Operador en Análisis de Datos

INFORMACIÓN GENERAL

- **Fecha de inicio:** 06/04/2026
- **Modalidad:** Virtual
- **Horas totales de cursada:** 36hs
- **Días y horarios:** 12 encuentros. Lunes de 18a 21hs.
- **Capacitación gratuita**
- **Destinado a:** Personas con conocimientos básicos en programación.
- **Requerimientos:**
    - Ser socie de la MIT

OBJETIVOS

- Reconocer y diferenciar las diversas formas de captura de datos, así como los distintos tipos de fuentes disponibles para su posterior análisis, procesamiento y uso en la generación de información.
- Comprender en profundidad qué significa garantizar la consistencia y calidad de los datos, desarrollando criterios claros que permitan resguardarlos a lo largo de todas las etapas de procesamiento y análisis.
- Detectar y enfrentar problemas frecuentes vinculados a la calidad de los datos —como valores ausentes, registros atípicos o inconsistencias—, aplicando para ello técnicas y procedimientos adecuados de depuración y validación.
- Conocer y evaluar distintos modelos de representación de datos, analizando sus ventajas y limitaciones de acuerdo con el tipo de información que se busca mostrar, la posibilidad de jerarquizar variables y el grado de facilidad o complejidad que presentan para su lectura e interpretación.
- Diseñar y producir visualizaciones claras y significativas utilizando herramientas programáticas (matplotlib, seaborn, plotly) y plataformas de tableros de control.
- Integrar todas las etapas del ciclo de vida del dato en un proyecto de análisis completo, desde la captura hasta la comunicación de resultados.

CONTENIDOS

BLOQUE 1: Análisis de datos (24hs)

Clase 1. El ciclo de vida del dato (3 h)

- Etapas principales: captura, preprocesamiento, análisis y visualización.
- Relación entre cada fase y su impacto en la gestión de información.
- Tipos de datos: estructurados, semiestructurados y no estructurados.
- Primer contacto con Google Colab: estructura de un notebook, celdas de código y texto.
- Ejemplos introductorios en Python: variables, listas, diccionarios y operaciones básicas.

Herramientas utilizadas:

- Google Colab para exploración conceptual y ejecución de ejemplos en vivo.
- Python (introducción a tipos de datos y estructuras básicas).

Clase 2. Captura de datos: fuentes y formatos (3 h)

- Formatos de archivos de datos: CSV, JSON, Excel y texto plano.
- Lectura de archivos con pandas: read_csv(), read_json(), read_excel().
- Exploración inicial de los datos: head(), tail(), info(), describe(), shape.
- Fuentes de datos abiertos: portales nacionales (datos.gob.ar) e internacionales (Banco Mundial, Kaggle).

Herramientas utilizadas:

- Python (pandas) para lectura y exploración de archivos en distintos formatos.
- Google Colab para desarrollo y documentación de ejercicios prácticos.

Clase 3. Captura de datos: APIs y web (3 h)

- Introducción a las APIs: concepto, protocolo HTTP y método GET.
- La biblioteca requests para realizar solicitudes web desde Python.
- El formato JSON: estructura y navegación de respuestas anidadas.
- Ejemplo práctico: consultar la API del Banco Mundial para obtener indicadores de
- países de América Latina.
- Google Sheets como fuente de datos: lectura directa desde Python.
- Google Drive como almacenamiento colaborativo.

Herramientas utilizadas:

- Python (requests, pandas) para recolección desde APIs públicas.
- Google Sheets y Google Drive para trabajo colaborativo.

Clase 4. Preparación de datos: estructuras y organización (3 h)

- DataFrames en profundidad: creación desde diccionarios y listas.
- Selección de datos: columnas, .loc[] (por etiqueta) e .iloc[] (por posición).
- Introducción a las bases de datos relacionales y al lenguaje SQL.
- SQLite con Python: crear tablas, insertar datos y consultar con SELECT.
- Conexión entre pandas y SQL: pd.read_sql_query().

Herramientas utilizadas:

- Python (pandas, sqlite3) para estructuración y consultas a bases de datos.
- Google Colab para desarrollo y documentación de ejercicios prácticos.

Clase 5. Preparación de datos: limpieza y depuración (3 h)

- Detección de valores faltantes: isna(), notna(), visualización con seaborn.heatmap().
- Estrategias de tratamiento: eliminación (dropna()), imputación (fillna() media/mediana), propagación (ffill(), bfill()).

- Detección y eliminación de duplicados: duplicated(), drop_duplicates().
- Valores atípicos (outliers): método IQR y diagramas de caja (boxplots).
- Tratamiento de outliers: eliminación, acotamiento (capping) y transformación logarítmica.
- Conversión de tipos de datos: astype(), pd.to_numeric(), pd.to_datetime().

Herramientas utilizadas:

- Python (pandas, numpy, seaborn) para limpieza, depuración y transformación de datos.
- Google Colab para ejercicios prácticos documentados.

