156	-78.5771455
Aeropuerto Chimborazo	Riobamba	-1.65343	-78.6560974
Aeropuerto Cosme Renella	Salinas	-2.2020466	-80.9924443
Aeropuerto de Gualاقiza	Gualاقiza	-3.4232099	-78.5670013
Aeropuerto de Jipijapa	Jipijapa	-1	-80.6666641
Aeropuerto de Macas	Macas	-2.29917	-78.1207962
Aeropuerto de Nueva Loja	Nueva Loja	0.093056	-76.8675003
Aeropuerto de Pedernales	Pedernales	0.067619	-80.050499
Aeropuerto de Putumayo	Puerto El Carmen	0.083333	-75.9000015
Aeropuerto de San Cristóbal	Puerto Baquerizo Moreno	-0.910206	-89.6174011
Aeropuerto de Sucúa	Sucúa	-2.483	-78.1669998
Aeropuerto El Carmen	Montalvo	-2.0670099	-76.9757004
Aeropuerto Francisco de Orellana	Francisco De Orellana	-0.4617447	-76.988412
Aeropuerto General Manuel Serrano	Machala	-3.2688999	-79.9616013

3. Aquí podemos visualizar el número de alojamientos por cantón que tiene el aeropuerto.

```
z.show(dfAeropuertos.join(df3, $"cantón" === upper($"Ciudad"), "inner")
    .select("cantón", "Aeropuerto")
    .where($"clasificación" === "ALOJAMIENTO")
    .groupby("cantón", "Aeropuerto")
    .count()
    .withColumnRenamed("count", "Número de alojamientos por cantón que tiene aeropuerto"))
```



The screenshot shows a data visualization interface with a toolbar at the top containing various icons for filtering, sorting, and saving data. The main area displays a table with three columns: 'canton' (Canton), 'Aeropuerto' (Airport), and 'Numero de alojamientos por canton que tiene aeropuerto' (Number of accommodations per canton that has an airport). The data is as follows:

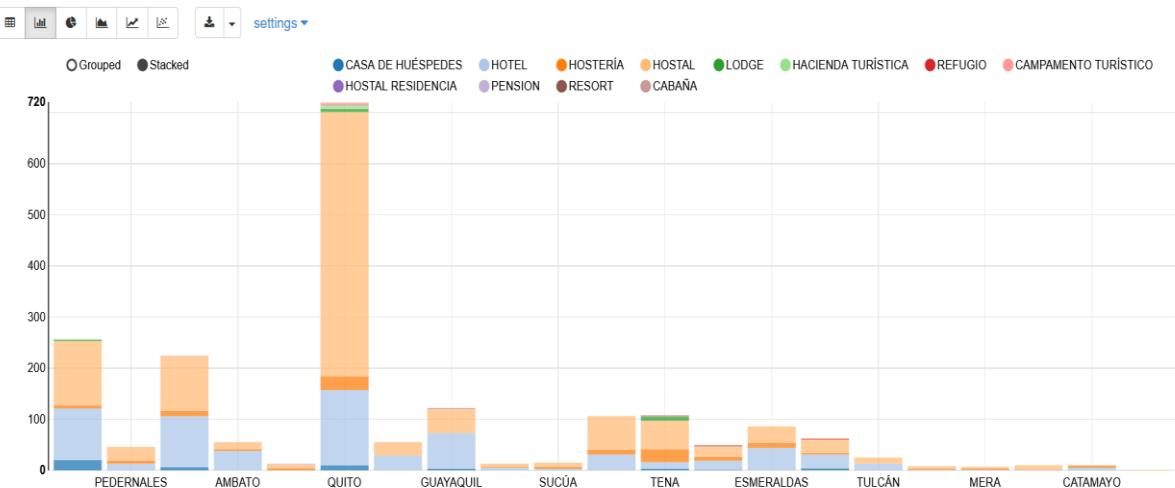
canton	Aeropuerto	Numero de alojamientos por canton que tiene aeropuerto
AMBATO	Aeropuerto Chachan	55
JIPIJAPA	Aeropuerto de Jipijapa	13
QUITO	Aeropuerto Internacional Mariscal Sucre	720
GUALAQUIZA	Aeropuerto de Gualajiza	13
GUAYAQUIL	Aeropuerto Internacional José Joaquín de Olmedo	122
SALINAS	Aeropuerto Cosme Renella	112
ESMERALDAS	Aeropuerto Coronel Carlos Concha Torres	43
RIOBAMBA	Aeropuerto Chimborazo	62
LATACUNGA	Aeropuerto Internacional Cotopaxi	49
TENA	Aeropuerto Internacional Jumandy	108
ESMERALDAS	Aeropuerto General Rivadeneira	43

