



PRACTICUM 1.2

PROYECTO INTEGRADOR

Memoria Final

GRUPO #8

29/07/2022

INTEGRANTES

Isaac Álvarez
Víctor Carrión
Tony Gaona

2022

Índices

Introducción.....	3
Planificación del Trabajo	3
Fuentes de Datos	3
Datos Base	3
Datos Complementarios.....	4
Componente de Base de Datos	5
Preprocesamiento de Datos	5
Diseño conceptual y diseño lógico	6
Diseño conceptual.....	6
Diseño Lógico	7
Implementación de la Base de Datos	7
Componente Programación.....	8
Explicación de herramientas utilizadas	8
Explicación de comandos y sentencias utilizadas	8
Resultados Obtenidos	9
Visualizaciones con su análisis e interpretación	9
Conclusiones	9
Bibliografía.....	¡Error! Marcador no definido.

Introducción

En el presente documento se redactará un informe con todo el proceso realizado para el prácticum 1.2. Dentro del componente de Base de Datos se realizaron los siguientes procesos: aumento de variables extraídas de la data (Consolidado-Nacional2022-publico-1-web), elaboración del diagrama entidad-relación, diagrama lógico, creación del script DDL (lenguaje de definición de datos), implementación física y población de la base de datos, en cambio en el componente de Programación Avanzada se nos encomendó el crea una página web local y con consultas en Spark con Apache Zeppelin con la finalidad de brindar una mejor explicación de lo realizado a lo largo del presente ciclo académico.

Planificación del Trabajo

Este informe se ha desarrollado el componente de Base de Datos como trabajos extra-clase mientras que en el componente de Programación Avanzada la mayoría se han realizado dentro de las horas de practica/tutoría. Por lo que a la planificación como tal se realizó según las directrices del docente y la planificación grupal realizada por nuestro grupo.

Fuentes de Datos

Datos Base

- Nombre comercial
- Actividad
- Clasificación
- Categoría
- Provincia
- Cantón
- Parroquia
- Referencia de dirección
- Dirección
- Teléfono principal
- Teléfono secundario
- Correo electrónico
- Dirección web

Datos obtenidos de:

<https://github.com/IsaacAlvarez12/ProyectoIntegradorG8/blob/main/BaseDeDatos/Consolidado-Nacional-2022-publico-1-web.xlsx>

