IELE756 — Ciencia de Datos

Migración y Salud en Chile: Tres Datasets, Una Pregunta

Leo Ferres

Semana 0 · 5–6 de marzo, 2026

¿Quiénes somos?

Tu nombre, tu formación,

y **una cosa** que esperas aprender.

∼30 segundos cada uno.

La pregunta

Tres números

$$2\% \rightarrow 8\%$$

Población nacida en el extranjero en Chile
2002 $\rightarrow$ 2024

Tres números

333 000

Registros de enfermedades de notificación obligatoria
(ENO, 2007–2024)

Tres números

5 millones

Episodios de egresos hospitalarios
(GRD, 2019–2024)

¿Por qué salud?

La migración se estudia por mercado laboral, vivienda, educación.

Pero **la salud es la pieza faltante**:

- ¿Cambia el paisaje de enfermedades cuando llegan 500 000 personas?
- ¿Los inmigrantes usan el hospital de forma distinta?
- ¿Hay comunas con más TB **y** más inmigrantes — coincidencia o patrón?

Chile tiene algo inusual: los datos para investigarlo.

Tres datasets

Censo 2024

19 millones de personas
Edad, sexo, nacionalidad, educación, empleo

Formato: parquet

ENO

333 000 notificaciones
TB, VIH, hepatitis, dengue, sarampión...

Formato: CSV (;)

GRD

5 millones de egresos
Diagnóstico (CIE-10), severidad, estadía

Formato: TXT (|)

Análisis ecológico

Punto clave

Estas son **personas distintas**. No rastreamos individuos entre datasets.

Pero **sí** podemos preguntar:

> *En comunas con más inmigrantes, ¿vemos patrones de enfermedad diferentes? ¿Tasas de hospitalización distintas?*

Esto es **análisis ecológico**: unidad = comuna, no persona.

Aprender cuándo es válido (y cuándo no) es un objetivo central del curso.

Cómo funciona el curso

Formato invertido

Este **no** es un curso de cátedra tradicional.

1. Libero material + tarea cada semana
2. Ustedes estudian **antes** de la clase
3. En clase: discutimos, debuggeamos, profundizamos

Si vienen sin preparar $\rightarrow$ perdidos.
Si vienen preparados $\rightarrow$ clase ultra productiva.

Ritmo semanal

Lunes: material disponible
Jueves: discusión + avance
Viernes: taller práctico

Parejas y regiones

- Trabajan en **parejas** todo el trimestre
- Cada pareja recibe una **región** de Chile
- Su ejemplo a lo largo del semestre: la demografía, inmigración y salud de *su* región

Asignaré parejas y regiones la próxima semana.

Si tienen preferencia de compañero/a, escríbanme por correo.

Evaluación

Componente	Puntos
Participación	10
Tarea 0 (setup)	5
Tarea 1 (demografía + migración)	10
Tarea 2 (salud: ENO + GRD)	10
Tarea 3 (modelamiento ecológico)	10
Proyecto final: repositorio GitHub	15
Proyecto final: video (8–12 min)	10
Total	**70**

Las tareas son **acumulativas**: T0 → T1 → T2 → T3 → Proyecto final.

Política de IA

- Herramientas de IA **están permitidas**
- Pero ambos miembros de la pareja deben entender todo el código
- Si les pregunto por una línea y no pueden explicarla $\rightarrow$ problema

Regla simple

Si la IA escribió el código y tú no lo entiendes, no es tu código.

Manos a la obra

Google Colab

1. Ir a `colab.research.google.com`
2. Iniciar sesión → **New Notebook**
3. Entorno Python gratis, en la nube, sin instalar nada

Escriban esto en la primera celda y presionen **Shift+Enter**:

```
import pandas as pd
print("Hello, IELE756!")
print(f"pandas version: {pd.__version__}")
```

Celdas Markdown

Agreguen una celda de texto (+ **Text**) con:

```
# My First Notebook
**Name**: [tu nombre]
**Date**: March 5, 2026

This is a *Markdown* cell. You can use:
- **bold**, *italics*
- bullet lists
- LaTeX: $\bar{x} = \frac{1}{n}\sum x_i$
```

Cada notebook que entreguen debe tener Markdown explicando qué hacen y por qué.

Código sin explicación no es análisis.

GitHub

- Si no tienen cuenta: `github.com` $\longrightarrow$ crear una ahora
- Usen un nombre de usuario profesional
- Cada pareja creará un repositorio para el proyecto

Lo configuraremos formalmente en Tarea 0.

