



## Proyecto #3

# Matrices

---

# IC-3101 Arquitectura de Computadoras

**Tecnológico de Costa Rica, Sede Central Cartago**  
**Escuela de Computación, Ingeniería en Computación**

**I Semestre 2017**

**Prof. Ing. Esteban Arias Méndez**

Hoy en día la mayoría de los procesadores cuentan con más de 1 núcleo de procesamiento, este poder extra debe ser aprovechado para que al plantear programas y problemas por resolver en computación se pueda aprovechar la concurrencia y en especial el paralelismo.

---

# PROGRAMACIÓN

---

## Propósitos

- Poner en práctica los conocimientos aprendidos sobre procesamiento distribuido, paralelismo, tiempo compartido y concurrencia.
- Aplicar de forma correcta instrucciones de lenguajes de programación en alto nivel que hacen uso de herramientas para implementar concurrencia y paralelismo como el uso de hilos (threads) entre otros para brindar una respuesta adecuada a los problemas planteados.

El objetivo de este proyecto es el desarrollo de un software que haga uso de las facilidades de los lenguajes de alto nivel para aprovechar la concurrencia de los procesadores actuales con varios núcleos, en especial la aplicación de paralelismo mediante uso de hilos (threads) y similares.

Se trabajará sólo sobre el Sistema Operativo Linux, utilizando lenguajes de programación NASM, C, Python o Java únicamente. Debe hacer uso del git de la Escuela de Computación [git.ec.tec.ac.cr](https://git.ec.tec.ac.cr) o similar para llevar todo el desarrollo de su proyecto, archivos de diseño, documentación, fuentes, etc.

Este trabajo es para ser desarrollado por parejas de 2 personas máximo.

El eje principal del trabajo será alrededor del uso de matrices y operaciones que se pueden llevar acabo con ellas.

Las actividades a implementar son las siguientes:

1. Un programa que genere matrices de números aleatorios con rangos de valores enteros posibles de entre -100 a +100, el usuario debe poder indicar el tamaño de dicha matriz  $N \times M$ , así como si desea valores solo positivos  $>0$ , o bien, negativos y positivos mezclados. Su programa deberá generar un archivo que contenga en la primera línea el tamaño de la matrix  $N \times M$  y a partir de la segunda línea de texto sus datos de la matriz, con valores separados por espacio. El usuario debe poder indicar el nombre de dicho archivo y generar todos los que desee.

2. Desarrollar un programa que permita al usuario aplicar operaciones entre matrices. Estas operaciones a implementar son:
  - a. Suma de matrices.
  - b. Producto de un escalar por una matriz.
  - c. Producto de matrices.
  - d. Matriz Inversa
  - e. Para una matriz dada indicar si el tipo de la misma: matriz fila, matriz columna, matriz rectangular, matriz cuadrada, matriz nula, triangular superior, triangular inferior, matriz diagonal, matriz escalar, matriz identidad o unidad.
  - f. Extra: Rango de una matriz
  - g. Extra: Calcular la transpuesta de una matriz
3. Para cada operación deberá verificar los requisitos para la operación, tal como los tamaños de las matrices u otros requisitos. En caso de haber errores se deberá informar al usuario.
4. Debe estudiar cada operación y para las operaciones donde los datos pueden ser paralelizables, habilitar el uso de hilos (threads), brindando al usuario la opción de indicar en cuantas partes desea dividir el conjunto de datos, para crear igual número de hilos para trabajar de forma independiente y concurrente la operación. El usuario podría seleccionar 0 hilos, en cuyo caso su programa realizará la operación en forma secuencial tradicional.
5. Su programa deberá en todo momento para toda operación indicar la cantidad de memoria en bytes que se ha consumido, así como el tiempo efectivo de ejecución en micro o nanosegundos para cada operación.
6. Deberán escoger 1 de las operaciones dadas para ser implementada completamente en NASM y conectarlo a su programa en alto nivel mediante el llamado a su programa en NASM, para que su programa en alto nivel solamente colecte los datos, los transmita a su programa NASM y con el resultado desde NASM desplegarlos al usuario. Verifique cómo conectar su lenguaje de alto nivel seleccionado con NASM antes de arrancar el desarrollo del trabajo.
7. El usuario debe poder ingresar y/o modificar los parámetros de su programa en cada iteración.
8. Extras:
  - a. 10% - Habilitar operaciones con números en punto flotante.
  - b. 10% - Operación Rango de una matriz
  - c. 10% - Operación Transpuesta de una matriz

