



---

## **Proyecto del Curso: Exploración de Herramientas Paralelas y Distribuidas para el Análisis de Datos en Python**

---

**PROFESOR**

Christian Asch

2025-10-09

# Índice

<b>1 Proyecto del Curso: Exploración de Herramientas Paralelas y Distribuidas para el Análisis de Datos en Python</b>	<b>1</b>
1.1 Objetivo . . . . .	1
1.2 Descripción . . . . .	1
1.3 Evaluación . . . . .	3
1.4 Notas adicionales . . . . .	4
1.4.1 Posibles fuentes de datos >1GB . . . . .	4
1.4.2 Plantillas para informes IEEE . . . . .	5

## 1. Proyecto del Curso: Exploración de Herramientas Paralelas y Distribuidas para el Análisis de Datos en Python

### 1.1. Objetivo

El objetivo de este proyecto es que el estudiante explore, analice y demuestre el funcionamiento de una o más bibliotecas modernas de Python para programación paralela y distribuida orientadas al análisis de datos. A través de un estudio práctico y una investigación documental, se busca que el estudiante desarrolle competencias tanto teóricas como aplicadas, evaluando el impacto y potencial de dichas herramientas en el contexto de tareas de procesamiento y análisis de datos.

### 1.2. Descripción

Cada estudiante (o grupo pequeño de máximo 4 personas) deberá seleccionar una de las siguientes bibliotecas, según su interés y afinidad con el tema:

- **Memoria compartida:** Joblib, Polars
- **Memoria distribuida y ejecución diferida:** Dask, Ray, Modin, Vaex
- **Aceleración con GPU y computación científica:** CuPy, RAPIDS (cuDF, cuML, Dask-cuDF), JAX, PyTorch, PyArrow

El proyecto consta de dos entregables finales obligatorios:

#### 1. Informe técnico en formato IEEE:

Cada estudiante o grupo de estudiantes deberá entregar un informe técnico siguiendo el formato IEEE para artículos científicos. Este debe incluir:

- **Introducción:** Contexto del problema y motivación para explorar la biblioteca seleccionada.
- **Trabajo relacionado:** Breve revisión de trabajos que utilicen esta biblioteca y sus respectivos campos de aplicación.
- **Marco teórico:** Explicación técnica del funcionamiento interno de la biblioteca (modelo de ejecución, paralelismo, etc.).
- **Metodología:** Descripción detallada del experimento realizado, incluyendo diseño, datos utilizados y configuración del entorno.
- **Resultados:** Análisis de desempeño, visualización de datos, comparaciones con herramientas tradicionales (e.g., pandas, numpy).
- **Conclusiones:** Reflexiones finales, ventajas/desventajas de la herramienta, posibles aplicaciones futuras.

## 2. Repositorio de código en GitLab:

Cada grupo deberá mantener un repositorio público o privado (compartido con el docente) en GitLab que contenga:

- Código fuente debidamente organizado y documentado.
- Scripts o notebooks Jupyter usados en la experimentación.
- Archivo README.md con instrucciones claras para replicar los experimentos.
- Resultados relevantes (figuras, tablas, archivos de salida, etc.).

Este proyecto permitirá a los estudiantes aplicar los conceptos aprendidos en clase, desarrollar pensamiento crítico sobre herramientas modernas en Python, y fortalecer habilidades de escritura científica y documentación de código en un contexto profesional.

### 1.3. Evaluación

#### Rúbrica de Evaluación - Entrega 1: Fundamentos y Diseño del Proyecto

Criterio	Descripción	Puntaje
Claridad del problema	El problema está bien planteado, con contexto y motivación clara.	0–5
Objetivos específicos	Se definen objetivos claros, alcanzables y alineados con el problema.	0–5
Justificación técnica	Se explica por qué las herramientas o librerías fueron elegidas.	0–5
Plan de trabajo	El cronograma es realista y está alineado con los objetivos.	0–5
Documentación y presentación	El documento es ordenado, bien escrito y cumple con el formato requerido.	0–5
<b>Total</b>		<b>/25</b>

#### Rúbrica de Evaluación - Entrega 2: Implementación Inicial y Validación Parcial

Criterio	Descripción	Puntaje
Avance funcional	Se demuestra código que realiza parte del objetivo general.	0–5
Organización del código	Código modular, legible, con comentarios y buenas prácticas.	0–5
Validación inicial	Resultados preliminares o pruebas que justifican el progreso.	0–5
Gestión de problemas	Se identifican obstáculos y se documenta cómo se abordaron.	0–5
Documentación y presentación	Informe conciso, claro, con instrucciones mínimas para ejecutar el código.	0–5
<b>Total</b>		<b>/25</b>

### Rúbrica de Evaluación - Entrega 3: Proyecto Final Completo

Criterio	Descripción	Puntaje
Funcionalidad del proyecto	El proyecto cumple los objetivos propuestos, produce resultados correctos.	0–5
Eficiencia o escalabilidad	Se analiza el rendimiento o se justifica el diseño implementado.	0–5
Análisis de resultados	Resultados interpretados con criterios técnicos, comparación si aplica.	0–5
Código limpio y reproducible	Código bien estructurado, con instrucciones claras para ejecutarlo.	0–5
Informe final	Documento completo, bien argumentado, con conclusiones y lecciones.	0–5
<b>Total</b>		<b>/25</b>

### Ponderación Sugerida

Entrega	Descripción	Peso
Entrega 1	Fundamentos y diseño del proyecto	20 %
Entrega 2	Implementación inicial y validación parcial	30 %
Entrega 3	Proyecto final completo	40 %
Entrega 4	Presentación	10 %

## 1.4. Notas adicionales

El proyecto está diseñado para ser flexible y adaptable al nivel de experiencia del estudiante. Se fomentará la creatividad, el aprendizaje autodirigido y el pensamiento crítico sobre las ventajas y desventajas de las distintas herramientas paralelas y distribuidas en el ecosistema Python.

### 1.4.1. Posibles fuentes de datos >1GB

- Kaggle
- HuggingFace
- ERA5 atmospheric variable dataset
- CHRS precipitation dataset

#### 1.4.2. Plantillas para informes IEEE

- L<sup>A</sup>T<sub>E</sub>X
- Word