Primer DataFrame real

```
import pandas as pd

data = {
    "nombre": ["Ana", "Boris", "Camila", "Diego", "Elena"],
    "edad": [28, 35, 42, 19, 55],
    "nacionalidad": ["Chilena", "Venezolana", "Chilena",
                     "Haitiana", "Chilena"],
    "comuna": ["Iquique", "Alto Hospicio", "Iquique",
               "Alto Hospicio", "Pozo Almonte"],
}
df = pd.DataFrame(data)
print("Shape:", df.shape)
print(df)
print(df["nacionalidad"].value_counts())
```

Filtro booleano

```
extranjeros = df[df["nacionalidad"] != "Chilena"]
print(extranjeros)
```

- Patrón básico: cargar → inspeccionar → filtrar → contar
- Los datasets reales tienen millones de filas, pero las operaciones son las mismas
- Van a escribir cientos de filtros como este

Para mañana

1. **Colab funcionando** — si tuvieron problemas, resuelvan hoy
2. **Traer laptop** — mañana es 100% práctico
3. Mañana cargamos los **tres datasets reales**

Nos vemos mañana a las 11:10.

Los tres datasets

Recap + hoja de ruta

Ayer: tres datasets, tres números, Colab listo.
Hoy: cargamos y exploramos cada uno.

Hora	Tema
11:15 – 12:00	Dataset 1 — Censo 2024
12:00 – 12:10	*Pausa*
12:10 – 12:50	Dataset 2 — ENO (enfermedades notificables)
12:50 – 13:30	Dataset 3 — GRD (egresos hospitalarios)
13:30 – 13:40	Todo converge en la comuna + Tarea 0

Ejemplo de hoy: **Tarapacá** (Región 1) — alta proporción de inmigrantes.

Censo 2024

¿Qué es el Censo?

Conteo completo de personas, hogares y viviendas en Chile (INE, 2024).

Tres tablas:

- **vivienda** — características físicas
- **hogar** — grupo que vive y come junto
- **persona** — demografía individual

Vinculadas: `id_vivienda` → `id_hogar` → `id_persona`

Formato

Archivos **parquet**:
columnar, comprimido, rápido.

Tabla persona: ∼19M filas.

Cargar el Censo

```python
import pandas as pd

persona = pd.read_parquet(
    "personas_censo2024.parquet",
    columns=["region", "comuna", "sexo", "edad",
             "p27_nacionalidad", "p27_nacionalidad_rec",
             "escolaridad", "sit_fuerza_trabajo"],
)
print(f"Total personas: {len(persona):,}")
print(persona.dtypes)
```

- Parquet permite seleccionar columnas → no cargamos las 50+
- Millones de filas en segundos (prueben eso con CSV)

Inspeccionar

```
print(persona.shape)
persona.head(10)
```

```
persona.info()
```

Sus dos mejores amigos al explorar datos nuevos:

- `.shape` $\longrightarrow$ dimensiones (filas, columnas)
- `.head()` $\longrightarrow$ primeras filas
- `.info()` $\longrightarrow$ tipos de dato y nulos

Filtrar a Tarapacá

```
# region es numérico en el Censo
tarapaca = persona[persona["region"] == 1]
print(f"Tarapacá: {len(tarapaca):,} personas")
```

De 19 millones a unos cientos de miles.
En sus tareas, filtran a **su** región asignada.

Nacionalidad

```
# Códigos: 1=Chileno, 2=Chileno+otra, 3=Extranjero, -99=NR
print(tarapaca["p27_nacionalidad"].value_counts())
```

```
# Variable recodificada (más conveniente)
print(tarapaca["p27_nacionalidad_rec"].value_counts())

foreign = tarapaca["p27_nacionalidad_rec"] \
    .value_counts(normalize=True)
print(f"% extranjeros: {foreign.get('Extranjero', 0):.1%}")
```

Datos crudos = códigos numéricos → necesitan el diccionario de datos.

Censo: lecciones

1. Datos en parquet — rápido, permite elegir columnas
2. 19M filas para todo Chile; filtran a su región
3. Variables codificadas con números $\rightarrow$ diccionario de datos
4. Operaciones clave: `read_parquet`, `filter`, `value_counts`

Trampas comunes

- `-99` = dato faltante, no "menos 99"
- Región es entero (`1`), no string
- Comuna es código numérico (`1101`), no nombre

Pausa — 10 minutos

Cuando volvamos: enfermedades.

ENO — Enfermedades de Notificación Obligatoria

¿Qué es ENO?

