

**Ubicación: Castellón de la plana
España Septiembre, 2025**



Requerimientos Funcionales - Análisis Epidemiológico Por Provincia y Día

Autora:

Verónica Balza Herrera

Analista de Datos | Ingeniería Civil | Python | Estrategia de Negocio

Objetivo del Proyecto

Desarrollar un sistema de análisis de datos epidemiológicos que permita visualizar y comprender la evolución de casos, hospitalizaciones, defunciones y pacientes en UCI por provincia y día de la semana. El sistema debe facilitar la toma de decisiones sanitarias y la comunicación de resultados mediante visualizaciones interactivas y reportes automatizados.

Alcance del Proyecto

El sistema debe:

- Procesar datos históricos desde un archivo CSV.
- Agrupar los datos por día de la semana y provincia.
- Calcular métricas acumuladas por categoría (defunciones, casos nuevos, hospitalizados, UCI).
- Exportar los resultados en formato JSON para su reutilización.
- Visualizar los datos mediante gráficos de barras y gráficos tipo “queso” (pie charts).
- Ofrecer un menú interactivo para explorar los datos por dimensión.
- Permitir la generación de informes visuales en herramientas como Gamma.

Requerimientos Funcionales

Carga y Procesamiento de Datos

- **Fuente de datos:** Archivo CSV (PP2 - Recurso Proyecto 3.csv) con delimitador ; y codificación latin-1.
- **Lectura estructurada:** Uso de `csv.DictReader` para convertir cada fila en un diccionario.
- **Conversión de fechas:** Uso de `dateutil.parser` para interpretar fechas y extraer el día de la semana.
- **Agrupación:** Datos agrupados por día y provincia usando `defaultdict`.

Limpieza y Validación

- Manejo de errores con `try-except` para evitar fallos por datos mal formateados.
- Eliminación de valores nulos o vacíos mediante conversión segura (`int(datos.get(..., 0) or 0)`).
- Normalización de nombres de provincias (`strip()`).

Exportación de Resultados

- Conversión de estructura `defaultdict` a `dict` estándar.
- Exportación a archivo `.txt` en formato JSON con codificación UTF-8.
- Validación de estructura JSON mediante carga y conteo de provincias por día.

Visualización de Datos

- **Gráficos de barras:** Comparación de métricas por día o provincia usando `seaborn`.

- **Gráficos tipo queso:** Distribución porcentual por provincia con matplotlib.pie.
- **Menú interactivo:** Navegación por consola para elegir métricas y dimensiones.

Requerimientos No Funcionales

- **Compatibilidad:** Python 3.9+ con librerías estándar (csv, json, pandas, matplotlib, seaborn, dateutil).
- **Documentación:** Comentarios explicativos en el código y estructura modular.
- **Escalabilidad:** Capacidad de procesar archivos CSV grandes sin pérdida de rendimiento.
- **Interactividad:** Menús por consola para facilitar la exploración sin interfaz gráfica.

Paneles de Control y Presentación

- Los datos pueden visualizarse en un **dashboard interactivo** (Power BI, Gamma).
- Se recomienda usar **Gamma.app** para generar informes animados y presentaciones dinámicas.
- Las visualizaciones deben incluir:
 - Comparativas por día de la semana
 - Distribución geográfica por provincia
 - Evolución de métricas clave (defunciones, hospitalizaciones, UCI)

Posibles Extensiones

- **Modelos predictivos:** Aplicar regresiones o series temporales para anticipar picos de casos.
- **Análisis económico:** Aplicar métricas como Lift, Confianza o RFM si se vincula con datos de consumo o comportamiento.
- **Automatización:** Generar reportes periódicos o alertas automáticas según umbrales definidos.

Entregables

- Script en Python con procesamiento y visualización.
- Archivo JSON con resultados agrupados.
- Capturas de gráficos generados.
- Informe de conclusiones con hallazgos clave.
- README documentado en el repositorio.