UNIVERSIDAD TECNOLÓGICA METROPOLITANA
UTEM del Estado de Chile

INFB6052 - HERRAMIENTAS PARA CS. DE DATOS
INGENIERÍA CIVIL EN CIENCIA DE DATOS

PRIMERA PRUEBA:

Primera Prueba: Ejercicios prácticos utilizando herramientas de Ciencia de Datos y propuesta de anteproyecto del Proyecto Integrador.

Dr. Ing. Michael Miranda Sandoval

Segundo Semestre de 2025

Instrucciones Generales

Esta prueba está compuesta por dos partes que deben entregarse y documentarse claramente en el repositorio GitHub del curso:

1. Primera Parte (Práctica): un conjunto de ejercicios prácticos de análisis e ingeniería de datos (ver sección 1). Cada ejercicio pide implementación, experimentos y conclusiones.
2. Segunda Parte (Anteproyecto): proposición del anteproyecto del Proyecto Semestral – Integrador (ver sección 2). Debe describir viabilidad, metodología y cronograma.

Los resultados de la prueba deben demostrar tanto habilidades técnicas (pipeline en Python, manejo de datos, visualización y modelado básico) como buenas prácticas de ingeniería (control de versiones y documentación).

Lineamientos generales obligatorios para todos los ejercicios y anteproyecto:

- El desarrollo de esta prueba es grupal con los grupos definidos previamente en las clases.
- Los grupos deben señalar claramente sus integrantes en un archivo tipo Markdown (README.md) en el repositorio GitHub. En este archivo se debe explicar la colaboración individual de cada integrante en el desarrollo de la prueba.
- El desarrollo de cada ejercicio debe estar en un pipeline completo en lenguaje Python: un único archivo Jupyter Notebook (.ipynb) por ejercicio. Cada notebook debe subirse al repositorio GitHub y contener celdas con explicación, código y resultados (gráficos/tablas). No se aceptan fragmentos sueltos sin contexto.
- Se debe manejar control de versiones en Git: clonar el repositorio del curso, crear ramas para trabajo aislado, realizar commits descriptivos y finalmente hacer pull/merge según corresponda. Incluya en el README del repositorio instrucciones breves del flujo de trabajo usado.
- Está permitido el uso de asistentes de código (por ejemplo, GitHub Copilot), pero el código entregado debe estar plenamente comentado y explicado por el/la estudiante; las decisiones de diseño y las adaptaciones deben quedar registradas en el notebook y en el informe LaTeX.
- Elaborar un informe en LaTeX (archivos fuente + PDF compilado) que resuma el proceso, decisiones, resultados y reflexiones sobre los ejercicios y el anteproyecto. El informe (máx. 10 páginas para el anteproyecto) debe subirse al mismo repositorio GitHub junto con los notebooks y archivos auxiliares.
- Los grupos pueden usar cualquier dataset público o privado, siempre que se indique claramente la fuente y se asegure acceso para la evaluación. Se recomienda usar datasets de tamaño moderado (100 MB - 1 GB) para facilitar pruebas y experimentos.
- Los grupos pueden realizar las suposiciones que consideren necesarias, siempre que las expliquen claramente en el notebook y en el informe LaTeX. Justifique las decisiones tomadas y cualquier limitación encontrada.
- El incumplimiento de alguna de estas directrices implica penalización en puntaje de la prueba, restando el puntaje total del ejercicio o parte afectada.
- **La entrega de toda la prueba debe realizarse por medio de la plataforma *Canvas*, proporcionando al profesor el enlace (link) del repositorio GitHub donde se alojan los notebooks, el informe LaTeX y los archivos auxiliares.** No se aceptan entregas por correo electrónico ni en otros formatos.

1. Primera Parte: Descripción detallada de los ejercicios a desarrollar (60 %)

Los siguientes ejercicios forman la primera parte de la prueba. Cada inciso pide implementación, análisis y documentación de resultados. Sea claro en las suposiciones, entregue código comentado en Jupyter Notebook y agregue breves conclusiones.

1) Comparar datasets estructurados y no estructurados (10 puntos).

- Procesar al menos un dataset tabular (CSV) y uno no tabular (por ejemplo JSON o texto libre).
- Mostrar el flujo de carga, limpieza y una breve exploración con estadísticas básicas: promedios, desviaciones, conteos y visualizaciones simples, proporcionando ejemplos para cada tipo.
- Explicar las diferencias en complejidad de análisis, preprocesamiento y utilidad en ciencia de datos. Incluya ejemplos concretos de transformaciones necesarias para cada tipo.

2) Pipeline de ingestión de datos grandes (10 puntos).