Clase 6. Preparación de datos: normalización y consistencia (3 h)

- Normalización de datos numéricos: escalado min-max y estandarización z-score.
- Manipulación de cadenas de texto: .str.lower(), .str.strip(), .str.replace(), .str.contains().
- Datos categóricos: exploración, corrección y estandarización de categorías.
- Manejo de fechas y horas: pd.to_datetime(), accesor .dt para extraer componentes.
- Combinación de conjuntos de datos: pd.merge() y pd.concat().
- Ejercicio integrador de las clases 4 a 6.

Herramientas utilizadas:

- Python (pandas, numpy) para normalización, transformación de texto y combinación de datasets.
- Google Colab para ejercicios prácticos documentados.

Clase 7. Análisis de datos: estadística descriptiva y reportes (3 h)

- Medidas de tendencia central: media aritmética, mediana y moda.
- Medidas de dispersión: rango, varianza, desviación estándar y rango intercuartílico (IQR).
- Tablas de frecuencias y distribuciones: value_counts(), pd.cut().

- Operaciones de agrupamiento: .groupby() y agregaciones múltiples con .agg().
- Tablas dinámicas con pd.pivot_table().
- Correlación entre variables: .corr() y su interpretación.
- Gráficos para el análisis descriptivo: histogramas, boxplots y mapas de calor.

Herramientas utilizadas:

- Python (pandas, matplotlib, seaborn) para análisis estadístico y generación de gráficos.
- SQL para extracción y filtrado de datos relevantes.
- Google Colab para ejercicios prácticos documentados.

Clase 8. Caso integrador: del dato al reporte (3 h)

- El flujo completo del análisis de datos aplicado a un caso real.
- Metodología para un proyecto de análisis: preguntas guía y estructura de un reporte descriptivo.
- Caso de estudio: indicadores socioeconómicos de América Latina.
- Carga y construcción del dataset.
- Exploración inicial.
- Limpieza y transformación (imputación por interpolación contextual).
- Análisis descriptivo y correlaciones.
- Visualización de resultados.
- Conclusiones y reporte textual.
- Consultas SQL para filtrado y agregación.
- Buenas prácticas y errores comunes en proyectos de datos.

Herramientas utilizadas:

- Python (pandas, numpy, matplotlib, seaborn, sqlite3) para el ciclo completo de análisis.
- SQL (SQLite) para consultas estructuradas.
- Google Colab para documentación del caso integrador.

BLOQUE 2: visualización de datos (12 horas)

Clase 9. Fundamentos de visualización (3 h)

- Semiótica aplicada a la visualización de datos.
- Principios de percepción visual: color, dimensiones, contraste y legibilidad.
- La gramática de gráficos: datos, geometrías, escalas y anotaciones.
- Fundamentos de matplotlib: estructura Figure y Axes, interfaz orientada a objetos.
- Gráficos básicos: líneas, barras, dispersión.
- Subgráficos (subplots) para comparaciones múltiples.
- Personalización: colores, etiquetas, títulos, leyendas y anotaciones.
- Seaborn como interfaz de alto nivel: relplot(), catplot(), regplot().
- Elección de colores y accesibilidad: tipos de paletas y consideraciones para daltonismo.

Herramientas utilizadas:

- Python (matplotlib, seaborn) para experimentar con representaciones visuales.
- Google Colab para ejercicios prácticos documentados.

Clase 10. Modelos de representación (3 h)

- Tipología de gráficos y su pertinencia según el objetivo comunicacional:
    - Gráficos de líneas: series temporales y tendencias.
    - Gráficos de barras: comparaciones categóricas.
    - Gráficos de torta: proporciones.
    - Gráficos de dispersión: relaciones entre variables.
    - Histogramas: distribuciones.
    - Mapas de calor: correlaciones y patrones.
- Análisis de ventajas, limitaciones y ejemplos de aplicación de cada tipo.
- Introducción a Plotly para gráficos interactivos: plotly.express y plotly.graph_objects.

Herramientas utilizadas:

- Python (matplotlib, seaborn, plotly) para la elaboración y comparación de distintos tipos de gráficos.
- Google Colab para ejercicios prácticos documentados.

Clase 11. Dashboards interactivos (3 h)

- ¿Qué es un dashboard? Principios de diseño: selección de KPIs, layout y jerarquía visual.
- Google Looker Studio: características, flujo de trabajo, ventajas y limitaciones.
- Introducción a Plotly Dash: arquitectura, componentes principales y callbacks.
- Estructura completa de una aplicación Dash con datos de indicadores económicos de Argentina.
- Gráficos interactivos avanzados con Plotly: menús desplegables en gráficos, subgráficos (subplots) con panel 2×2.
- Exportar gráficos como HTML para distribución sin servidor.
- Comparación de herramientas de dashboards: Looker Studio, Dash, Power BI, Streamlit.