4.

```

z.show(dfAeropuertos.join(df3, $"canton" === upper($"Ciudad"), "inner")
      .select("canton","Aeropuerto","subclasiificación")
      .where($"clasiificación" === "ALOJAMIENTO")
      .groupBy("canton","Aeropuerto","subclasiificación")
      .count()
      .withColumnRenamed("count", "Número de alojamientos por cantón que tiene aeropuerto"))

```



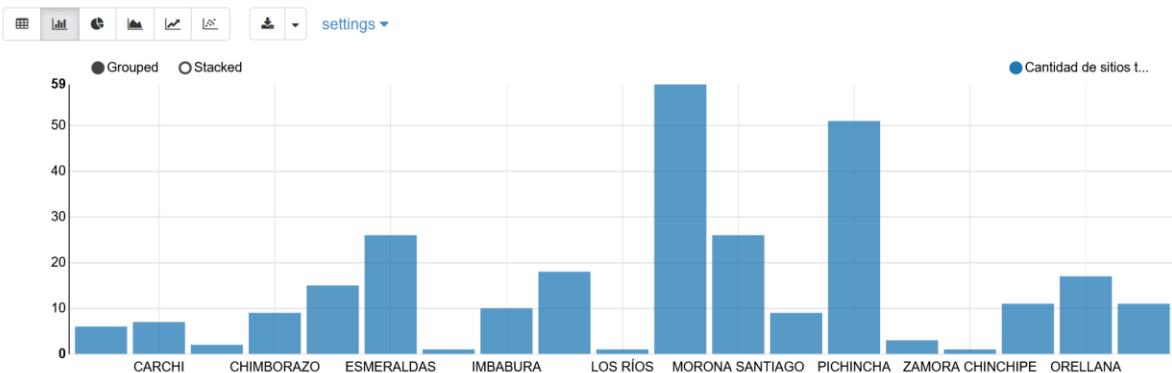
Análisis de datos usando como fuente MySQL

1. Se selecciona la base de datos a usar

```
%mysql  
USE practicagrupal
```

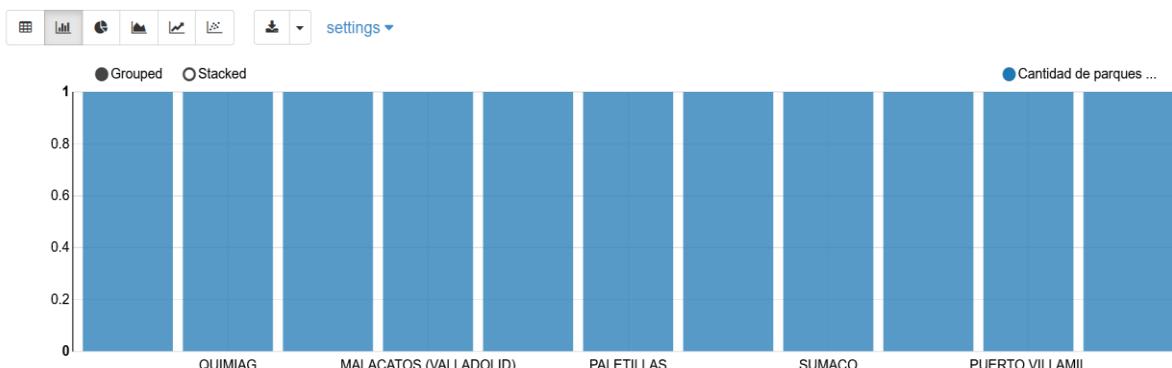
2. Cantidad de sitios turísticos por provincia.

```
%mysql  
SELECT p.nomProvincia, count(*) AS "Cantidad de sitios turísticos" FROM sitiosturisticos s  
INNER JOIN provincia p ON s.idProvincia = p.idProvincia  
GROUP BY p.nomProvincia;
```