Datos Complementarios

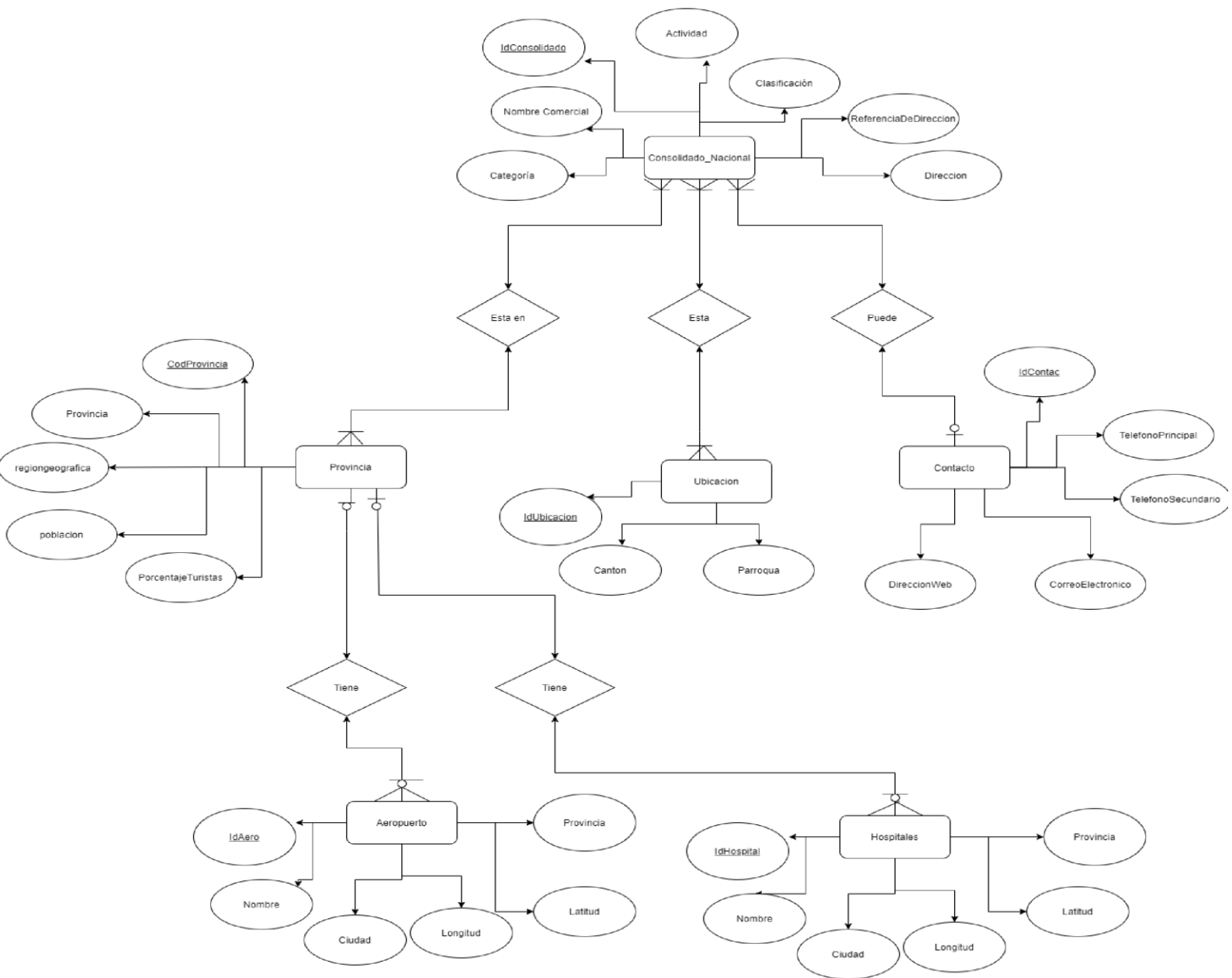
- Aeropuertos por provincia:
Datos obtenidos de <http://ambar.utpl.edu.ec/dataset/aeropuertosecuador>.
- Hospitales:
Datos obtenidos de <http://ambar.utpl.edu.ec/dataset/hospitales-del-ecuador>
- Porcentaje Turistas:
Datos obtenidos de
<https://www.cfn.fin.ec/wpcontent/uploads/2017/10/Ficha-Sectorial-Turismo.pdf>
- Códigos de parroquias-cantón:
Datos obtenidos de
<https://www.ecuadorencifras.gob.ec/clasificadorgeografico-estadistico-dpa/>
- Población por Provincias:
Datos obtenidos de <https://es.statista.com/estadisticas/1191532/numero-depersonas-en-ecuador-por-provincia/>
- Región Geográfica:
Datos obtenidos de <https://ec.viajandox.com/provincias-ecuador-PV5>
- Todos los archivos mencionados en este punto del informe se encuentran dentro del archivo de Excel en el repositorio:
<https://github.com/IsaacAlvarez12/ProyectoIntegradorG8/blob/main/BaseDeDatos/DatosLimpios.xlsx>.

