

## **Resumen ejecutivo - Práctica 1:**

### **Desarrollo de código para el procesador ARM**

Para este trabajo se van a implementar diversas versiones del juego conecta 4, el cual consiste en hacer 4 (o más) en raya en un tablero de 7x6, siendo que solo se pueden introducir fichas por turnos y se colocará siempre en la fila con menor valor disponible en la columna seleccionada.

#### **Objetivo:**

Se busca optimizar el rendimiento del juego acelerando las funciones computacionalmente más costosas. Para así familiarizarnos con el código en ARM, y aprender a analizar y medir los resultados obtenidos.

En adición se busca aprender a utilizar código en C y en ARM combinado para todo ello se realizarán las siguientes funciones:

- conecta4\_buscar\_alineamiento\_arm
- conecta4\_hay\_linea\_arm\_c
- conecta4\_hay\_linea\_arm\_arm

Las cuales son equivalentes a conecta4\_buscar\_alineamiento\_c, conecta4\_hay\_linea\_c\_c y la unificación de estas dos funciones respectivamente.

Todo ello se realizará ateniéndonos al estándar ATPCS. El cual se puede consultar en el siguiente [link](#).

#### **Metodología:**

Para ello se ha utilizado el simulador keil, en concreto se trabajará sobre el microprocesador LPC2150, basado en el procesado 16/32-bit ARM7TDMI-S En el cual se ha estudiado el material proporcionado y se han afianzado los conocimientos sobre el encapsulamiento del código y el funcionamiento de la pila.

Tras lo cual se han implementado las versiones anteriormente mencionadas, por último se ha estudiado el coste temporal de la ejecución de cada una de las versiones, al igual que el tiempo de las funciones en C con diversos niveles de optimización.

#### **Conclusiones:**

De todo ello se ha extraído la conclusión de que el código en ensamblador directo tiende a ser más óptimo que el que se ha realizado a alto nivel siempre que el compilador no aplique ningún nivel de optimización posteriormente al código en alto nivel.

De igual forma se ha concluido que los diversos niveles de optimización eliminan y reorganizan el código llegando a obtener resultados sorprendentes.

El resto de la documentación se encuentra en la Memoria\_p1 adjunta en el zip.