

# Laboratorio 4: Arquitectura y Organización de Computadores

05 de Junio 2025

## 1 Reglas Generales

Para la siguiente tarea se debe utilizar el Simulador RARS<sup>1</sup> para crear un programa que cumpla con los requerimientos de la sección 2. Se exigirá que el formato de los códigos se presente de la forma más limpia y ordenada posible. Deberá incluir un README con la identificación de los estudiantes que desarrollaron la tarea, además de cualquier supuesto utilizado.

## 2 Criptosistema del Núcleo Central

El **Núcleo Central** de la nave **ANDRÓMEDA XII** ha sido vulnerado. Los sistemas criptográficos encargados de proteger las comunicaciones y sensores han sido saboteados, comprometiendo la integridad operativa de la misión. Su tarea consiste en recuperar el control de los subsistemas esenciales mediante el diseño y programación de algoritmos de bajo nivel en lenguaje ensamblador RISC-V.

- Un módulo de **análisis térmico predictivo**, encargado de detectar patrones crecientes de temperatura en el sistema de refrigeración.
- Un sistema de **verificación avanzada de códigos de misión**, mediante cifrado XOR y validación estructural.

Se debe desarrollar un programa independiente por cada subsistema, los cuales deberán cumplir con los requisitos funcionales y técnicos detallados en las siguientes secciones:

### 2.1 Desafío 1: Análisis Térmico Predictivo

Se ha detectado una secuencia histórica de temperaturas en un buffer circular de 100 posiciones. Tu tarea es implementar un algoritmo que:

- Calcule la **suma móvil** de cada bloque de "**k**" temperaturas consecutivas.
- Identifique si **hay una tendencia creciente sostenida** en al menos 3 bloques consecutivos.
- Reporte:
  - Los resultados de los bloques crecientes consecutivos detectados.
  - La posición inicial del primer bloque que inicia la tendencia (si existe).

---

<sup>1</sup>Disponible en <https://github.com/TheThirdOne/rars/releases>

### 2.1.1 Entradas:

- Un entero  $n$  (cantidad de temperaturas,  $5 \leq n \leq 100$ ).
- Un entero  $k$  (tamaño del bloque,  $k < n$ ).
- $n$  enteros que representan las temperaturas.

### 2.1.2 Salida:

- Resultados de los bloques crecientes consecutivos detectados.
- Mensaje "**Tendencia detectada desde índice X**" o "**Sin tendencias detectadas**".

### 2.1.3 Ejemplo:

#### Entrada:

$n = 8, k = 3$

[10, 12, 13, 15, 18, 14, 17, 20]

#### Bloques de suma:

- $10 + 12 + 13 = 35$
- $12 + 13 + 15 = 40$
- $13 + 15 + 18 = 46$
- $15 + 18 + 14 = 47$
- $18 + 14 + 17 = 49$
- $14 + 17 + 20 = 51$

#### Salida:

Bloques crecientes:  $35 \rightarrow 40 \rightarrow 46 \rightarrow 47 \rightarrow 49 \rightarrow 51$

Tendencia encontrada desde índice 0

## 2.2 Desafío 2: Códigos de Misión con Criptografía XOR y Validación Par-Impar

Cada código de misión (cadena ASCII de hasta 64 caracteres) debe validarse mediante una verificación y encriptarse:

#### Validación:

- Calcular la **cantidad de letras mayúsculas** y la **cantidad de dígitos numéricos** en el texto original (antes de encriptar).
- El código es **válido** si:
  - La cantidad de letras mayúsculas es par.
  - La cantidad de dígitos numéricos es impar.

#### Encriptación XOR:

- Aplicar una operación XOR entre cada carácter del código y un valor clave constante (por ejemplo, 0x5A).
- Mostrar el código encriptado (debe imprimirse como valores hex).

### 2.2.1 Entrada:

- Una cadena ASCII (máx. 64 caracteres).

### 2.2.2 Salida:

- Mensaje: **"Código válido"** o **"Código inválido"**.
- Cadena encriptada con valores hex (si es válido el código).

### 2.2.3 Ejemplo "Código inválido":

#### Entrada:

Código: "ALFA2BETA3"

#### Logica:

- Letras mayúsculas: A, L, F, A, B, E, T, A  $\rightarrow$  8 (par)
- Dígitos: 2, 3  $\rightarrow$  2 (par)

#### Salida:

Código inválido

### 2.2.4 Ejemplo "Código válido":

#### Entrada:

Código: "X7J9Z2K"

#### Logica:

- Letras mayúsculas: X, J, Z, K  $\rightarrow$  4 (par)
- Dígitos: 7, 9, 2  $\rightarrow$  3 (impar)

Carácter	ASCII	XOR con 0x5A	Resultado
X	88	$88 \oplus 90 = 2$	0x02
7	55	$55 \oplus 90 = 101$	0x65
J	74	$74 \oplus 90 = 16$	0x10
9	57	$57 \oplus 90 = 99$	0x63
Z	90	$90 \oplus 90 = 0$	0x00
2	50	$50 \oplus 90 = 104$	0x68
K	75	$75 \oplus 90 = 17$	0x11

Table 1: Operación XOR de caracteres con  $0x5A = 90$

#### Salida:

Código encriptado: [0x02, 0x65, 0x10, 0x63, 0x00, 0x68, 0x11]

Código válido

### 3 README

Debe contener como mínimo:

- Nombre, Rol y Paralelo de los integrantes.
- Especificación de los algoritmos y desarrollo realizado.
- Supuestos utilizados

### 4 Consideraciones

- Se deberá trabajar de a pares. Se deberá entregar en Aula a mas tardar el día 27 de junio de 2025 a las 23:59 horas. Se descontarán 5 puntos por cada hora o fracción de atraso. Las copias serán evaluadas con nota 0 en el promedio de las tareas.
- La tarea debe realizarse en RARS. Se recomienda que se familiarice rápidamente con la plataforma, y ante cualquier duda consulte con sus compañeros o directamente con los ayudantes lo antes posible. El único responsable si no acude a alguien para resolver sus dudas a tiempo es usted.
- La entrega considera dos archivos, sub1 y sub2, junto con el README. Los archivos deberán ser comprimidos y enviados juntos en un archivo .zip de nombre LAB4\_ROL1\_ROL2.
- Si no se entrega README, o si su programa no funciona, la nota es 0 hasta la corrección.
- Una vez entregadas las notas de la tarea existirá un plazo de 3 días para apelar. Transcurrido este plazo las notas no podrán ser modificadas.