

## **“Tutoría Académica 2: Algunos conceptos útiles para construir programas en MIPS”**

### ***Método de trabajo:***

1. Lee detenidamente el guión para saber qué preguntas tienes que contestar (5 minutos).
2. Visiona los [vídeos](#) de explicación de esta sesión académica antes de acudir a tu sesión
3. Consulta la lista de [operaciones con enteros](#) más destacadas disponibles en el MIPS
4. En la sesión el profesor te hará una explicación usando la [presentación preparada](#) para esta sesión.
5. Plantea tus dudas al profesor durante el proceso. Al finalizar la sesión trata de resolver las preguntas que se plantean.

### ***Resultado del trabajo:***

**Obtendrás más información sobre cómo programar algoritmos en MIPS. Aprenderás cómo usar registros y cómo codificar las condicionales y estructuras de control más habituales. Este trabajo te ayudará en las sesiones prácticas.**

### ***Introducción***

Las personas programadoras suelen desarrollar sus algoritmos en un lenguaje de alto nivel. Para desarrollar las prácticas y los problemas, el primer paso lógico es plantear una solución en un pseudocódigo similar a un lenguaje de alto nivel para después traducirlo a lenguaje ensamblador, tal y como lo haría un compilador. En las prácticas realizadas hasta ahora hemos podido trabajar algo con las variables (bajo la directiva `.data`) y hemos realizado algunas operaciones sencillas. Al finalizar esta sesión podrás utilizar los registros de una manera correcta y además podrás utilizar las sentencias de control básicas que aparecen en todos los algoritmos.

### ***Material***

Para realizar la actividad de esta sesión debes visualizar los [videos de explicación](#) descargarte la [presentación](#) que está en el aula virtual y atender las explicaciones del profesor sobre la misma.

### ***Actividad a realizar***

Atiende a las explicaciones que el profesor te da. Al finalizar la explicación podrás realizar las siguientes actividades.

1. Programa en ensamblador para el MIPS un ejemplo de estructura condicional múltiple (switch).
2. Programa en ensamblador para el MIPS un ejemplo de estructura repetitiva do-while.
3. Trata de realizar un ejemplo en ensamblador que realice la evaluación de una expresión lógica compuesta: `if (($0 < $s1) && ($s1 == $s2)) resultado = 1;`