UNIVERSIDAD NACIONAL DE INGENIERIA

FACULTAD DE CIENCIAS

Escuela Profesional de Ciencias de la Computación

Algoritmos y Estructuras de Datos

3° Practica Calificada

Fecha: 30.05.2025

Codigo : 20202176G Apellidos y Nombres: Magno Muro Renzo Jesus

***No modifique la pag.1 del formato de la prueba, consigne sus respuestas a partir de la pag.3.*** *Todas las clases, interfaces que formen parte de su solución deben tener como encabezado su Código de estudiante.*

1.-Desarrolle el programa que resuelva el Caso. Proyecto del curso: <https://github.com/zcodlab/251CC232AProject>, paquete: uni.aed.queueTDA.simularagenda

Implemente un menú de opciones, la opción visualizar es obligatorio para considerar el puntaje:

a) Cargar Agenda de Trabajo (2pts)

b) Obtener el tiempo de espera promedio (2pt)

c) Obtener el tiempo de espera máximo por nivel de prioridad (2pt)

d) Eliminar actual Agenda de Trabajo (1pt)

e) Visualizar Agenda de Trabajo

**Restricciones:**

1.Definir la solución en clases debidamente desacopladas por responsabilidad (2pts.).

2.Reutilizar las estructuras de tipo TDA del proyecto, que se requieran en la solución del caso(no JCF). Analice su impacto en la solución. Fundamente su respuesta. (2pts.)

3.Documentar en el mismo Código fuente las líneas de código significativas. (1pt.)

**Caso 1** (Pag 1097-Ej 9 )

Escriba un programa de simulación de agenda de trabajo. En las buenas épocas, mucho antes de la revolución de las PC, los estudiantes de ciencias de la computación escribían programas con el uso de máquinas perforadoras de tarjetas y los ejecutaban en las computadoras centrales. Un programa se registraba sobre una pila de tarjetas perforadas, una sentencia por tarjeta perforada, Para ejecutar sus programas, los estudiantes debían entregar el programa (la pila de tarjetas) al operador de la computadora. Un programa se llamaba trabajo (Job). A los estudiantes que tomaban un curso de computación se les asignaba cierto número de unidades (un tipo de dinero virtual) y se les cargaban N unidades para ejecutar un programa. La cantidad real cargada se determina por la prioridad que los estudiantes asignaban a sus programas. Para este ejercicio, suponga que las prioridades varían de 1 a 5, con 1 como la más alta.

Escriba un programa en java que simule la agenda de trabajo. Las entradas al programa con M, que es el número de minutos a simular y N que es el número de trabajos que la computadora puede ejecutar concurrentemente. Trate a cada minuto como un evento discreto. Suponga que un solo trabajo llega cada minuto. Cuando llega un trabajo, asigne en forma aleatoria su prioridad y el número de minutos que necesita para ejecutarse. Elija cualquier entero entre 1 minuto y 10 minutos, inclusive, para el tiempo de ejecución. Un trabajo que llega se coloca en una cola de prioridad. En cada evento discreto, si el número de trabajos asignado a la computadora esta abajo del número máximo de trabajos que puede manejar la computadora, remueve trabajos de la cola de prioridad y los asigna a la computadora.

Al final de la simulación, obtenga el tiempo de espera promedio y el tiempo de espera máximo para cada nivel de prioridad. Vea el ejercicio 8 acerca de como manejar la situación en la que la cola no este vacía al final de los minutos simulados.

**Procedimiento de Entrega de la Prueba:**

Presentar el presente formato, consignando su Código, apellidos y nombre, capturas de pantalla que demuestre la resolución de cada uno de los enunciados. El presente formato y su código fuente, publíquelo en su cuenta Github (modo privado) agregando como colaborador del proyecto a la cuenta del docente del curso: **zcodlab.** Consigne la URL de su repositorio github como respuesta a la tarea aperturada en el aula virtual UNI.

De tener inconvenientes con Github, puede comprimir el proyecto y formato Word y enviarlo al correo de la profesora: zmamanir@uni.edu.pe

Nombre del Proyecto en Github: **251CC232A<CODIGOESTUDIANTE>PC3**

**Asegúrese que el nombre de su cuenta Github lo identifique claramente: ApellidosNombres**

*\*Si se detecta 2 o más trabajos iguales se anula a todos los implicados*

*En las imágenes se muestran el código, no tuve tiempo para compilar, me parece que tenia un error con el IDE.*

   