Trabajo Práctico N°2

**1.-** Sean P1 y P2 dos procesos cuya secuencia de pedido y liberación de los recursos A y B es la que se muestra a continuación:

**P1 P2**

. . . . . .

Solicita A Solicita B

. . . . . .

Solicita B Libera B

. . . . . .

Libera A Solicita A

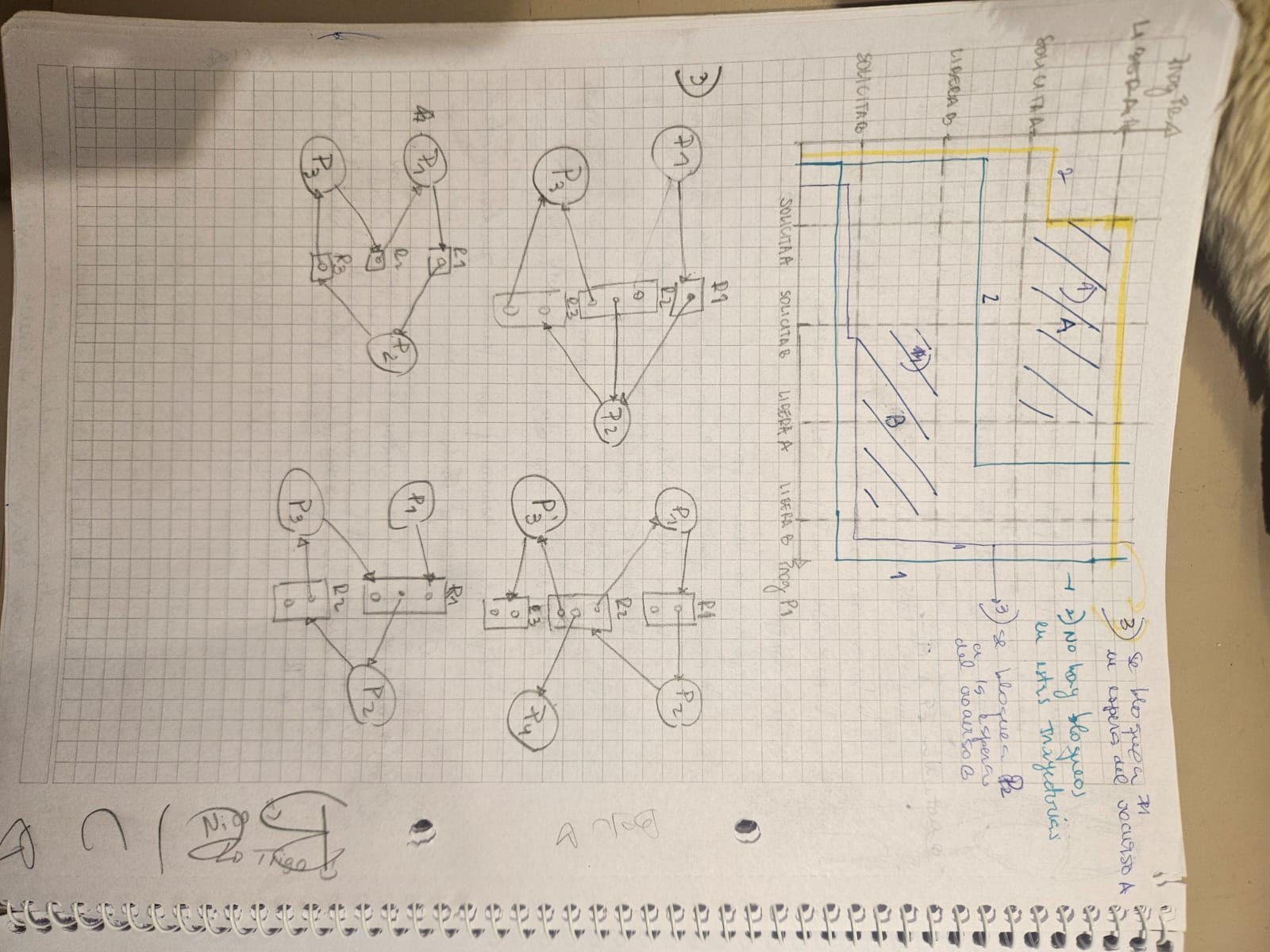
. . . . . .