Cuando un médico diagnostica ciertas enfermedades (TB, VIH, hepatitis, dengue…), está **legalmente obligado** a notificar a la autoridad sanitaria.

- Cada notificación = una fila: enfermedad, fecha, comuna, nacionalidad, edad, sexo
- Período: 2007–2024
- $\sim$333 000 registros
- CSV delimitado por **punto y coma** (;)

Cargar ENO

```
eno = pd.read_csv(
    "materials/eno/20241218_base_eno_final.csv",
    sep=";", encoding="utf-8-sig")
print(f"Total: {len(eno):,}")
print(eno.columns.tolist())
```

```
eno.head()
```

Observen: `region` es texto (`"Región de Tarapacá"`), no número.

Filtrar ENO a Tarapacá

```
eno_tar = eno[eno["region"] == "Región de Tarapacá"]
print(f"Tarapacá: {len(eno_tar):,}")
```

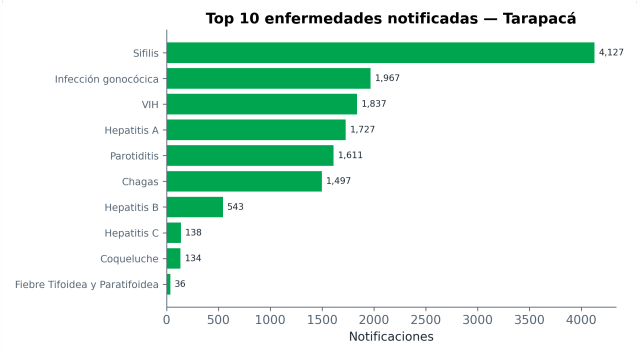
¡Ojo!

Censo: `region == 1` (número).
ENO: `region == "Región de Tarapacá"` (texto).

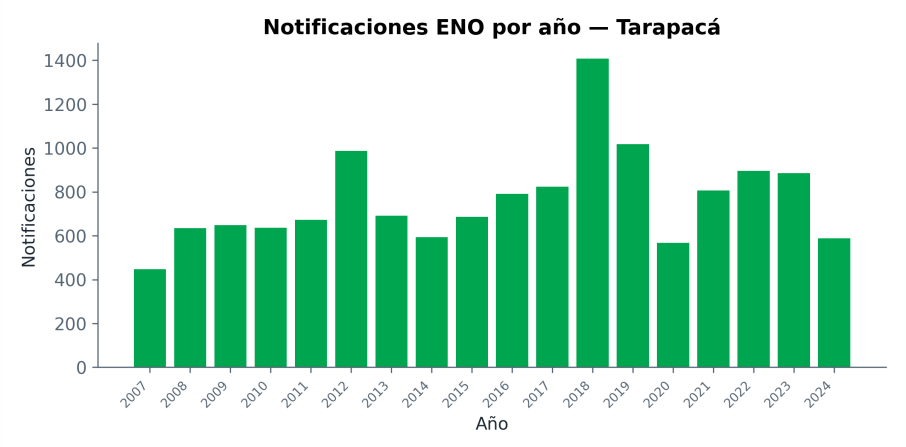
Distintas agencias, distintas codificaciones. Bienvenidos a datos reales.

Top 10 enfermedades — Tarapacá

```
eno_tar["ENO"].value_counts().head(10)
```



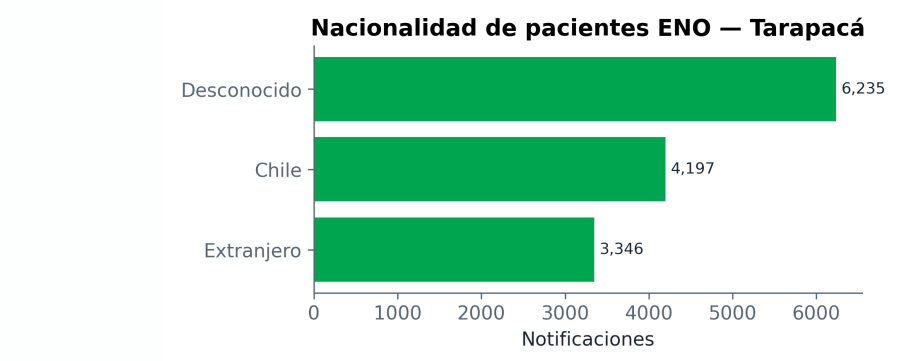
Notificaciones por año — Tarapacá



¿Ven tendencias? ¿Saltos? Piensen qué estaba pasando en Chile — y en el mundo — en esos años.