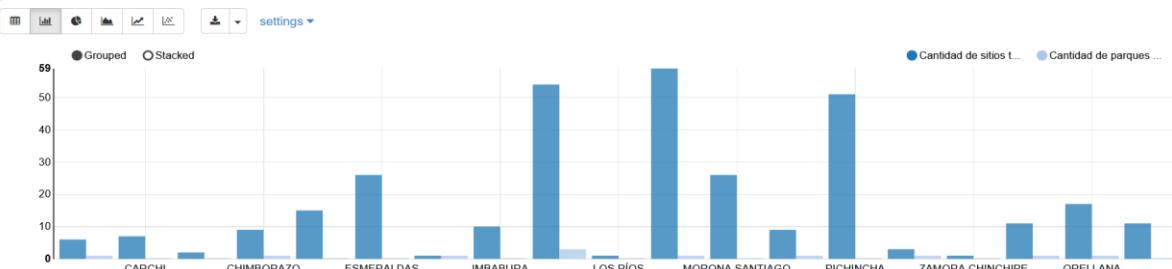
3. Cantidad de parques nacionales por parroquia.

```
%mysql  
SELECT p.nomParroquia, count(*) AS "Cantidad de parques nacionales" FROM parquesnacionales s  
INNER JOIN parroquias p ON s.idParroquia = p.idParroquia  
GROUP BY p.nomParroquia;
```

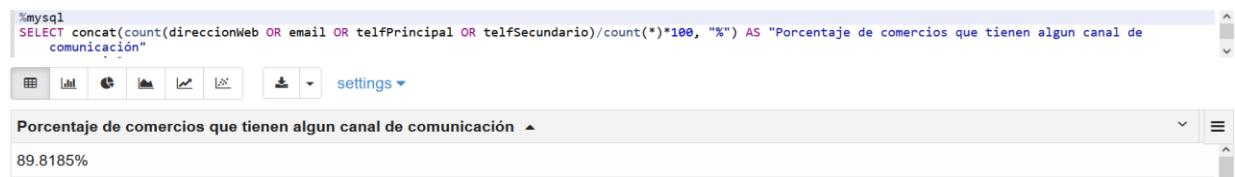


4. Cantidad de sitios turísticos y de parques nacionales por provincia

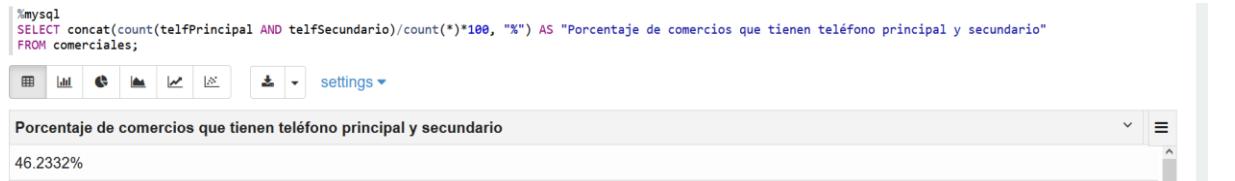
```
%mysql  
SELECT a.nomProvincia AS "Provincia", a.num_sitios AS "Cantidad de sitios turísticos", b.num_parques AS "Cantidad de parques nacionales"  
FROM (  
    SELECT p.nomProvincia, count(*) AS num_sitios FROM sitiosturisticos s  
    INNER JOIN provincia p ON s.idProvincia = p.idProvincia  
    GROUP BY p.nomProvincia  
) a  
LEFT JOIN (  
    SELECT pr.nomProvincia, count(*) AS num_parques FROM parquesnacionales s  
    INNER JOIN parroquias p ON s.idParroquia = p.idParroquia  
    INNER JOIN canton c ON p.idCanton = c.idCanton  
    INNER JOIN provincia pr ON c.idProvincia = pr.idProvincia  
    GROUP BY pr.nomProvincia  
) b ON a.nomProvincia = b.nomProvincia;
```