- Trabajar con un dataset mayor a 200 MB que se obtenga o consuma directamente desde la nube (el archivo no debe estar almacenado localmente). Puede usar fuentes públicas como UCI Machine Learning Repository (https://archive.ics.uci.edu), Kaggle (https://www.kaggle.com) u otros repositorios, como también el drive propio de su correo UTEM.
- Investigar el uso, compararando la lectura y manipulación con python de las librerías **Pandas** (https://pypi.org/project/pandas/) y **Dask** (https://pypi.org/project/dask/). Muestre lecturas por fragmentos (chunks) y operaciones típicas (filtrado, agregados) con cada herramienta.
- Explique la funcionalidad de las librerias, las ventajas y limitaciones de pandas vs dask en términos de memoria, paralelismo y facilidad de uso. Incluya tiempos y observaciones de rendimiento si es posible.

3) Comparación de Pandas y PySpark (10 puntos).

- Investigar el uso, compararando la lectura y manipulación con python de las librerías **Pandas** (https://pypi.org/project/pandas/) y **PySpark** (https://pypi.org/project/pyspark/).
- Cargar un dataset (indique la fuente) y realizar operaciones de filtrado, agrupamiento y conteo con ambas herramientas (Pandas y PySpark).
- Comparar rendimiento, escalabilidad y facilidad de uso. Incluya ejemplos de código y mediciones simples (tiempos de ejecución en su entorno).
- Completar datos faltantes usando distintos métodos: media, mediana y forward-fill. Compare resultados y costo computacional.

4) Comparar librerías de visualización (10 puntos).

- Investigar el uso, compararando la lectura y manipulación con python de las librerías **Matplotlib** (https://pypi.org/project/matplotlib/), **Seaborn** (https://pypi.org/project/seaborn/) y **Plotly** (https://pypi.org/project/plotly/).
- Realizar análisis y visualizaciones de los datos usando Matplotlib, Seaborn y Plotly.
- Para cada librería, describir ventajas, desventajas y escenarios de uso recomendados (publicación estática, exploración interactiva, dashboards, etc.).
- Incluya ejemplos: al menos una gráfica estática y una interactiva (plotly) basadas en el mismo subconjunto de datos.

5) Implementar un perceptrón desde cero (20 puntos).

- Buscar y usar un dataset linealmente separable (por ejemplo, clases del dataset Iris reducidas a dos características y dos clases).

- Visualizar los datos en un scatter plot para confirmar que son linealmente separables, trazando la línea de decisión estimativa que separa los datos. Puede ser dibujada con un editor gráfico o programáticamente.
- Preprocesar los datos: normalización, división en conjuntos de entrenamiento y prueba.
- Programar el entrenamiento completo de un perceptrón sin usar librerías de machine learning (solo NumPy permitido).
- Mostrar: inicialización de pesos, forward pass, regla de actualización (Perceptron learning rule), proceso de entrenamiento (épocas) y evaluación final. Entregue gráficos de la frontera de decisión y la evolución del error o número de errores por época.

2. Segunda Parte: Descripción del Anteproyecto del "Proyecto Integrador" (40 %)

El estudiante debe proponer el anteproyecto de su Proyecto Semestral – Integrador. Las ideas de proyecto deben ser originales, viables y **alineadas con los temas del curso**. El documento debe ser claro, conciso y demostrar viabilidad técnica y académica. A continuación se indica la estructura mínima exigida; complete cada ítem con la mayor precisión posible.

(i) Estructura mínima exigida: describa las secciones que contendrá el anteproyecto (resumen, introducción, objetivos, metodología, cronograma, referencias, etc.).

(ii) Título tentativo del proyecto.

(iii) Justificación: relevancia del problema a resolver y relación con la ciencia de datos. Explique el impacto esperado y el estado actual del problema en el dominio seleccionado.

(iv) Objetivos generales y específicos: formule un objetivo general y al menos 3 objetivos específicos medibles.

(v) Metodología inicial propuesta: fuentes de datos (incluya ejemplos y accesibilidad), técnicas (limpieza, modelado, evaluación) y herramientas a utilizar (p. ej., pandas, dask, PySpark, scikit-learn, TensorFlow, PyTorch, etc.).

(vi) Estado del arte breve: incluya al menos 3 referencias académicas o técnicas recientes (año, autores, título, enlace si corresponde) y un análisis crítico que explique cómo su propuesta se diferencia o mejora trabajos existentes.

(vii) Cronograma tentativo: presente un cronograma por semanas o entregables (tabla o lista) que cubra el semestre y muestre hitos intermedios.

Formato de entrega: Documento en LaTeX (máx. 10 páginas) + PDF compilado. Debe subirse al mismo repositorio GitHub del curso junto con los archivos fuente (imágenes/datos auxiliares si aplica). Incluya además un breve README indicando cómo compilar el PDF.

3. Etapas y fechas de entrega

a) Fecha de publicación de la prueba: 02 de octubre de 2025. El plazo de desarrollo de la presente prueba es de una semana a partir de la fecha de publicación.

b) Fecha de entrega y presentación: jueves 09 de octubre de 2025, en el horario de clases habituales. La entrega consiste en proporcionar el enlace del repositorio GitHub al profesor (ver instructivo de entrega en la sección de Instrucciones Generales).

c) Asistencia y evaluación de presentación:

- La entrega, exposición y defensa de la prueba será realizada en clases el día jueves 09 de octubre de 2025.
- La no asistencia de algún integrante del grupo en la sesión de exposición será calificada con nota mínima 1.0 para ese integrante y su nota será promediada con la nota grupal (es decir, el integrante ausente recibirá 1.0 y afectará la nota del grupo).
- La no entrega o la no presentación total o parcial de la prueba será calificada con 0 puntos para la parte o el ejercicio no entregado o no presentado.