Preprocesamiento de Datos

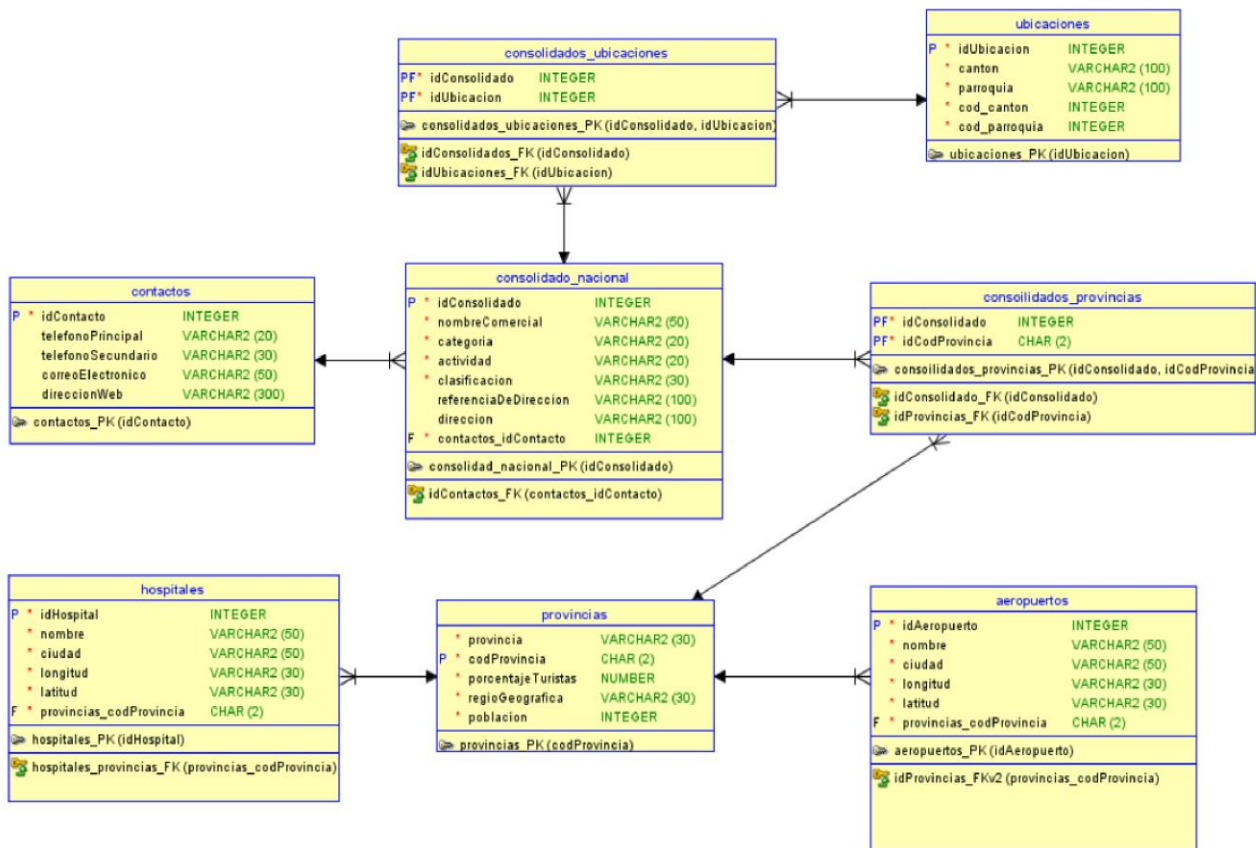
[illegible]

Diseño conceptual y diseño lógico

Diseño conceptual

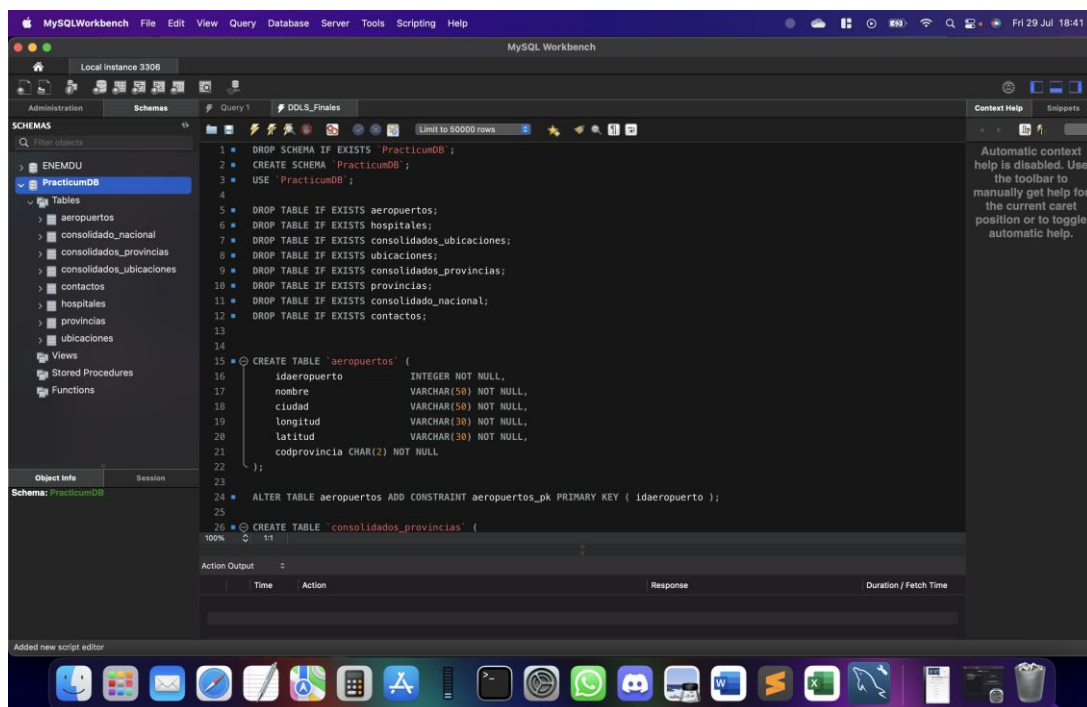


Diseño Lógico



Implementación de la Base de Datos

La implementación de la base se desarrolló en MySQLWorkbench mediante la copia de los scripts creados mediante el Excel y la creación de tablas mediante la conversión del modelo lógico del mismo