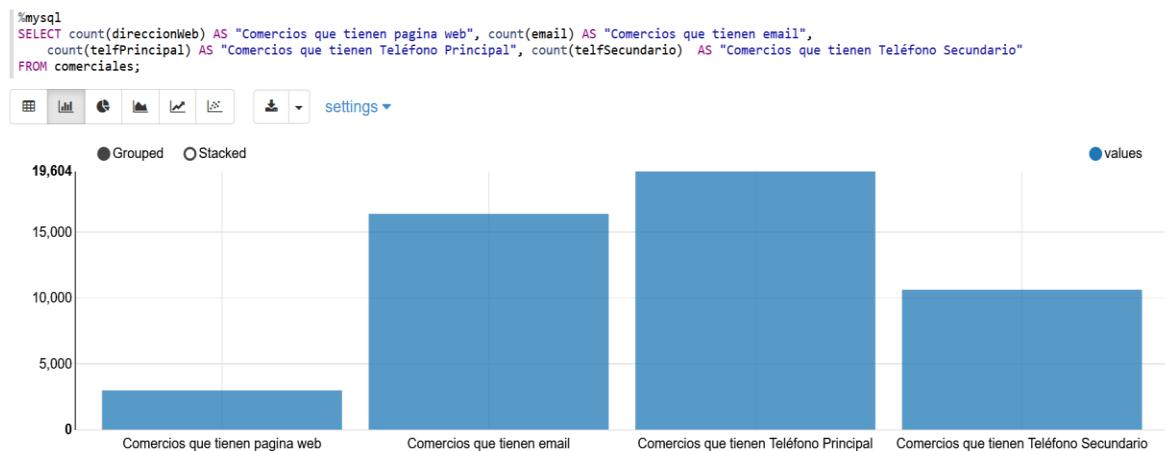
5. Porcentaje de comercios que tienen algun canal de comunicación



6. Porcentaje de comercios que tienen teléfono principal y secundario.

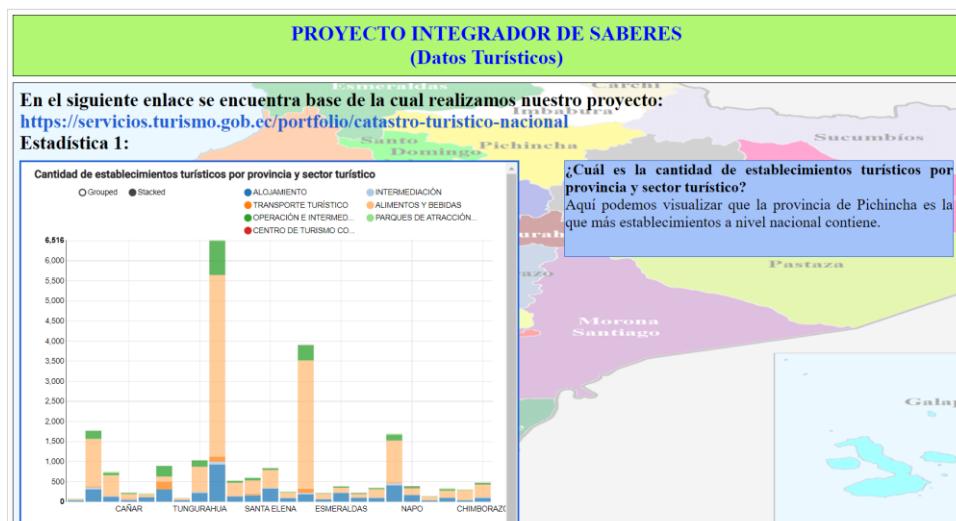


7. Cantidad de comercios con un tipo de página web, email, teléfono principal, teléfono secundario.

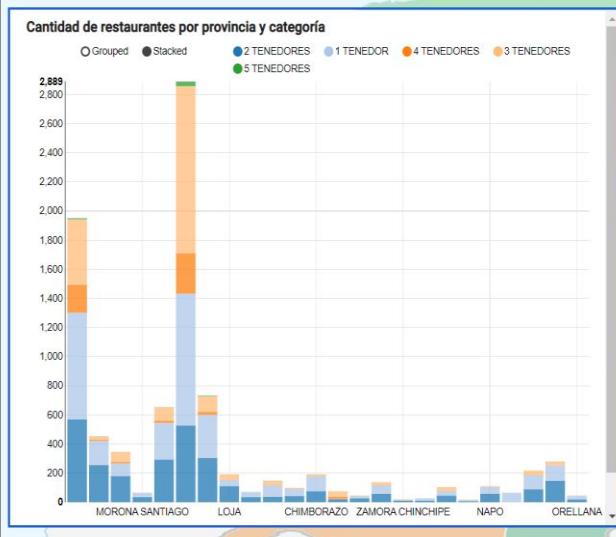


• Resultados obtenidos

A continuación, se mostrará los resultados que se pudo obtener implementando todos los datos y pudiendo mostrarlos en una página web de una manera didáctica y mejor visualmente para su respectivo análisis de datos.