Nacionalidad de pacientes — Tarapacá

```
eno_tar["nacionalidad"].value_counts()
```



"Desconocido" es importante: ¿lo incluyen, excluyen, reportan aparte? Decisión analítica que deben justificar.

ENO: lecciones

1. Separador: `sep=";"` (no coma)
2. Región es texto, no número
3. Nacionalidad: Chile / Extranjero / Desconocido
4. Dimensión temporal: 17 años de vigilancia
5. Columnas clave: `ENO`, `anho_notificacion`, `region`, `nacionalidad`

Ejercicio: 10 minutos

1. ¿Cuál es la enfermedad más común en Tarapacá para extranjeros?

```
eno_tar[eno_tar["nacionalidad"] == "Extranjero"] \
    ["ENO"].value_counts().head()
```

2. ¿Cuántas comunas únicas hay en ENO Tarapacá?

```
eno_tar["codigo_comuna_residencia"].nunique()
```

GRD — Grupos Relacionados de Diagnóstico

¿Qué es GRD?

Base de egresos hospitalarios del sistema público chileno.

Cada registro:

- Diagnóstico principal (código CIE-10)
- Comuna del paciente, nacionalidad, sexo
- Severidad del caso, días de estadía

- Archivos delimitados por **pipe** (`|`), comprimidos en ZIP
- Un archivo por año; 2024 tiene $\sim$1M filas y **129 columnas**
- Codificación: `latin-1`

Cargar GRD (solo columnas necesarias)

```python
import zipfile

cols = ["COMUNA", "NACIONALIDAD", "SEXO", "DIAGNOSTICO1",
        "FECHA_INGRESO", "FECHAALTA",
        "IR_29301_SEVERIDAD", "IR_29301_COD_GRD"]

with zipfile.ZipFile("materials/grd/GRD_PUBLICO_2024.zip") as z:
    with z.open("GRD_PUBLICO_2024.txt") as f:
        grd = pd.read_csv(f, sep="|", usecols=cols,
                          low_memory=False)

print(f"Total egresos 2024: {len(grd):,}")
```

`usecols`: 8 de 129 columnas $\rightarrow$ ahorra memoria y tiempo.

Filtrar GRD a Tarapacá

```
comunas_tarapaca = ["IQUIQUE", "ALTO HOSPICIO",
    "POZO ALMONTE", "HUARA", "CAMIÑA",
    "COLCHANE", "PICA"]
grd_tar = grd[grd["COMUNA"].isin(comunas_tarapaca)]
print(f"Tarapacá: {len(grd_tar):,}")
```

`COMUNA` es **texto en mayúsculas** — ni código numérico (Censo) ni texto largo (ENO). Tercer formato distinto.

No hay columna `region`: hay que saber qué comunas pertenecen a cada región.

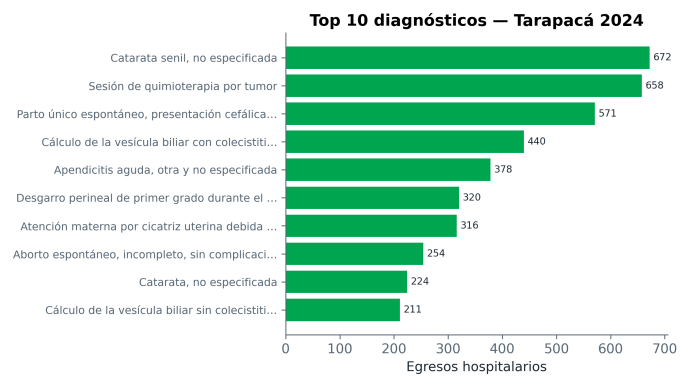
Join con CIE-10

```
cie10 = pd.read_excel("materials/grd/CIE-10.xlsx",
                      sheet_name="CIE 10")

grd_tar = grd_tar.merge(
    cie10[["Código", "Descripción", "Capítulo"]],
    left_on="DIAGNOSTICO1", right_on="Código",
    how="left")
```

