

Primer trimestre – TEMA 1

A 10x10 grid of 100 small squares. Each square contains a black 'X' mark, which is formed by two diagonal lines intersecting at the center of the square. The 'X' marks are uniformly distributed across the entire grid.

Licencia



Reconocimiento – NoComercial – CompartirIgual (by-nc-sa): No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original.

1. Evaluación

1.1. Validación del código

Cuando acabes el examen debes acceder a un segundo enlace proporcionado por el profesor para realizar la validación del examen, que constará de varias preguntas y posibles cambios sobre tu código.

1.2. Infracciones que conllevan el suspenso del examen

Posibles infracciones:

- Se ha copiado en el examen.
- El alumno no es capaz de contestar a preguntas relacionadas con su examen.
- No se realiza la validación del examen.
- El uso de inteligencia artificial para la resolución.
- El programa no lleva el nombre del alumno/a:
 - Si me llamo Miguel Oliver, mi entrega se llamará: Examen_miguel_oliver.py

1.3. Puntuación

Si tu programa no se ejecuta correctamente, la nota máxima será un 6. Soluciones válidas pero que no sean eficientes o las descritas en el enunciado, restarán hasta 5 puntos del examen.

Se valorará la programación modular, la reutilización de funciones, etc.

Como mínimo se deberá imprimir por pantalla la misma información que se pide en el enunciado.

Si el código no incluye las funcionalidades que se piden en el enunciado, la nota máxima será un 6.

Será obligatorio el control de excepciones, si no se implementa correctamente la nota máxima será de 6.

2. Ejercicio 1

Implementa un sistema multihilo que simule la operación de una fábrica de cerveza. Este sistema debe gestionar la producción, almacenamiento y distribución de diferentes tipos de cerveza utilizando tres embotelladoras, un almacén y dos camiones de reparto. Además, deberás diseñar e implementar un **algoritmo personalizado** que fusione aspectos de FIFO, Shortest Job First (SJF) y prioridades para organizar la cola de producción de las embotelladoras. Se implementarán hilos personalizados con clases derivadas de Thread. Además, Se deben utilizar las mínimas variables globales y utilizar los argumentos.

Especificaciones

1. Simulación

- Todas las tareas se simularán con esperas.
- Para analizar el sistema, un día de producción equivale a 8 segundos, la semana tendrá 5 días, por lo tanto, la semana serán 40 segundos.

2. Embotelladoras

- **3 embotelladores**, cada una representada por un hilo personalizado con nombres Embotelladora1, Embotelladora2 y Embotelladora3.
- Cada embotelladora puede procesar 6 tipos de cerveza en diferentes envases.
- **Tiempos de embotellado por 1 palet** (segundos):

```
tiempos_produccion = [0.35, 0.53, 0.48, 0.65, 0.8, 1] # 1, 2, 3, 4, 5 ,6
```

3. Almacén

- **Capacidad máxima de stock** por producto en palets:

```
capacidad_maxima = [120, 90, 100, 80, 70, 60] # 1, 2, 3, 4, 5 ,6
```

- **Stock inicial** de cada producto en palets:

```
stock_inicial = [80, 60, 70, 60, 50, 40] # 1, 2, 3, 4, 5 ,6
```

- Producción activada cuando el stock cae por debajo del nivel necesario para cubrir una semana de pedidos, 5 días.
- **Cálculo de prioridad de producción:**
 - **Prioridad 1:** Menos de 3 días de stock.
 - **Prioridad 2:** Entre 3 y 5 días de stock (1 semana).
- El transporte de productos desde la fábrica al almacén será instantáneo (simulado como cinta).

4. Camiones de reparto

- Disponemos de 2 camiones, representados por hilos personalizados con nombres Camion1 y Camion2.
- Cada camión realiza 1 viaje al día, suponemos que necesita todo el día para realizar la entrega (8 segundos).
- Cada camión recibe órdenes de compra semanales que indican la cantidad necesaria para cada tipo de cerveza.
- Capacidad de los camiones **diaria**:

```
camiones_max = [20, 20] # Camion1, Camion2
```

- La demanda **semanal** de cada tipo de cerveza se calcula como:

```
demanda_base = [50, 35, 40, 30, 25, 20] # Tipos 1, 2, 3, 4, 5, 6
```

- Sobre esta demanda tendremos un **coeficiente variable** que dependerá de la estacionalidad, se calculará cada **dos semanas** de forma **aleatoria** entre **0,5 y 1,5**. Este coeficiente se **multiplicará** por la demanda_base y tendremos la demanda_semanal, que estará vigente durante **dos semanas**.
- **Pedidos:** La demanda_semanal para cada tipo de cerveza se dividirá en 5 pedidos (1 pedido diario para cada tipo de cerveza).
- Los camiones operan en base a una planificación **First Fit Decreasing (FFD)**:
 - Los pedidos se ordenan por tamaño total (**mayor a menor**) y se creará una cola.
 - Cada camión se asigna al pedido **más grande** que pueda cargar sin superar su capacidad.
 - Cada día se añadirán los nuevos pedidos creados a la cola y se ordenará.
- **(EXTRA 1 punto) Restricción de carga:** un camión solo puede llevar **2 tipos de cerveza** en un determinado viaje.

5. Algoritmo personalizado

- Diseña un algoritmo que combine:
 - **FIFO o FCFS.**
 - **Las Prioridades** de producción citadas anteriormente.
- La cantidad de cada producto en cada una de las solicitudes a la fábrica será:
 - Un porcentaje que puedes elegir de la demanda_semanal de ese tipo de cerveza, sin sobrepasar la capacidad máxima del almacén.
- Explica el funcionamiento del algoritmo que has diseñado.

Información a Mostrar

Producción en las Embotelladoras:

- Finalización de producción: Ejemplo: Embotelladora1 produjo 30 palets de Tipo5.

Almacén:

- Stock antes y después de cargar el camión. Ejemplo: Almacén envió 20 palets de Tipo4 con el Camion2. Stock restante: 50.
- Activación de producción: Ejemplo: Pedido de 20 palets de Tipo1 (Prioridad 1).

Camiones:

- Órdenes completadas: Ejemplo: Camion1 completó entrega de 100 palets de Tipo2.

Puntuación

1. Programación modular y estructura básica del programa con la herencia y la creación de los hilos. **(2 puntos)**
2. Control de excepciones. **(1 punto)**
3. Información por pantalla. **(1 punto)**
4. Control de concurrencia. **(1 punto)**
5. Implementación del algoritmo personalizado y justificación. **(2 puntos)**
6. Gestión de las ordenes y las entregas de los camiones. Qué problema presenta la planificación **First Fit Decreasing?** **(2 puntos)**
7. Responder a las siguientes preguntas: **(1 punto)**
 1. Evalúa las ventajas y desventajas de tu algoritmo.
 2. ¿Cómo afecta el porcentaje que has decidido sobre la demanda semanal para las peticiones de producción? Analiza su impacto en el rendimiento del sistema.
8. Restricción de carga en los camiones. **(1 punto EXTRA)**