Herramientas utilizadas:

- Python (plotly, dash) para visualizaciones programáticas avanzadas.
- Google Looker Studio para introducción a tableros de control no programáticos.
- Google Colab para ejercicios prácticos documentados.

Clase 12. Proyecto integrador final (3 h)

- Recorrido del curso: síntesis de las competencias adquiridas en los dos bloques.
- Metodología para un proyecto de análisis de datos: definición del problema, recopilación, limpieza, análisis, visualización y comunicación de resultados.
- Estructura de un informe de análisis de datos.
- Buenas prácticas: resumen integral del curso.
- Recursos para continuar aprendiendo: documentación oficial, comunidades y plataformas.
- Actividad práctica final: proyecto de punta a punta con dataset de indicadores socioeconómicos.

Herramientas utilizadas:

- Python (pandas, numpy, matplotlib, seaborn, plotly, sqlite3) para el proyecto integrador completo.
- Google Colab para documentación y presentación del trabajo final.

DOCENTE

Guillermo Leale (él)

Antecedentes académicos

Doctorado en Ingeniería de Sistemas, Universidad Tecnológica Nacional (UTN), Argentina. Tema y línea de trabajo: modelado, minería de datos, big data y simulación.

● Dirección de proyectos de investigación en Modelado, Minería de Datos y Big Data (UTN). Enfoques: aprendizaje supervisado/no supervisado, modelado probabilístico, simulación de eventos discretos, e integración ML–BI.

● Publicaciones seleccionadas relacionadas con el curso (ML):

○ Using Machine-Learning Models for Field-Scale Crop Yield and Condition Modeling in Argentina — aplicación de modelos ML con métricas satelitales para predicción de rendimiento y condición de cultivo.

○ Using Hybrid Bayesian Networks to Detect Audience Behaviour Changes in YouTube — redes bayesianas híbridas y MCMC para detección de cambios de comportamiento.

○ VIRTA-Yoke: A Virtual-Integrated Poka-Yoke System for Error Prevention and Operator Training in Manufacturing Processes — sistema con AR y visión por computadora para prevención de errores y entrenamiento operativo.

● Lista de publicaciones en https://scholar.google.com/citations?user=wZIyDVoAAAAJ

● Intereses actuales de investigación: RAG aplicado a dominios educativos y corporativos, evaluación de LLMs y agentes, métodos de validación estadística y evaluación de modelos en producción.

Experiencia laboral

● Radium Rocket – Responsable de Knowledge Management / Educación y Tech Manager en proyectos de desarrollo.

○ Diseño e implementación de arquitecturas de datos para analítica y ML (Azure/GCP/Snowflake/Databricks, pipelines ETL/ELT, gobierno y seguridad de datos).

○ Liderazgo técnico en iniciativas de Analytics/ML: definición de features,

evaluación de modelos, MLOps ligero, reporting de negocio y experimentación.
○ Proyectos de IA y agentes: prototipos RAG internos, asistentes para documentación técnica, evaluación de prompts y calidad de recuperación.
● Consultoría independiente – Servicios en Ciencia de Datos, BI e IA aplicada.

Proyección de stock y optimización de inventario con modelos predictivos.
Desarrollo de tableros BI y métricas de desempeño (Looker Studio/Power BI).
Integración de datos multimodales y preparación de datasets para ML.
Supervisión y seguimiento del ciclo completo de equipos de trabajo para proyectos de analítica y ML.
● Proyectos de I+D aplicados
○ Modelos de recomendación y scoring de acciones sobre grandes volúmenes de datos.
○ Detección de comportamiento y segmentación con enfoques probabilísticos y de ML clásico.
○ Predicción de rendimiento y fenología en cultivos con técnicas de ML

Experiencia docente
● Universidad Tecnológica Nacional (UTN) – Profesor titular/adjunto en carreras de grado y posgrado.
○ Cátedras: Simulación, Investigación Operativa, Ciencia de Datos y afines.
○ Dirección de proyectos y tesis en ML, minería de datos y analítica aplicada.
● Maestrías y posgrados – Coordinación y dictado de módulos de Ciencia de Datos, Minería de Datos y herramientas para posgrados en industria y humanidades aumentadas.
○ Talleres prácticos con Orange, Python y ecosistema científico (NumPy/Pandas/Scikit-learn).
● Cursos y capacitaciones profesionales
○ Curso intensivo de "Machine Learning con Python y OpenCV" (6 horas): ML tradicional (regresión, clasificación, métricas, validación) e introducción a procesamiento de imágenes.
○ Entrenamientos en Business Intelligence y Data Engineering (arquitectura de DW, ETL, modelado dimensional, analítica de negocio).
○ Formación y mentoreo de equipos en uso de IA generativa, RAG y

evaluación de calidad de respuestas.

○ Curso de “Introducción al Machine Learning y Agentes LLM”