Libera B Libera A

1. Construya un Diagrama de progreso conjunto en donde:
   1. Identifique las zonas en la que P1 y P2 requieran A y B, y verifique

si puede existir una zona en donde pueda ocurrir un Interbloqueo inevitable .

* 1. Muestre al menos una trayectoria de evolución de P1 y P2 en que no hay bloqueo ante solicitudes de ambos recursos.
  2. Muestre dos trayectorias (en una P1 y en otra P2) en que deben bloquearse a la espera de que el otro proceso evolucione y libere el recurso que posee para poder tomarlo y continuar con la ejecución.



1. Describa paso por paso lo que sucede en la secuencia temporal en 2) y en 3) en ambos casos.

En la primera trayectoria de 2) P1 solicita los recursos A y B los obtiene y los libera. Una vez que termina su ejecución, P2 ejecuta solicita los recursos y los libera.

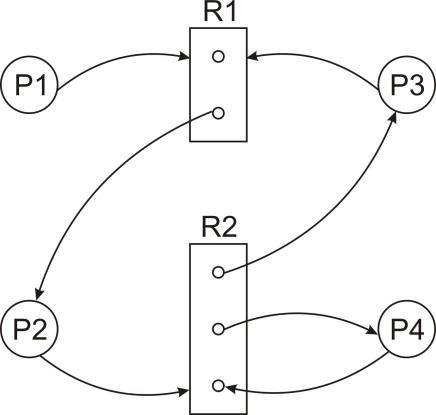
En la segunda trayectoria de 2) P2 ejecuta, solicita B y lo libera. Luego, comienza P1 que solicita los recursos A y B en dicho orden y finalmente los libera en el mismo orden. Una vez que termina, P2 solicita A, lo obtiene y por último lo libera.

En la primera trayectoria de 3), P2 comienza. Luego, se le da el uso del CPU a P1 quien solicita los recursos A y B en ese orden, y los obtiene. Luego, ejecuta P2 que solicita B pero se bloquea pues está en posesión de P1. Continúa ejecutando P1, que libera A y luego, B. Terminada su ejecución, P2 obtiene el recurso B y lo libera. Luego, solicita A, lo obtiene y finalmente lo libera.

En la segunda trayectoria de 3), ejecuta P2: solicita y obtiene el recurso B. Luego, lo libera y continua ejecutando hasta que solicita y obtiene el recurso A. Después, comienza a ejecutar P1 quien solicita el recurso A y se bloquea pues lo posee el proceso P2. Por lo tanto, ejecuta P2 hasta que libera A y termina. Una vez terminado, P1 obtiene A. Luego, solicita B y lo obtiene. Finalmente, libera los recursos A y B en ese orden y termina.

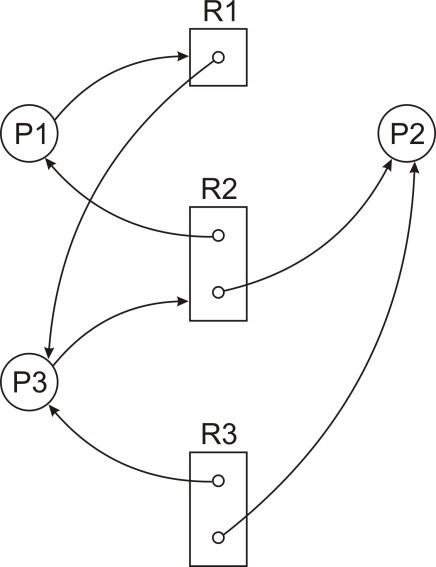
**2.-** Dados los siguientes grafos, determinar los conjuntos de vértices P y R, y el conjunto de arcos E. Además, analizar si éstos se encuentran o no en estado de abrazo mortal.