Estadística 2:

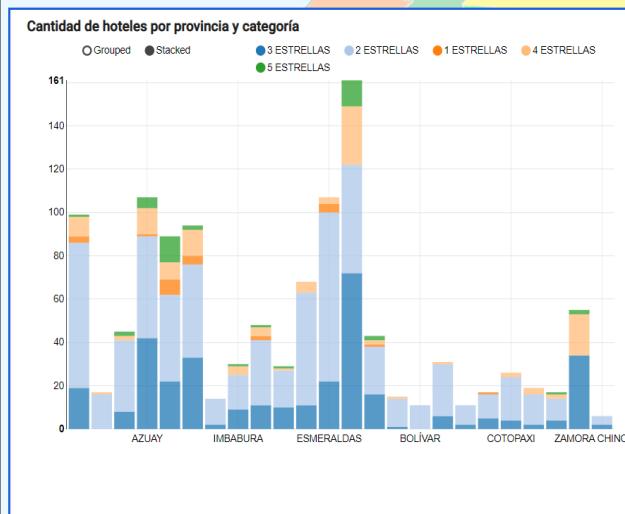


¿Cuál es la cantidad restaurantes por provincia y categoría?

Visualizamos las cantidades de restaurantes que existen en nuestro país con sus respectivas categorías.



Estadística 3:

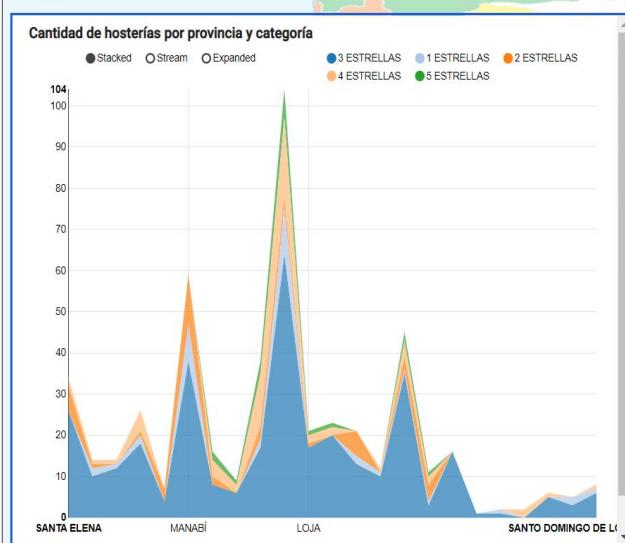


¿Cuál es la cantidad de hoteles por provincias y categoría?

Podemos visualizar el número de hoteles que se encuentra cada una de las provincias.



Estadística 4:

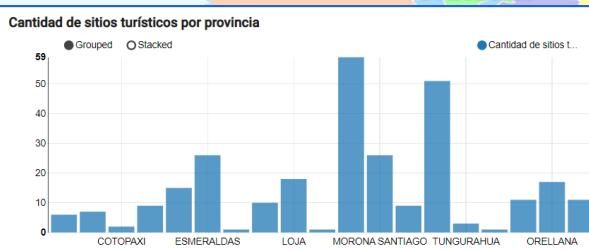


¿Cuál es la cantidad de hosterías por provincias y categoría?

Podemos visualizar el número de hosterías que se encuentra cada una de las provincias.

