Parte B: Información territorial basada en las comunas de Chile

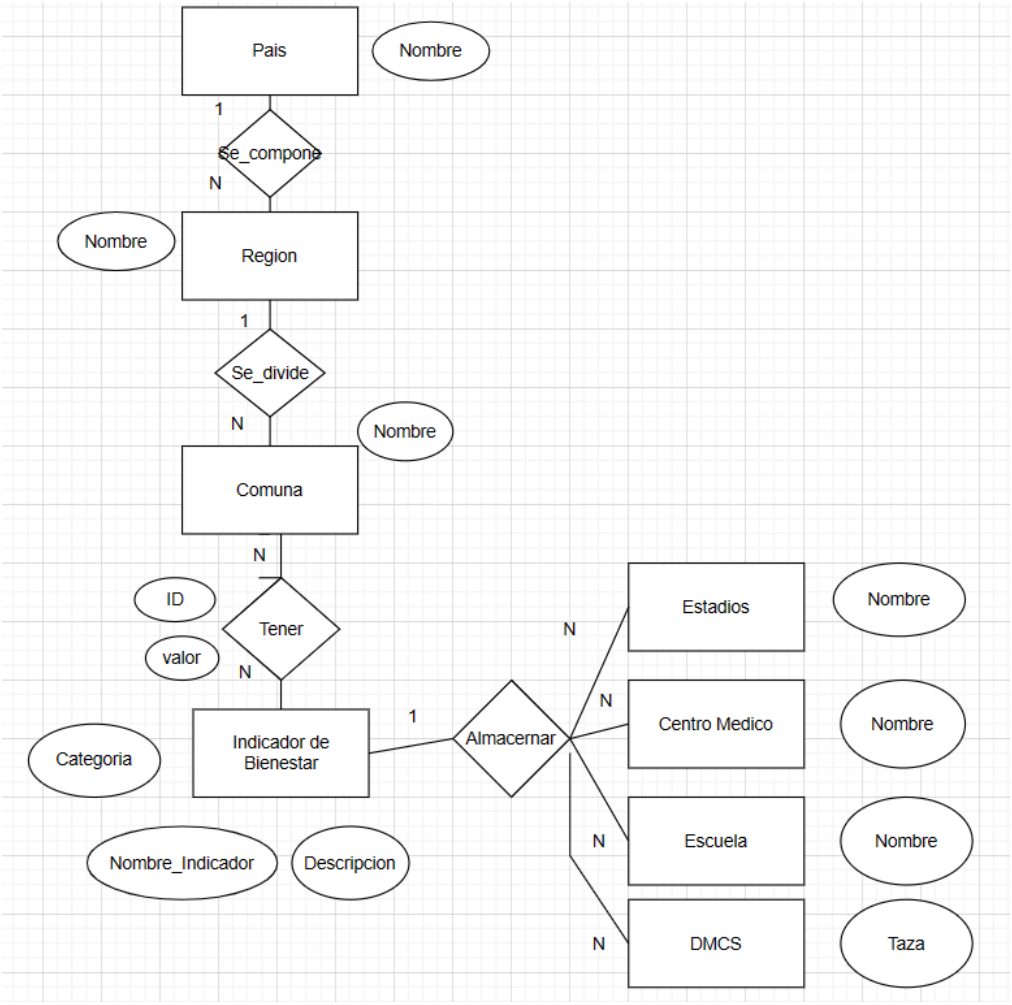
Integrantes: Cristóbal Felipe Rebolledo Oyanedel

Benjamín Enrique Parra Barbet

Carolina Andrea Obreque Higueras

Profesores: Luis Veas y Matthieu Vernier

Modelo Entidad-Relación:



Modelo Relacional:

Pais (PK_Nombre_pais)

Region (PK_Nombre_region, FK_ID_Pais)

Comuna (PK_Nombre_comuna, FK_ID_region)

Tener (PK_ID, Valor, FK_Nombre_comuna, FK_Nombre_indicador)

Indicador de Bienestar (PK_Nombre_indicador, Descripcion, Categoria)

Estadios (PK(Nombre_Estadio, FK_Comuna), FK_Nombre_Indicador)

CentrosMedicos (PK(Nombre_Centro, FK_Comuna), FK_Nombre_Indicador)

Escuelas (PK(Nombre_Escuela,FK_Comuna), FK_Nombre_Indicador)

DMCS (PK(FK_Nombre_Comuna), tasa, FK_Nombre_Indicador)

Diccionario de Datos:

Nombre de tabla Pais

pk	Atributo	Tipo de dato	Tamaño	fk	tabla
x	Nombre	VARCHAR	30		

Nombre de tabla Region

pk	Atributo	Tipo de dato	Tamaño	fk	tabla
x	Nombre_Region	VARCHAR	100		
	Nombre_Pais	VARCHAR	30	x	Pais

Nombre de tabla Comuna

pk	Atributo	Tipo de dato	Tamaño	fk	tabla
x	Nombre_Comuna	VARCHAR	100		
	Nombre_Region	VARCHAR	100	x	Region

Nombre de tabla Tener

pk	Atributo	Tipo de dato	Tamaño	fk	tabla
x	**ID**	**SMALL INT**	**6**		
	Año	YEAR	4		
	Valor	SMALL INT	1000000		
	Nombre_Comuna	VARCHAR	100	x	Comuna
	Nombre_indicador	VARCHAR	100	x	Indicador de bienestar

Nombre de tabla Indicador de Bienestar

pk	Atributo	Tipo de dato	Tamaño	fk	tabla
x	Nombre_Indicador	VARCHAR	100		
	Descripcion	VARCHAR	300		
	Categoria	VARCHAR	100		

Nombre de tabla Estadios

pk	Atributo	Tipo de dato	Tamaño	fk	tabla
x	Nombre_Estadio	VARCHAR	200		
x	Comuna	VARCHAR	100	x	
	Nombre_indicador	VARCHAR	100	x	

Nombre de tabla CentrosMedicos

pk	Atributo	Tipo de dato	Tamaño	fk	tabla
x	Nombre_Centro	VARCHAR	200		
x	Comuna	VARCHAR	100	x	
	Nombre_indicador	VARCHAR	100	x	

Nombre de tabla Escuelas

pk	Atributo	Tipo de dato	Tamaño	fk	tabla
x	Nombre_Escuela	VARCHAR	200		
x	Comuna	VARCHAR	100	x	
	Nombre_indicador	VARCHAR	100	x	

Variables a utilizar

Las variables que vamos a utilizar en nuestra base de datos son entretención, salud, educación y seguridad.

De cada una de estas variables guardaremos su nombre, descripción del lugar, como ha cambiado al pasar los años y su valor. Las elegimos porque son importantes para cada comuna, ya que si un usuario desea informarse sobre una región específica querrá buscar información en un sitio simple y que abarque la información más importante, como son las variables que ocupamos, en donde sus datos específicos será la cantidad de locales, sus direcciones, disponibilidad de trabajo, la seguridad, educación para la familia, etc. Entonces al saber todos estos datos relevantes de cada comuna, podrá decidir fácilmente sobre qué comuna le conviene más vivir gracias a la ayuda de nuestra base de datos.

Documentar Proceso de Descarga

Para tablas Pais y Region se escribieron los datos manualmente.

Para conseguir todas las comunas de Chile, manualmente las copiamos en un archivo de texto, luego se hizo un programa de Python llamado procesaComunas.py para poder escribirlas en un formato de arreglo (según las instrucciones) para luego usar getData.py y transformarlo en un archivo csv.

Para métrica de entretención, nos dispusimos a calcular la cantidad de estadios por comuna, para obtener los datos accedimos a el siguiente link: Anexo:Estadios de fútbol de Chile - Wikipedia, la enciclopedia libre. Gracias a que los datos ya estaban correctamente tabulados, bastó hacer una simple importación con Excel para obtener la tablas. Ahora con los datos en la planilla, el mismo Excel nos permitió eliminar las columnas de más (Nos quedamos con ciudad, nombre y capacidad) y anexar todo, además de eliminar los paréntesis para poder tener el nombre de la comuna limpio en cada fila. Guardamos el archivo como un csv y usamos un pequeño programa de Python que llamamos quita.py el cual devuelve un formato correcto del texto. El código es el siguiente:

```
import unicodedata

import codecs

def remove_accents(text):

    return ''.join(c for c in unicodedata.normalize('NFD', text) if
unicodedata.category(c) != 'Mn').replace('"',"").replace("'","")

file = codecs.open("estadios_Salida.csv","r","utf-8-sig")

text = "".join(file.readlines())

output = codecs.open("estadios_Salida_SIN_TILDES.csv","w","utf-8-sig")

output.write(remove_accents(text))

file.close()

output.close()
```

Este código normaliza los caracteres UTF-8 y elimina las comillas. Guardamos el archivo de salida como estadios_Salida_SIN_TILDES.csv.

