PROYECTO FINAL:

1. Objetivo general del proyecto
Desarrollar una solución integral de análisis de datos: desde la recopilación y la ingesta de datos, pasando por la limpieza y exploración (EDA), hasta la creación y validación de un modelo de Machine Learning (y, si se desea, la integración de un modelo de lenguaje o algún componente de IA generativa).

Demostrar las capacidades técnicas y prácticas adquiridas durante el bootcamp, reflejando el entendimiento de conceptos de ingeniería de datos, análisis, modelado, despliegue en la nube y buenas prácticas de desarrollo.

2. Temática y alcance
Cada grupo tiene la libertad de elegir la temática que más le interese. Se puede trabajar con datos del área de marketing, salud, finanzas, deportes, medio ambiente, redes sociales, entre otros.

El proyecto debe cubrir todo el ciclo de vidadel dato:
Ingesta y/o recolección de datos (API, bases de datos públicas, web scraping, etc.).
Almacenamiento y tratamiento inicial (limpieza, conversión de formatos, verificación de calidad).

Exploración de datos (EDA), con visualizaciones y estadísticas descriptivas.
Desarrollo de modelo(s) de Machine Learning: regresión, clasificación, clustering, series temporales o un modelo de lenguaje (opcional).

Deploy o presentación de resultados en la nube o entorno local.
Reporte y documentación finales (conclusiones, insights y/o recomendaciones).

3. Roles y responsabilidades sugeridas

Cada grupo deberá organizarse y asignar a sus miembros distintos roles (una misma persona puede desempeñar más de uno si fuera necesario). Algunas posibles designaciones son:

Data Engineer

Responsable de la recolección, limpieza y transformación de los datos.
Definir la estructura del flujo (pipeline), manejar la conexión con bases de datos o servicios en la nube, y garantizar la calidad de los datos.

Data Analyst / Data Scientist

Liderar la exploración y el análisis estadístico y descriptivo (EDA).
Trabajar en el diseño y construcción de modelos predictivos/analíticos.
Generar visualizaciones y dashboards que evidencien el comportamiento y la naturaleza de los datos.

Machine Learning Engineer / MLOps

Encargarse de la puesta en producción y la automatización del pipeline de ML, si fuera requerido.

Considerar la arquitectura en la nube, el versionado y la escalabilidad del modelo.
Optimizar el rendimiento y la confiabilidad del sistema de Machine Learning.

Cloud / DevOps (opcional, pero recomendable)

Configurar la infraestructura en la nube para desplegar la solución.

Nota: Estas sugerencias no son excluyentes; un mismo integrante puede asumir más de un rol, siempre que el proyecto final abarque las áreas de responsabilidad mencionadas.

4. Requisitos técnicos mínimos

Repositorio de GitHub:
Código organizado y limpio.
Uso de branches y commits con mensajes claros.
Estructura de carpetas clara (por ejemplo: src/, notebooks/, data/, docs/).

Documentación:
Un archivo README.md que describa claramente el objetivo del proyecto, los integrantes del equipo, los pasos para reproducir el entorno y la estructura del repositorio.
Documentación de los procesos realizados (puede ser un archivo doc/ o cuadernos Jupyter con explicaciones).

Procesamiento y Limpieza de Datos:
Evidencias de cómo se obtuvieron los datos y de la metodología empleada para la limpieza y normalización.

EDA y Visualizaciones:
Se deben incluir gráficas y/o cuadros estadísticos que reflejen las principales conclusiones de la exploración de los datos.

Modelado y Resultados:
Al menos un modelo de machine learning o técnica estadística avanzada que justifique la elección del método.
Métricas claras de evaluación (RMSE, Accuracy, Precision, Recall, F1, etc., según corresponda).
(Opcional) Integración de un Modelo de Lenguaje:
Si el equipo lo desea, puede integrar un modelo de lenguaje (por ejemplo, para clasificación de texto, generación de resúmenes, chatbots o análisis de sentimiento).

Despliegue / Infraestructura:
Demostrar el uso de alguna plataforma en la nube o despliegue desde Streamlit cloud.

Conclusiones y Recomendaciones:
Incluir un apartado final con lecciones aprendidas, limitaciones y posibles mejoras futuras.

5. Entregables

Repositorio de GitHub con todo el código y la documentación.
Presentación(slides o notebook) con un resumen ejecutivo de:

- Objetivo y motivación del proyecto.
- Fuentes de datos y métodos de obtención.
- Resultados de la EDA.
- Modelos desarrollados y resultados (métricas, visualizaciones).
- Conclusiones, hallazgos relevantes y posibles siguientes pasos.
- Sugerencia: Se valora la creatividad en la forma de presentar los hallazgos (dashboards interactivos, reportes automatizados, aplicaciones web, etc.).

6. Criterios de evaluación

Calidad del código y organización (estructura de repositorio, legibilidad, uso de buenas prácticas).
Cumplimiento de objetivos (que se abarquen todos los aspectos de un proyecto de datos: ingesta, tratamiento, EDA, modelado, despliegue, conclusiones).
Originalidad y profundidad de la investigación (elección del dataset, complejidad de la problemática, justificación de las decisiones técnicas).
Claridad en la comunicación (documentación, explicaciones en el código o notebooks, presentación de resultados).
Trabajo en equipo (evidencia de contribuciones de cada miembro).

7. Recomendaciones finales

Comenzar el proyecto lo antes posible para reservar tiempo suficiente para la recopilación de datos, los ajustes del modelo y la documentación.
Hacer un plan de trabajo detallado (cronograma), asignando tareas y responsabilidades.
Usar control de versiones: cada cambio debe quedar registrado con mensajes de commit claros.
Mantener una comunicación fluida entre los integrantes y aprovechar las herramientas colaborativas (GitHub Projects, Issues, Slack, Trello, etc.).
Documentar todos los pasos: un proyecto de datos sin documentación pierde valor y dificulta su reproducción.