Componente Programación

Explicación de herramientas utilizadas

Para el desarrollo de este componente se utilizó Spark como punto fuente, así mismo para poder trabajar con este se utilizó Apache Zeppelin e IntelliJ Idea como IDE inteligente para modificar las consultas y aumentar la velocidad de escritura, también se usaron Plugin para Zeppelin de la familia Helium llamado GeoSpark

Explicación de comandos y sentencias utilizadas

Usaremos como ejemplo la siguiente consulta en Spark para explicar algunos de los comandos a utilizar

```
z.show(dfCatastroClean.select("provincia")
.where($"subclasificacion" === "BAR" && $"categoria" === "3 COPAS")
.groupBy($"provincia").count()
.orderBy(desc("count")).limit(5))
```

El primer z.show, es el encargado de presentar los datos de manera elegante en tablas, luego hacemos referencia al DataFrames y con el comando select, colocamos las provincias a mostrar, el where que nos sirve para proponer condiciones de búsqueda, el group by para agruparlos en bloques, el count para contar cuantos elementos hay dentro de estas agrupaciones y el orderby para ordenarles, así mismo dentro de este está el desc para referirnos a un ordenamiento descendente y el limit para solo mostrar un máximo de filas

Para el caso de comandos a nivel de base de datos usaremos la siguiente consulta para su presentación

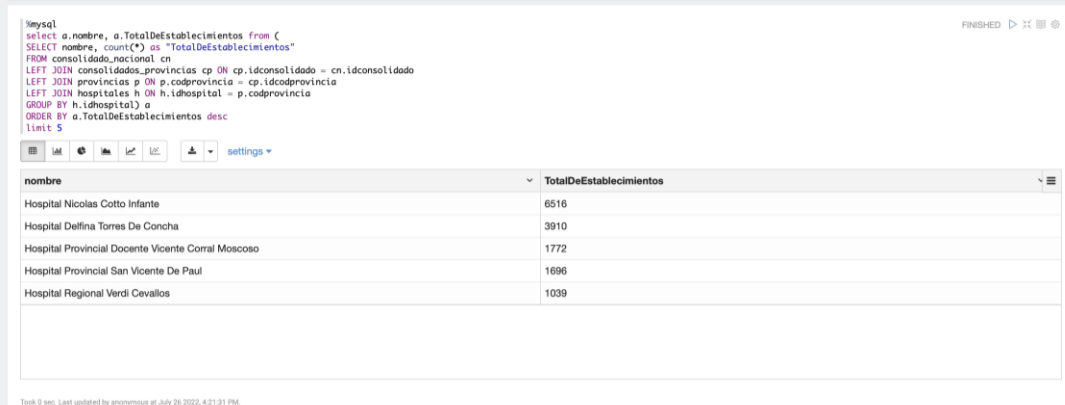
```
%mysql
SELECT nombre, CONCAT('POINT(',longitud,',',latitud,')') "geo"
FROM consolidado_nacional cn
LEFT JOIN consolidados_provincias cp ON cp.idconsolidado = cn.idconsolidado
LEFT JOIN provincias p ON p.codprovincia = cp.idcodprovincia
LEFT JOIN hospitales h ON h.idhospital = p.codprovincia
GROUP BY nombre, longitud, latitud
```

Primero que nada usamos el %mysql para hacer conexión con la base mediante el intérprete creado en las configuraciones de Zeppelin, luego de esto el SELECT para poder referenciar las columnas a mostrar, así mismo el CONCAT, que nos permite unir textos para generar cadenas mediante una secuencia y las comillas al final es para renombrar el campo, el FROM referencia a una tabla principal y los LEFT JOIN para unir de manera que la tabla principal siempre prevalezca y el GROUP BY para agruparlos según esas condiciones

Resultados Obtenidos

Visualizaciones con su análisis e interpretación

La siguiente consulta nos da a conocer el total de establecimientos cercanos a los hoteles, esto nos sería útil para saber por ejemplo cuales hospitales pueden tender a tener una mayor afluencia de aforo y por ende una atención más lenta



The screenshot shows a Databricks SQL interface with a MySQL query and its results. The query is as follows:

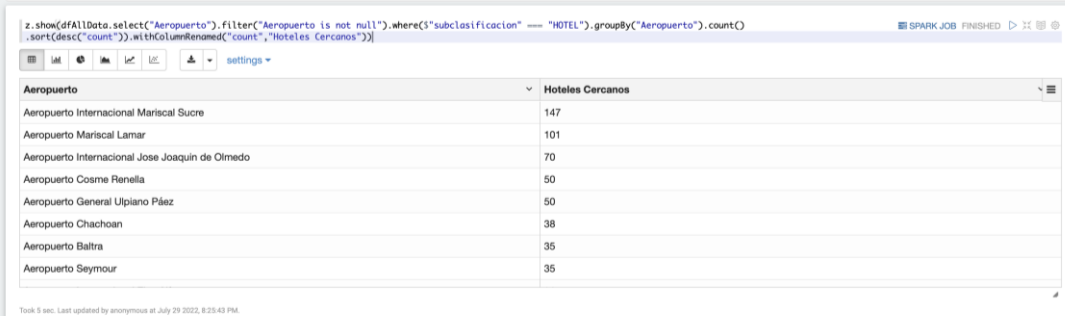
```
mysql
select a.nombre, a.TotalDeEstablecimientos from (
SELECT nombre, count(*) as "TotalDeEstablecimientos"
FROM consolidado_nacional cn
LEFT JOIN consolidados.provincias cp ON cp.idconsolidado = cn.idconsolidado
LEFT JOIN provincias p ON p.codprovincia = cp.idcodprovincia
LEFT JOIN hospitales h ON h.idhospital = p.codprovincia
GROUP BY h.idhospital) a
ORDER BY a.TotalDeEstablecimientos desc
limit 5
```

The results table shows the top 5 hospitals by total establishments:

nombre	TotalDeEstablecimientos
Hospital Nicolas Cotto Infante	6516
Hospital Delfina Torres De Concha	3910
Hospital Provincial Docente Vicente Corral Moscoso	1772
Hospital Provincial San Vicente De Paul	1696
Hospital Regional Verdi Cevallos	1039

Task 0 sec. Last updated by anonymous at July 29 2022, 4:21:31 PM.

Esta consulta nos muestra el total de hoteles cercanos a un Aeropuerto, los cual nos sería útil en casos de requerir algún tipo de alojamiento rápido cerca de estos y saber cuáles tienen mayor disponibilidad de habitaciones



The screenshot shows a Databricks SQL interface with a Spark SQL query and its results. The query is as follows:

```
z.show(dfAllData.select("Aeropuerto").filter("Aeropuerto is not null").where($"subclasificacion" === "HOTEL").groupBy("Aeropuerto").count()
.sort(desc("count")).withColumnRenamed("count", "Hoteles Cercanos"))
```

The results table shows the total number of hotels near various airports:

Aeropuerto	Hoteles Cercanos
Aeropuerto Internacional Mariscal Sucre	147
Aeropuerto Mariscal Lamar	101
Aeropuerto Internacional Jose Joaquin de Olmedo	70
Aeropuerto Cosme Renella	50
Aeropuerto General Ulpiano Pérez	50
Aeropuerto Chachoan	38
Aeropuerto Baltra	35
Aeropuerto Seymour	35

Task 0 sec. Last updated by anonymous at July 29 2022, 8:25:43 PM.

Conclusiones

- En conclusión, decimos que para trabajar con datos excesivos una herramienta que nos facilito el trabajo fue el apache zeppelin, el cual con su debido tratamiento y configuración nos permitió lograr manipular los datos usados en este proyecto para de esta manera dar una mejor explicación de estos.
- Le evidencia que presentamos anteriormente demuestra que al trabajar con bases de datos nos enriquecimos de conocimiento sobre el lenguaje SQL, ya que es el lenguaje que manejamos al realizar este proyecto.
- Después de realizar las consultas correspondientes, manejo correcto de spark y de la programación usada en sí, se concluye que este proyecto nos ayudó a adquirir nuevos conocimientos de bases de datos y programación avanzada ya que no solo aprendimos a manejarlos, sino que los juntamos y logramos entender su funcionamiento para cumplir el objetivo del proyecto integrador.

Referencias

- <https://zeppelin.apache.org/>
- <https://www.gitpod.io/>
- <https://github.com/>
- <https://www.mysql.com/>
- <https://app.diagrams.net/>
- <https://www.oracle.com/database/sqldeveloper/>