

## Introducción a las Computadoras Grado de Inteligencia Artificial

### DESCRIPCIÓN DEL PROYECTO FINAL

#### Objetivo

El proyecto final se utiliza como medio para reforzar la adquisición de competencias formativas, en particular para que los estudiantes demuestren los conocimientos adquiridos de tal forma que sean capaces de aplicarlos en aplicaciones reales y hayan desarrollado aquellas habilidades de aprendizaje necesarias para facilitar estudios posteriores con un alto grado de autonomía. En este sentido, el proyecto tiene como objetivo fundamental consolidar y potenciar la capacidad de los alumnos en cuanto a:

- comprender el funcionamiento interno de un computador y de sus bloques funcionales;
- comprender la interrelación entre el software del sistema operativo y el hardware sobre el que se ejecuta;
- tener la capacidad para desarrollar códigos que aprovechen de forma óptima los recursos hardware disponibles en el computador;
- ser capaz de desarrollar códigos que se ejecuten en sistemas paralelos de memoria compartida y distribuida, así como en aceleradores hardware;
- comprender los mecanismos para analizar el rendimiento y optimizar la eficiencia de códigos paralelos.

El proyecto consiste en la desarrollo de un programa concurrente y el análisis de su ejecución en CPU. Queremos mapear el ciclo de vida de los hilos del programa concurrente registrando los tiempos cuando cambian sus estados. Para ello, se deben seguir los siguientes pasos para investigar el comportamiento de los hilos:

- escribe un programa principal y una clase que extiende Thread. En el programa principal, comienza la ejecución de una lista de hilos que van a realizar una operación matemática (que se explica en el próximo paso).
- el hilo debe ejecutar una operación de cálculo intensivo, en particular, la prueba de remojo PDP-11 (<https://en.wikipedia.org/wiki/PDP-11>) que se basa en la aplicación continua de funciones trascendentes. Para esto, podemos aplicar continuamente la tangente y su inverso, seguidos por la raíz cúbica. Una implementación puede ser la siguiente:

```

for(int i=0; i<1000000; ++i) {
    double d=tan(atan(
        tan(atan(
            tan(atan(
                tan(atan(123456789.123456789))
            ))
        ))
    ));
    cbrt(d);
}

```

- Queremos entender lo que hace cada hilo durante su ciclo de vida. Para ello, debes insertar diagnósticos (utilizando print) en tu código para mapear el ciclo de vida de cada hilo. Debes distinguir entre el momento en que se ejecuta y el momento en que termina. ¿Puedes distinguir entre la creación del hilo, la ejecución y la terminación final? ¿Qué problemas encuentras cuando intentas determinar las diferentes fases?
- A continuación, deberéis analizar cómo varía el tiempo de ejecución respecto a número de hilos: estudia el comportamiento de tu código a medida que aumentas la cantidad de subprocesos implementados; es decir, comienza con un hilo y aumenta hasta un gran número de hilos (max. 200), imprimiendo el tiempo total de ejecución. Debéis desarrollar un script bash que guarde los resultados de la medición en un archivo de texto simple.
- Con el archivo del paso anterior debéis generar gráficas del tiempo de ejecución en función del número de subprocesos (es decir, el eje x es el número de subprocesos, mientras que el eje y es el tiempo de ejecución). Podéis usar matplotlib o seaborn de Python para generar dichas gráficas. A partir de estas gráficas, ¿qué conclusiones podéis extraer de este análisis?

La realización del proyecto se llevará a cabo en **grupos de 3 alumnos**. Se recomienda que los alumnos asistan a clases con el fin de poder discutir en detalle las distintas tareas, así como la preparación del informe final.

## Evaluación para asistentes y no asistentes

**Todos los grupos** deberán realizar, **obligatoriamente, la entrega del informe y del código en la misma fecha (27 de Noviembre hasta las 23:59 CET)**, prefijada y publicada en Moovi. Si existen dudas sobre los resultados entregados, se solicitará la defensa del proyecto, por parte de todos los miembros del grupo (obligatorio), en horario de tutorías, incluyendo la ejecución del código entregado en una máquina local.