---

# EVALUACIÓN

---

1. Rúbrica de evaluación:

- El proyecto se calificará con los siguientes criterios:
  - i. 40% - Funcionamiento correcto de las operaciones con matrices.
  - ii. 20% - Uso correcto de threads
  - iii. 30% - Operación en NASM
  - iv. 10% - Documentación del trabajo.
  - v. 30% - varios extras.

2. El proyecto debe resolverse, implementándolo de la mejor manera.

3. Se revisará el código del proyecto, para asegurar que se cumpla con lo solicitado de forma interna y no solo el comportamiento del programa sea como el solicitado.

4. De forma global, se evaluará la presentación del trabajo según los parámetros solicitados, estrategias empleadas y la calidad, la entrega a tiempo del trabajo y la documentación completa correspondiente.

5. Sobre la documentación y presentación:

a. 2pts - El subject del correo a ser enviado debe ser:

**[Arqui] – Proyecto # 3 – Sus Nombres Completos**

b. 2pts - El correo debe contener de forma separada:

- i. los archivos de texto de los códigos fuentes que permiten la solución y funcionalidad del mismo.
- ii. archivos de diseño o diagramas
- iii. un archivo PDF con la documentación completa

No envíe archivos ejecutables o binarios.

c. La documentación en PDF con el nombre de archivo igual al subject del correo enviado. Esta documentación debe tener un apartado, que indique los pasos a seguir, para poder ejecutar el código (librerías a instalar y otros).

- i. 5pts – La documentación debe incluir una portada con los datos completos: TEC, carrera, sede, curso y código, profesor, periodo, fecha de entrega, número de proyecto, título del proyecto, nombres completos con número de carnet de la pareja de trabajo y un abstract en inglés en la misma portada.
    - ii. 5pts – La introducción del documento es una descripción breve del trabajo realizado y herramientas usadas.
    - iii. 10pts – Como desarrollo debe explicar, los procedimientos, rutinas, la lógica que utilizo para resolver el problema. indicar los ejemplos de código que ha usado como guía para el desarrollo de los mismos usando las referencias bibliográficas correspondientes. Explicar el uso y funcionamiento.
    - iv. 20pts – Análisis de resultados, explicando el trabajo implementado, la forma de realización, funcionamiento, general, ejemplos documentados, problemas presentados, estructuras de datos empleadas, algoritmos usados, etc.
    - v. 10pts – Una sección de conclusiones y/o observaciones sobre el proyecto.
    - vi. 10pts – En una sección de Apéndices incluya el código fuente documentado del proyecto, diagramas usados y su explicación. Explique la estructura del código empleada en su proyecto, módulos, diagramas, etc.
  - d. 10pts – Las líneas del código deben venir con un comentario que haga referencia a su funcionalidad
6. Se debe entregar en digital a más tardar el Viernes 16 de Junio, antes de la media noche. Debe hacerlo de forma simultánea a los correos siguientes y copiarse usted mismo y su compañero de trabajo. Cada día de atraso serán 15pts menos de la nota de la tarea:
- a. [earias@ic-itcr.ac.cr](mailto:earias@ic-itcr.ac.cr)
  - b. [arirodriguez@ic-itcr.ac.cr](mailto:arirodriguez@ic-itcr.ac.cr)
7. Cualquier consulta puede hacerla al foro, o personalmente en clase o al correo del profesor con copia al asistente a los correos anteriores.
8. Durante la revisión del proyecto deben estar presentes ambos miembros de la pareja de trabajo, la no presentación les restará 10pts a cada uno de los ausentes.