



# Programación Concurrente y Distribuida (CC65)

*Tarea Académica3 2024-1*

Secciones: Todas  
Profesores: Carlos Alberto Jara García

## Instrucciones

- La tarea se desarrolla de manera grupal (grupos de 3 integrantes).
- Desarrollar uno de los algoritmos según indicación de su profesor.
- Se usará software para detección de plagio.

## Contexto

La programación concurrente se refiere a la capacidad de un sistema para realizar múltiples tareas simultáneamente, lo que es esencial en el contexto actual de desarrollo de software, donde la demanda de sistemas eficientes y escalables es alta. La combinación de programación concurrente con machine learning ofrece varios beneficios en la actualidad.

Estos beneficios se centran principalmente en el aprovechamiento eficiente del uso de goroutines para la mejora del rendimiento de las aplicaciones, así como, estructuras para sincronización entre goroutines usando canales, especialmente cuando se trabaja con grandes conjuntos de datos.

## Descripción del Trabajo

El trabajo parcial consiste en desarrollar una aplicación que usando GO implemente uno de los algoritmos siguientes sin usar librerías externas, para resolver un problema específico (Caso de Uso) que deberá plantear su enunciado, elaborando su dataset para su solución de la manera más eficiente posible, haciendo uso de mecanismos de paralelización y sincronización con goroutines y canales:

- **Regresión Lineal**
- **K-Means**
- **Clasificación con Árboles de Decisión**

## Rúbrica de calificación:

- ✓ **(3 puntos)** Preparar el enunciado del caso de uso del trabajo y el procedimiento para preparar el dataset de 1 millón de elementos, explicando su estructura de datos.
- ✓ **(2 puntos)** Publicar el código fuente y el dataset como RAW format URL, en Github.
- ✓ **(2 puntos)** El programa debe leer el dataset en su código fuente desde RAW format URL.
- ✓ **(6 puntos)** Implementar el algoritmo seleccionado de manera concurrente usando el lenguaje Go y sincronización con canales, se calificará con 0 puntos si hace uso de librerías de terceros).
- ✓ **(5 puntos)** Realizar la prueba del algoritmo con un número de ejecuciones, por ejemplo 1000 veces para obtener los resultados óptimos que den solución al problema planteado en el caso de uso.
- ✓ **(2 puntos)** Presentar la documentación completa (es excluyente para la calificación de los puntos anteriores).

**Documentación:**

1.- Presentar un informe en documento word conteniendo:

- a) Carátula
- b) Índice
- c) Introducción
- d) Planteamiento del caso de uso (adicionalmente a la explicación del caso de uso, explica la preparación del dataset)
- e) Explicación del algoritmo concurrente usado y la justificación del uso de goroutines y canales para la solución del problema del caso de uso planteado.
- f) Explicación de las pruebas realizadas con sus resultados y pegar las imágenes de evidencia claramente entendible.
- g) Conclusiones del trabajo
- h) Bibliografía de ayuda usando APA.
- i) Anexos:
  - a. Enlace de github donde subió su código fuente y se pueda descargar (Se bajará 5 puntos menos si se evidencia que hubo actualizaciones en el repositorio luego de la fecha de presentación)
  - b. Evidencia de publicación del dataset como RAW format URL desde github (Se bajará 5 puntos menos si se evidencia que hubo actualizaciones en el repositorio luego de la fecha de presentación)

**Presentación:**

Colocar como nombre del documento word su código de alumno (ejemplo 201616054.docx) finalmente subir el archivo al aula virtual.

**OJO: Cada integrante del grupo sube su archivo de forma individual.**

Santiago de Surco, mayo 2024