**a)**



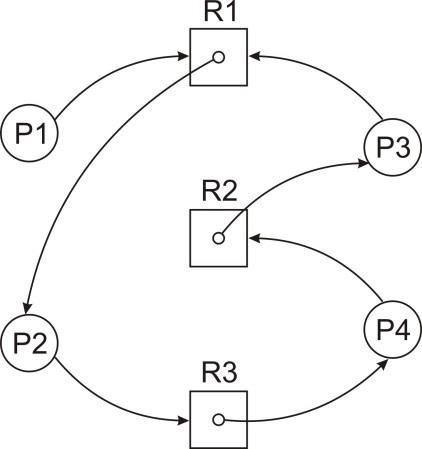
P = {P1, P2, P3, P4}, R={R1, R1, R2, R2, R2}

Hay un ciclo {R1 -> P2, P2 -> R2, R2 -> P4, P4 -> R2, R2 -> P3, P3 -> R1} por lo que hay posibilidades de abrazo mortal. Además, podemos afirmar que este ciclo es irresoluble ya que todos los procesos están bloqueados a la espera de recursos asignados a otros procesos. Como ninguno puede seguir su ejecución no se liberarán ningún recurso y se mantendrán indefinidamente bloqueados. Por lo tanto, hay abrazo mortal.

**b)**

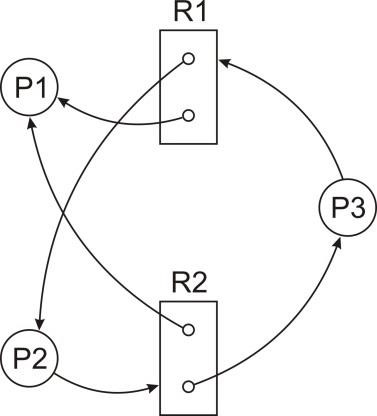
P = {P1, P2, P3}, R = {R1, R2, R2, R3, R3}

Hay un ciclo {P3 -> R2, R2 -> P1, P1-> R1, R1 ->P3} que podría indicar una situación de abrazo mortal. Sin embargo, el ciclo no es irresoluble entonces esto no llevará a una situación de abrazo mortal porque P2 es un proceso que puede seguir su ejecución ya que tiene los recursos asignados. Luego, este libera los recursos al terminar su ejecución. Como P3 solicita R2, recibirá el recurso y romperá el ciclo evitando el abrazo mortal.

****

P = {P1, P2, P3, P4}, R = {R1, R2, R3}

Hay un ciclo {R1 -> P2, P2 -> R3, R3 -> P4, P4 -> R2, R2 -> P3, P3 -> R1} por lo que hay posibilidades de abrazo mortal. Además, podemos afirmar que este ciclo es irresoluble ya que todos los procesos están bloqueados a la espera de recursos asignados a otros procesos. Como ninguno puede seguir su ejecución no se liberarán ningún recurso y se mantendrán indefinidamente bloqueados. Por lo tanto, hay abrazo mortal.

