

# EXAMEN DEL SEGUNDO PARCIAL

Programación de Alto Rendimiento

8 de agosto de 2025

## Instrucciones:

A continuación se le presentan una serie de preguntas, léalas cuidadosamente, resuelvalos en forma clara y ordenada. Todas las respuestas deben de presentarse en formato .pdf y escritos en LaTeX; todo el código que presente debe de estar debidamente formateado usando el ambiente `minted`. No incluya archivos de código completos, puede solo presentar el código relevante a la solución a cada problema pero debe estar preparado para presentar el código completo y funcionando en caso de solicitarlo. Puede usar el código de este examen como plantilla.

## 1. Imagen Docker

Escriba un `Dockerfile` que genere una imagen de un contenedor docker con las siguientes Instrucciones (5 pts/cu):

1. Instale la última versión de la distribución de linux `alpine`.
2. Instale los paquetes necesarios para compilar código C y descargar repositorios de `github.com` usando en manejador de paquetes de `alpine`
3. Clone el repositorio de la clase: `https://github.com/lab156/PAD-2025`.
4. Compile los programas `generate.write.c` y `lineal.args.c` ambos ubicados en la carpeta `PAD-2025/gradiente/regresion`.
5. Establecer en el `Dockerfile` que al correr la imagen solo con el comando `run` (es decir, **sin** modo interactivo `-it`) debe de crear una carpeta llamada `data`, ejecutar `generate.write` y guardar los resultados en `data` y finalmente ejecutar este archivo con el ejecutable `lineal.args`.

## 2. Python ctypes

En este problema, es necesario utilizar el “virtual environment” que creamos en clase y disponible en el repositorio de la clase en la carpeta `ProyectoUno`.

Primero, agregue la librería de Python `scikit-learn`. Esta librería contiene la base de datos llamada `iris`. El código para leer esta base de datos está incluido en el archivo `read_iris.py`, específicamente en la función `get_iris_data` :

```
def get_iris_data():
    iris = load_iris()
    # target is the dependent variable
    y = iris.target
    # data has 4 columns, we will only use the first one
    x = [row[0] for row in iris.data]
    return x, y
```

Usando la librería `ctypes` de Python, enviar los datos a una función en C que encuentre la recta de regresión de los datos en `x` y `y`. Finalmente regrese a la función en Python los valores de  $m$  y  $b$ . Muestre los resultados imprimiéndolos en la pantalla **desde Python**. (40pts).

### 3. LAPACK

Escriba un programa en C que genere dos matrices aleatorias con componentes distribuidos uniformemente entre 0 y 100 de tamaño  $300 \times 200$  con componentes de tipo `float`. Calcule el producto de la traspuesta de la primera por la segunda de forma que quede una matrix simétrica de  $200 \times 200$ . Finalmente encuentre e imprima el número de condicionamiento del producto aprovechando la simetría de la matrix, e.j. `sircom`. Use solo funciones de la librería LAPACK (35pts).