De esta manera, podemos usar el valor para comparar directamente cuantos estadios tienen dos comunas, o podríamos comparar manualmente estadios/habitantes para tener una métrica porcentual para comparar dos comunas con un número de habitantes muy distinto. De lo anterior no guardamos el número de habitantes, ya que la cantidad varía mucho año a año, entonces dependerá de la persona que quiera hacer las comparaciones estimar la población comunal (Podríamos agregar el valor habitante a la tabla comunas en una entrega futura si lo consideramos necesario).

Como métrica de seguridad, quisimos contabilizar los delitos DMCS (Delitos de mayor connotación social) tales como homicidio, lesiones, violación, robo con violencia, etc. Para poder hacer una correcta comparación tomaremos la tasa de incidencia cada 100 000 habitantes, de esta manera podemos comparar varias comunas sin importar su cantidad de habitantes.

Extrajimos la información de la plataforma pazciudadana.cl. Desde la pestaña de Ranking comunal se nos proporciona un acceso directo a estas estadísticas a través de un documento Excel. Lo descargamos directamente e hicimos un proceso similar al anterior:

1. Eliminamos las columnas que no aportaban a la investigación. (Nos quedamos con Comuna y Tasa cada 100mil)
2. Guardamos el archivo usando una extensión csv
3. Ejecutamos quita.py para normalizar el texto, eliminando tildes y comillas
4. Ahora tenemos un archivo csv listo para ser leído.

Para comparar la educación, accedimos a Planes y Programas de estudio – Datos Abiertos (mineduc.cl) y descargamos los planes y programas de estudios de 2022, entro de este archivo comprimido había un csv que contenía información acerca de todos los planes de estudios del país, pero lo que nos interesaba era principalmente la cantidad de escuelas por comuna, entonces había que descartar el resto de la información. Debido al gran tamaño del archivo nos vimos con dificultades, ya que ningún editor de texto común, ni tampoco Excel podían abrirlo sin producir una pérdida de datos, ya que debían cargar más de 200MB. La solución fue hacer un algoritmo en python que lo procesara procedural mente, para que solo queden las columnas que nos interesan (Nombre escuela y comuna) que elimina todas las instituciones educativas que ya hayan aparecido en el archivo. El algoritmo es el siguiente:

```
import codecs

output = codecs.open("planesYProgramas_Salida.csv","w","utf-8-sig")

with
codecs.open("20230307_Planes_y_programas_de_estudios_2022_20220131_PUBL.csv","
r","utf-8-sig") as file:

    file.readline() # Elimina la primera fila (Nombres de columnas)

    line = file.readline()

    nombresColegio = set()

    while (line):

        columnas = line.split(";")

        nombre = columnas[3]

        if(nombre not in nombresColegio):

            columnas_importantes = ";".join([columnas[3],columnas[10]])

            output.writelines(columnas_importantes)

            output.write("\n")

            nombresColegio.add(nombre)

        line = file.readline()

output.close()

print("Terminado!")
```

Luego, había que quitar los tildes y las comillas (si existían), por lo que usamos quita.py, resultando en la salida esperada. Además, colocamos los nombres de las columnas a mano, en la primera fila.

Como nota, puede que no se justifique tanto que DMCS sea una tabla aparte, pero su existencia hace que la base de datos sea más escalable, de esta manera si queremos que se guarde información adicional podemos modificar esta tabla sin afectar la tabla comuna

Por último, como métrica de salud decidimos ver la cantidad de centros médicos por comunas.

Obtuvimos la información a través del MINSAL más concretamente: Listado De Establecimientos (minsal.cl). Esta página contiene una tabla que tiene los datos que buscamos, los que son nombre y comuna. Ya que la información ya está tabulada, usamos Excel para importarla directamente desde la página al igual como hicimos previamente. En Excel, había errores en los que algunos establecimientos (cantidad diminuta) que estaban digitados dos veces, tenían el mismo nombre exacto y en la misma comuna, metrics.py se encarga de ignorar los duplicados así que no hubo problema.

Para subir los archivos a la base de datos usamos metrics.py, que además vuelve a validar las entradas, por ejemplo, revisando nuevamente por comillas.

Con la base de datos poblada, como backup hicimos un dump que se encuentra en la carpeta baseDeDatos, al que le agregamos las líneas CREATE DATABASE bienestar y USE bienestar para poder importar la base de datos directamente, solo escribiendo source seguido de la ruta del archivo dump en Mariadb.