4. Entrega y presentación final

La entrega y presentación final de la prueba debe cumplir con los siguientes requisitos:

- **Entrega digital:** Toda la prueba (notebooks, informe LaTeX, archivos auxiliares y README) debe subirse al repositorio GitHub del grupo. El enlace al repositorio debe ser entregado exclusivamente a través de la plataforma *Canvas* en la tarea correspondiente, antes del plazo indicado en la sección de fechas.
- **Formato de los archivos:** El informe debe entregarse en formato PDF compilado y acompañado de los archivos fuente (.tex, imágenes, datos auxiliares si aplica). Los notebooks deben estar en formato .ipynb y ejecutados completamente, mostrando todos los resultados y visualizaciones.
- **README obligatorio:** El repositorio debe incluir un archivo README.md que indique los integrantes del grupo, explique la colaboración individual y describa brevemente el flujo de trabajo seguido (ramas, commits, integración, etc.), así como instrucciones para compilar el informe y ejecutar los notebooks.
- **Presentación oral:** La defensa y exposición del trabajo se realizará en la fecha y horario indicados, de manera presencial. Todos los integrantes deben participar y estar preparados para responder preguntas sobre cualquier parte del trabajo.
- **Criterios de evaluación:** La calificación considerará tanto la calidad técnica y presentación de los entregables como la claridad, justificación y defensa oral. El incumplimiento de los requisitos de entrega, formato o presentación implica penalización según la rúbrica.
- **Penalizaciones:** La no entrega, entrega fuera de plazo, presentación incompleta o la ausencia de algún integrante en la defensa serán penalizadas según lo indicado en la sección de fechas y reglas.

Nota: Revise cuidadosamente que todos los archivos estén presentes y correctamente documentados antes de entregar. No se aceptarán entregas por correo electrónico ni en otros formatos.

5. Rubrica y ponderaciones

Descripción	**Excelente (100 %)**	**Bueno (75-99 %)**	**Suficiente (60-74 %)**	**Insuficiente (0-59 %)**
Primera Parte				
Ejercicio 1 — Datos estructurados/no estructurados (10 pts)	Cumple todos los requisitos, análisis profundo, código claro y bien comentado, visualizaciones y reflexión crítica.	Cumple la mayoría, detalles menores, análisis suficiente, código mayormente claro.	Parcialmente cumple, resultados incompletos, análisis superficial, código poco claro.	No cumple requisitos, resultados incorrectos o ausentes, análisis pobre, código confuso.
Ejercicio 2 — Pandas vs Dask (10 pts)	Comparación completa, uso correcto de ambas librerías, análisis de rendimiento y justificación clara.	Comparación suficiente, uso aceptable, análisis parcial, justificación aceptable.	Comparación superficial, uso incompleto, análisis débil, justificación débil.	No hay comparación real, uso incorrecto o ausente, sin análisis ni justificación.
Ejercicio 3 — Pandas vs PySpark (10 pts)	Operaciones y comparación completas, análisis de escalabilidad y rendimiento, código claro, reflexión crítica.	Operaciones y comparación suficientes, análisis parcial, código mayormente claro.	Operaciones o comparación incompletas, análisis superficial, código poco claro.	No cumple requisitos, sin comparación ni análisis, código confuso o ausente.
Ejercicio 4 — Visualización comparada (10 pts)	Uso adecuado de las 3 librerías, visualizaciones claras y justificadas, análisis de ventajas/desventajas.	Uso suficiente, visualizaciones aceptables, análisis parcial.	Uso limitado, visualizaciones básicas, análisis débil.	No hay visualizaciones relevantes, análisis ausente o incorrecto.
Ejercicio 5 — Perceptrón desde cero (20 pts)	Implementación completa, explicación clara, visualización de frontera, reflexión crítica.	Implementación suficiente, explicación aceptable, visualización parcial.	Implementación incompleta, explicación superficial, visualización básica.	No hay implementación funcional, explicación o visualización ausente.

Segunda Parte				
Documento en LaTeX (Anteproyecto, 40 %)	Cumple estructura, justificación sólida, objetivos claros, metodología y cronograma detallados, estado del arte crítico, presentación impecable.	Cumple la mayoría, justificación y objetivos aceptables, metodología suficiente, presentación adecuada.	Parcialmente cumple, justificación débil, objetivos poco claros, metodología superficial, presentación mejorable.	No cumple estructura, justificación, objetivos, metodología ausentes o irrelevantes, presentación deficiente.

Nota: Cada ejercicio y el anteproyecto serán evaluados según estos niveles, considerando tanto la calidad técnica como la claridad, justificación y presentación.