

Universidad del Desarrollo

Magíster en Data Science – Facultad de Ingeniería

Asignatura: Visualización de Datos y Storytelling

Profesor: Carlos Elías Pérez Pizarro

Tarea 2 – Visualización interactiva con Streamlit

Fecha de entrega: Domingo 23 de noviembre

Modalidad: trabajo grupal (3–4 personas)

Formato de entrega: link repositorio en github + link app de streamlit

🎯 objetivo general

Diseñar y desplegar una **aplicación web interactiva** que explore las **emisiones de CO₂ a nivel global** utilizando:

- datos de **Our World In Data (OWID)**,
- visualizaciones interactivas con **plotly**,
- una interfaz desarrollada en **streamlit**,
- y un flujo de trabajo versionado en **github**.

El foco está en la **combinación de buen diseño visual con implementación técnica reproducible** (código limpio, estructura de app, manejo de datos y control de versiones).

📊 datasets

Trabajarán, como mínimo, con el siguiente dataset de OWID:

1. Annual CO₂ emissions per country

- Columnas típicas:
 - `country` , `code` (ISO3), `year`
 - `co2` : emisiones anuales de CO₂ (millones de toneladas, según OWID)
- Cobertura: países y regiones agregadas (World, Asia, Europe, etc.) por año.

Opcionalmente, pueden complementar con otros datasets de CO₂ de OWID (per cápita, sectorial, consumo vs producción, etc.), siempre que:

- documenten claramente qué archivo usaron,
 - justifiquen por qué agrega valor a la app,
 - mantengan un flujo de datos consistente.
-

desafío

A partir del **código base del mapa de CO₂ con streamlit** revisado en clases, cada grupo debe:

1. crear un repositorio en github

- Un integrante del grupo debe crear un repositorio en github (público).
- El repositorio debe contener, como mínimo:
 - `app.py` (aplicación de streamlit),
 - carpeta `data/` con los csv utilizados,
 - cualquier módulo auxiliar (`utils/`, etc.),
 - un `README.md` con:
 - descripción breve de la app,
 - instrucciones para ejecutarla localmente,
 - versiones mínimas de python y librerías,
 - referencia a la fuente de datos (Our World In Data).

Se evaluará que el repositorio tenga **estructura razonable** y no sea solo un dump desordenado de archivos.

2. adaptar y extender el código base de streamlit

A partir del ejemplo visto en clases (app con mapa global de CO₂):

- Mantengan la idea de “**mapa base de países**” con shapefile de Natural Earth + códigos ISO3.
- Aseguren que:
 - cada año tenga un mapa con **todos los países visibles**,
 - los países sin datos para ese año se muestren en **gris** (y esta decisión se explique en la app),
 - el usuario pueda seleccionar el año con un **slider o selectbox**.

La app debe tener al menos:

- un **sidebar** con controles (selección de año, tipo de vista, filtros, etc.),
 - un **layout organizado** (columnas, tabs o secciones),
 - mensajes claros cuando no hay datos para ciertas combinaciones (sin errores en pantalla).
-

3. recrear (o adaptar críticamente) visualizaciones de CO₂ de Our World In Data

En lugar de indicar qué gráficos deben usar, la tarea es:

1. Navegar la página:

<https://ourworldindata.org/co2-emissions>

2. Identificar visualizaciones relevantes para responder preguntas analíticas sobre emisiones.
3. **Seleccionar, justificar y recrear** al menos **tres visualizaciones interactivas** basadas en lo que encuentren ahí.
 - pueden ser adaptaciones directas,
 - combinaciones de ideas,
 - o variaciones que ustedes consideren más adecuadas.

4. Integrarlas en la app de streamlit de forma coherente.

No se evalúa la fidelidad estética respecto a OWID, sino la **intención analítica** y la calidad técnica de la recreación.

4. hacer la app realmente interactiva (no solo estática)

Además de cambiar de año, la app debe incluir al menos:

- un control de **selección de países**,
- un control de **tipo de métrica** o modo de visualización (ustedes deciden cuál),
- y algún tipo de **estado compartido** entre gráficos (por ejemplo, el mismo año debe actualizar mapa y otras visualizaciones).

La interactividad debe tener un propósito claro, no ser decorativa.

5. explicar brevemente las decisiones de diseño dentro de la app

En una pestaña, sección o bloque del app, incluyan:

- listado breve de los datasets usados,
- aclaración de unidades y períodos,
- justificación de **dos decisiones de diseño relevantes** (paleta, escalas, agregaciones, normalizaciones, etc.),
- nota breve sobre limitaciones:
 - países sin datos,
 - años incompletos,
 - otros relevantes.



entregables

La entrega consiste **solo en dos URL**:

1. URL del repositorio público en GitHub

Debe contener todo el código, datos y documentación necesaria para reproducir la app.