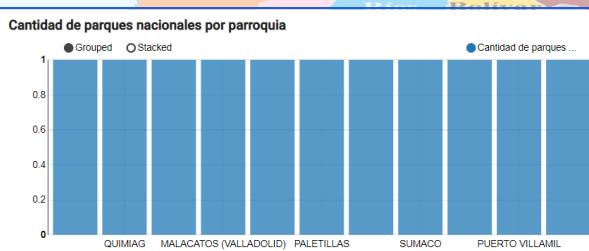


Estadística 5:



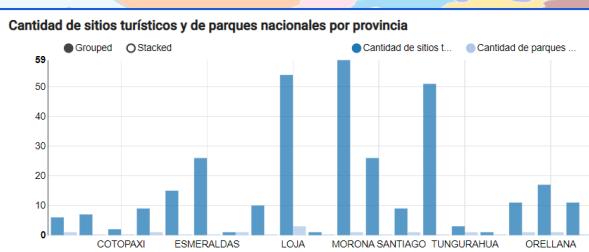
¿Cuál es la cantidad de sitios turísticos por provincia?
Aquí podemos visualizar los sitios turísticos de cada provincia y podemos observar que la provincia de Manabí es la que más sitios tiene a nivel nacional.

Estadística 6:



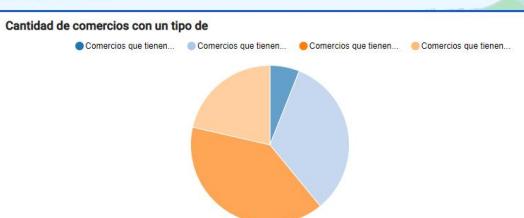
¿Cuál es la cantidad de parques nacionales por parroquia?
Visualizamos que cada una de las parroquias tiene un solo parque nacional.

Estadística 7:



¿Cuál es la cantidad de sitios turísticos y de parques nacionales por provincia ?
En esta imagen visualizamos que Manabí tiene la gran cantidad de sitios turísticos y que la provincia de Loja es la que más parques nacionales tiene.

Estadística 8:



¿Cuál es la cantidad de comercios con un tipo de página web, email, teléfono principal y teléfono secundario?
En esta imagen visualizamos que la columna teléfono principal es la que más datos nos proporciona de las demás columnas.

Elaborado por:

Renny Poma
Kevin Cabrera
Gilmar Jiménez

- **Conclusiones**

La Universidad Técnica Particular de Loja, está dando grandes pasos hacia la excelencia académica, mediante estrategias metodológicas aplicadas a la nueva estructura curricular vigente, por lo que se torna de mucha relevancia, dejar constancia no solo del logro del resultado del aprendizaje, sino también, de la eficacia del mismo. Por lo tanto, el estructurar y orientar de manera correcta el Proyecto integrador de Saberes donde interactúan en esta ocasión la materia de “Base de datos Avanzada” y “Programación Avanzada”, nos ha permitido a nosotros como estudiantes aprender a resolver en forma progresiva problemas de la profesión, y así mismo evidenciar el desarrollo de las habilidades acordes al perfil de egreso. Por tanto, el Proyecto Integrador de Saberes nos ha servido para la medición del resultado del proceso académico, convirtiéndose en un verdadero acierto.

Por lo expuesto se podría considerar al proyecto integrador como una técnica o instrumento muy eficiente para la gestión formativa y participativa, pues nos promueve a la investigación – acción, puesto que entrelaza un enfoque experimental con programas de acción, que responden a problemas, en este caso en un contexto real, generando así, los cambios requeridos y necesarios para mejorar o solucionar los mismos, es decir, una técnica que fusiona la formación teórica con lo práctico, y que se evidencia efectivamente en los resultados de aprendizajes, corroborando así el grado de complejidad de la temática a resolver.

- **Bibliografía**

GitHub - alexander-jimenez11/Proyecto-Integrador. (s. f.). GitHub.
<https://github.com/alexander-jimenez11/Proyecto-Integrador>

Procesando Datos con Spark (I) — Configurando Apache Spark y Apache Zeppelin. (s. f.). Medium. <https://medium.com/@crscardellino/procesando-datos-con-spark-i-configurando-apache-spark-y-apache-zeppelin-b8dbda672aa4>

Lugares Turístico - Lugares Turísticos XLSX - Ambar DataStore. (s. f.). LT.
<http://ambar.utpl.edu.ec/it/dataset/lugares-turistico/resource/ba3c85d5-bbc9-4f3ca059-949532ae685b>

Parques Nacionales del Ecuador. (s. f.). GoRaymi. <https://www.goraymi.com/ec/ecuador/parques-nacionales/parques-nacionales-ecuador-ag356h63t>