

---

## Laboratorio 4: Código ensamblador: ARMv4

---

Fecha de asignación: 09 de mayo 2025

Fecha de entrega: 21 de mayo 2025

Profesores: Luis Barboza Artavia  
Jeferson González Gómez

---

### 1. Introducción

El código ensamblador es un lenguaje de programación de bajo nivel que representa instrucciones específicas para un procesador. A diferencia de los lenguajes de programación de alto nivel, el código ensamblador está estrechamente vinculado a la arquitectura del procesador y utiliza mnemónicos y códigos de operación para representar las operaciones que la CPU debe realizar. Cada instrucción en código ensamblador se traduce directamente a una instrucción de máquina ejecutable por el procesador.

ARMv4 es una arquitectura de procesador desarrollada por ARM Holdings a principios de la década de 1990, conocida por su eficiencia energética y versatilidad. Introdujo características importantes como Thumb, una tecnología de instrucción de 16 bits para mejorar la densidad de código y el rendimiento en dispositivos con recursos limitados. ARMv4 fue ampliamente adoptada en una variedad de dispositivos móviles y embebidos, sentando las bases para el éxito continuo de ARM en la industria de semiconductores.

En este laboratorio el estudiante resolverá problemas mediante el uso de código ensamblador de ARMv4.

### 2. Investigación

Esta evaluación es **individual**. Para el desarrollo de este laboratorio se deben responder las siguientes preguntas.

1. Investigue los tipos de instrucción que tiene ARMv4. Nombre dos ejemplos de cada tipo y un posible uso.
2. Explique el set de registros que tiene ARM. ¿Cuál es el contenido de cada uno?
3. Explique el funcionamiento del *branch*.
4. ¿Cómo se implementa un condicional (*if/else*) en ARMv4?

5. Explique el procedimiento para transformar el código en ensamblador a binario. Investigue herramientas que realizan el proceso.
6. Explique la diferencia de *little endian* y *big endian*. ¿Cuál es su implicación cuando pasan las instrucciones a lenguaje máquina (binario)?

### 3. Ejercicios Prácticos

A continuación se presentan 3 ejercicios prácticos, los cuales debe resolver de manera completa **individualmente**. Para cada ejercicio debe presentar el código ensamblador en ARMv4, prueba del código ensamblador en herramienta correspondiente (e.g., [VisUAL](#), [CPULator](#), etc.) y el archivo en lenguaje máquina (binario) equivalente.

#### 3.1. Problema 1

Asuma que se tiene un arreglo con 10 valores y una constante definida por cada estudiante. Realice el código ensamblador (ARMv4) para el siguiente pseudocódigo:

```
int [] array = {...}
int y =

for (i = 0, 1, ..., 10)
    if array[i] >= y
        array[i] = array[i] * y
    else
        array[i] = array[i] + y
```

#### 3.2. Problema 2

Realice el código ensamblador (ARMv4) para calcular factorial de un número  $X$ . Es decisión propia cómo se manejará el número  $X$  en el código. Realice la demostración para 3 valores de  $X$  distintos.

#### 3.3. Problema 3

Suponga que un procesador tiene anexo un contador, que representa la posición en pantalla (VGA) de un *sprite*, así como un teclado que debe controlar dicha posición cuando se recibe la tecla de flecha superior (para aumentar el contador) y la flecha inferior (para disminuirlo). Asuma que el valor de la tecla se encuentra siempre disponible en la dirección de memoria 0x1000, y el

contador se encuentra mapeado en la dirección 0x2000. Los valores en hexadecimal de las teclas son:

Flecha de arriba -> 0xE048

Flecha de abajo -> 0xE050

Con base en lo anterior, realice un programa usando ARMv4 que constantemente lea el valor de la tecla y actualice el contador correspondiente, aumentándolo o disminuyéndolo dependiendo de cuál flecha se presionó. El programa NO debe cambiar el contador en caso de que la tecla presionada no sea una de las dos opciones válidas.

**Nota:** Para efectos de la prueba del código, simule el valor de la tecla, escribiendo en la dirección de memoria correspondiente de manera manual para al menos 5 iteraciones del bucle. Se debe probar las dos teclas válidas, así como al menos una tecla inválida, para asegurarse de que el sistema funciona correctamente.

## 4. Evaluación

La evaluación de este laboratorio será individual, por medio de un informe escrito y repositorio donde se encuentren los códigos fuente (ensamblador y lenguaje máquina) con la solución a los problemas planteados. En el informe escrito deberán presentarse las respuestas de la sección de investigación, así como capturas de pantalla de las pruebas (simulación en herramienta elegida) para cada uno de los ejercicios prácticos.

La entrega se debe realizar por medio del TEC-Digital en la pestaña de evaluación. No se aceptan entregas extemporáneas después de la fecha de entrega a las 11:59 pm como máximo. **Los documentos serán sometidos a control de plagios.**