2. URL de la app de Streamlit desplegada

En Streamlit Community Cloud.

Debe cargar y funcionar sin errores.

No se aceptan PDFs, zips, capturas de pantalla ni archivos adjuntos fuera del repositorio.

rúbrica de evaluación (1–100 puntos)

La nota total se calcula sobre 100 puntos.

Cada criterio tiene 3 niveles: **insuficiente, adecuado, sólido**.

1. rigor analítico y uso de datos (20 puntos)

Qué evalúa:

uso correcto de las variables, periodos y unidades; coherencia entre lo que muestra la app y lo que realmente dicen los datos.

Nivel	Descripción	Puntos
Insuficiente	Errores conceptuales importantes (unidades mal interpretadas, ejes sin contexto, años mezclados sin control). No se distingue entre datos faltantes y valores reales.	0–10

Nivel	Descripción	Puntos
Adecuado	Uso razonable de los datos, con algunos detalles poco claros (por ejemplo, falta explicar mejor qué es exactamente “co2”). No hay errores graves pero la lectura requiere esfuerzo.	11–16
Sólido	Variables, periodos y unidades están claros y son consistentes en toda la app. Se explica bien qué representa cada métrica y se trata explícitamente el tema de países sin dato.	17–20

2. diseño visual e interactividad (20 puntos)

Qué evalúa:

calidad de las visualizaciones en plotly, elección de escalas y paletas, y uso sensato de la interactividad.

Nivel	Descripción	Puntos
Insuficiente	Gráficos confusos o saturados; interactividad irrelevante o inexistente (apenas sliders sin impacto real). Colores o escalas dificultan la lectura.	0–10
Adecuado	Visualizaciones correctas pero algo genéricas. La interactividad ayuda pero podría estar mejor alineada con preguntas analíticas concretas.	11–16
Sólido	Visualizaciones claras, con buenas decisiones de escala y color. La interactividad está bien pensada: permite comparar, explorar y responder preguntas relevantes sin abrumar.	17–20

3. replicación / adaptación de gráficos de OWID (20 puntos)

Qué evalúa:

capacidad de recrear o adaptar de forma crítica los gráficos de la página de CO₂ de OWID en plotly.

Nivel	Descripción	Puntos
Insuficiente	Los gráficos se alejan mucho de las ideas de OWID o son versiones muy pobres (sin contexto, sin ejes claros, sin relación con la referencia).	0–10
Adecuado	Se reconocen claramente 2–3 tipos de gráficos inspirados en OWID, aunque con simplificaciones importantes. La intención visual se mantiene, pero podría mejorarse la fidelidad o detalle.	11–16
Sólido	Las visualizaciones capturan bien el espíritu de OWID (mapas, series, rankings, etc.), adaptadas a plotly con criterio. No son copias ciegas, sino versiones bien razonadas.	17–20

4. arquitectura de la app y experiencia de uso (20 puntos)

Qué evalúa:

cómo está organizada la app de streamlit, claridad de la interfaz, manejo de controles y estados compartidos.

Nivel	Descripción	Puntos
Insuficiente	App caótica: todo en una sola página sin estructura, controles que se pisan entre sí, errores frecuentes. Difícil entender qué hacer.	0–10
Adecuado	Estructura básica pero utilizable. Sidebar con algunos controles claros. Hay margen para mejorar organización y textos	11–16

Nivel	Descripción	Puntos
	explicativos.	
Sólido	App limpia y bien organizada. El usuario entiende qué puede hacer, cómo filtrar y cómo leer los gráficos. Controles y textos están alineados con el objetivo de la tarea.	17–20

5. trazabilidad técnica y uso de github / documentación (20 puntos)

Qué evalúa:

orden del repositorio, documentación mínima y evidencia de un flujo de trabajo reproducible.

Nivel	Descripción	Puntos
Insuficiente	Repositorio incompleto o desordenado. Falta README o es inservible. Cuesta reproducir la app. No se documenta el uso de IA.	0–10
Adecuado	Repositorio razonablemente claro, con README básico y archivos principales presentes. Se puede reproducir, aunque con algo de esfuerzo.	11–16
Sólido	Repositorio ordenado, con estructura clara, README útil y requisitos bien definidos. Se declara el uso de IA (si aplica) y hay trazabilidad técnica suficiente para confiar en el trabajo.	17–20



propósito pedagógico

Esta tarea marca la transición desde visualizaciones “de notebook” hacia la **comunicación interactiva** mediante aplicaciones web con datos reales, integrando:

- visualización,
 - diseño,
 - reproducibilidad,
 - y pensamiento crítico sobre decisiones analíticas.
-



entrega

Solo se deben entregar en aula virtual:

1. **URL del repositorio GitHub público,**
2. **URL de la app Streamlit desplegada.**

Un integrante del grupo realiza la entrega por todo el equipo.