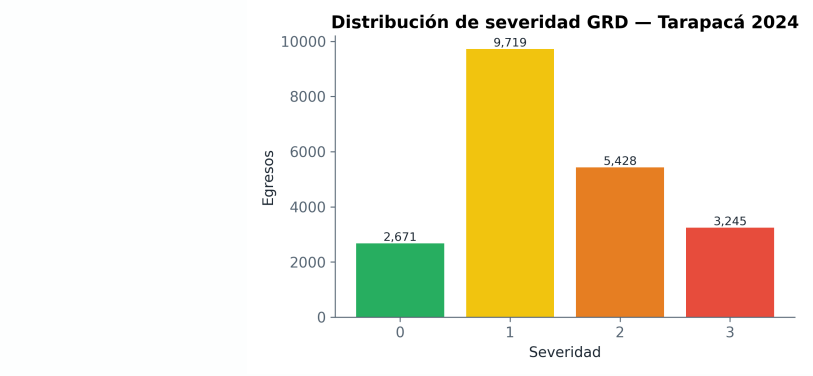
- CIE-10 traduce códigos (“J18.9”) a nombres (“Neumonía, no especificada”)
- Un **join**: vinculamos dos tablas por una clave compartida
- Operación fundamental en análisis de datos

Top 10 diagnósticos — Tarapacá 2024



¿Qué aparece? ¿Partos? ¿Neumonía? ¿Apendicitis?

Distribución de severidad — Tarapacá 2024



0 = Sin gravedad, 1 = Menor, 2 = Moderada, 3 = Mayor.
La mayoría es baja severidad, pero los casos nivel 3 consumen más recursos.

GRD: lecciones

1. Pipe-delimited, zipped, 129 columnas $\rightarrow$ siempre `usecols`
2. `COMUNA` es texto en mayúsculas, no código numérico
3. Diagnósticos en CIE-10 $\rightarrow$ necesitan la tabla de lookup
4. Joins son esenciales: GRD + CIE-10 = resultados comprensibles

Las tres codificaciones

Dataset	Separador	Codificación	Región / Comuna
Censo 2024	parquet (columnar)	UTF-8	Código numérico: `region==1`
ENO	;	UTF-8-sig	Texto: `"Región de Tarapacá"`
GRD	|	latin-1	Nombre comuna: `"IQUIQUE"`

No es un bug — es la vida real. Diferentes agencias, diferentes décadas, diferentes estándares.

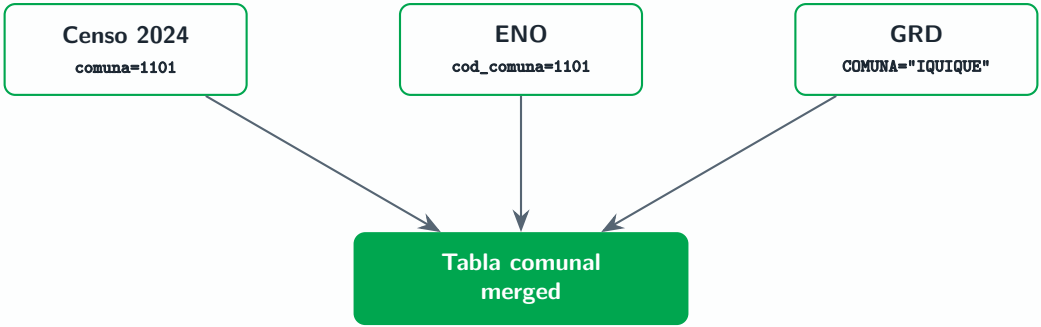
Armonizar estas representaciones es una habilidad central del curso.

Todo converge en la comuna

La idea central

Tres datasets, personas distintas, formatos distintos.

Pero comparten una cosa: geografía.



La tabla que construirán

Código	Comuna	Pob.	% Ext.	Tasa notif.	Tasa hosp.	Sev. media
1101	Iquique	250 000	15%	12.3	45.7	1.2
1107	Alto Hospicio	150 000	25%	18.1	52.3	1.4
1401	Pozo Almonte	16 000	8%	5.2	28.9	0.9

Valores ilustrativos — los calcularán en Tarea 3

Y luego preguntarán:

> *¿El porcentaje de inmigrantes en una comuna predice la tasa de notificación? ¿La tasa de hospitalización? ¿La severidad?*

Eso es **regresión ecológica** — y es la Tarea 3.

Tarea 0

Entrega: jueves 12 de marzo

1. Crear cuenta GitHub (si no tienen)
2. Crear repositorio de equipo
3. Cargar cada dataset para su región y reportar:
   - Censo: `shape`, `info`, primeras filas
   - ENO: notificaciones por año, top 5 enfermedades
   - GRD: descomprimir, cargar, join CIE-10, top 5 diagnósticos
4. Entregar notebook (PDF) en Canvas con link al repo

El código de hoy es $\sim$80% de la Tarea 0. Solo cambien Tarapacá por su región.

19M + 333K + 5M

En una clase. Eso es el poder de pandas.

Nos vemos la próxima semana — vengan con la Tarea 0 lista.