****

P = {P1, P2, P3} R = {R1, R1, R2, R2}

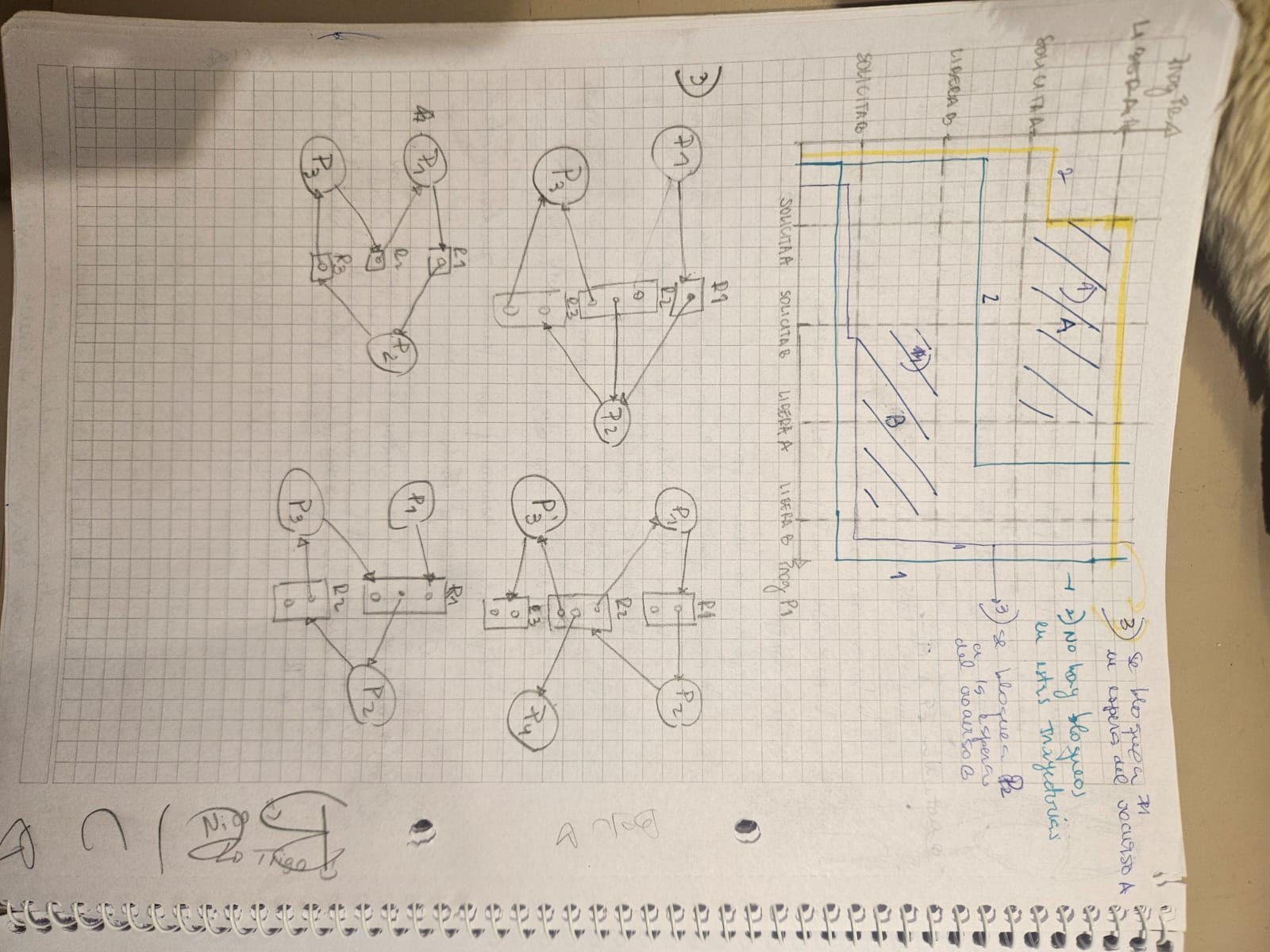
Hay un ciclo {R1 -> P2, P2 -> R2, R2 -> P3, P3 ->R1} que podría indicar una situación de abrazo mortal. Sin embargo, el ciclo no es irresoluble entonces esto no llevará a una situación de abrazo mortal porque P1 es un proceso que puede seguir su ejecución ya que tiene los recursos asignados. Luego, este libera los recursos al terminar su ejecución. Como P2 solicita R2 y P3 solicita R2, recibirán su respectivo el recurso que ha sido liberado por P1 y romperá el ciclo evitando el abrazo mortal.

**3.-** Dados los conjuntos de vértices (procesos y recursos) y arcos, obtenga los correspondientes grafos de asignación de recursos. Los vértices repetidos en el conjunto de recursos indican que se cuenta con más de un recurso de esa clase. Analice cada uno de los grafos obtenidos e indique si representan o no una situación de abrazo mortal.

1. P = {P1, P2, P3}

R = {R1, R2, R2, R2, R3, R3}

E = {P1R1, R1P2, P2R3, R2P2, R2P3, R3P3}

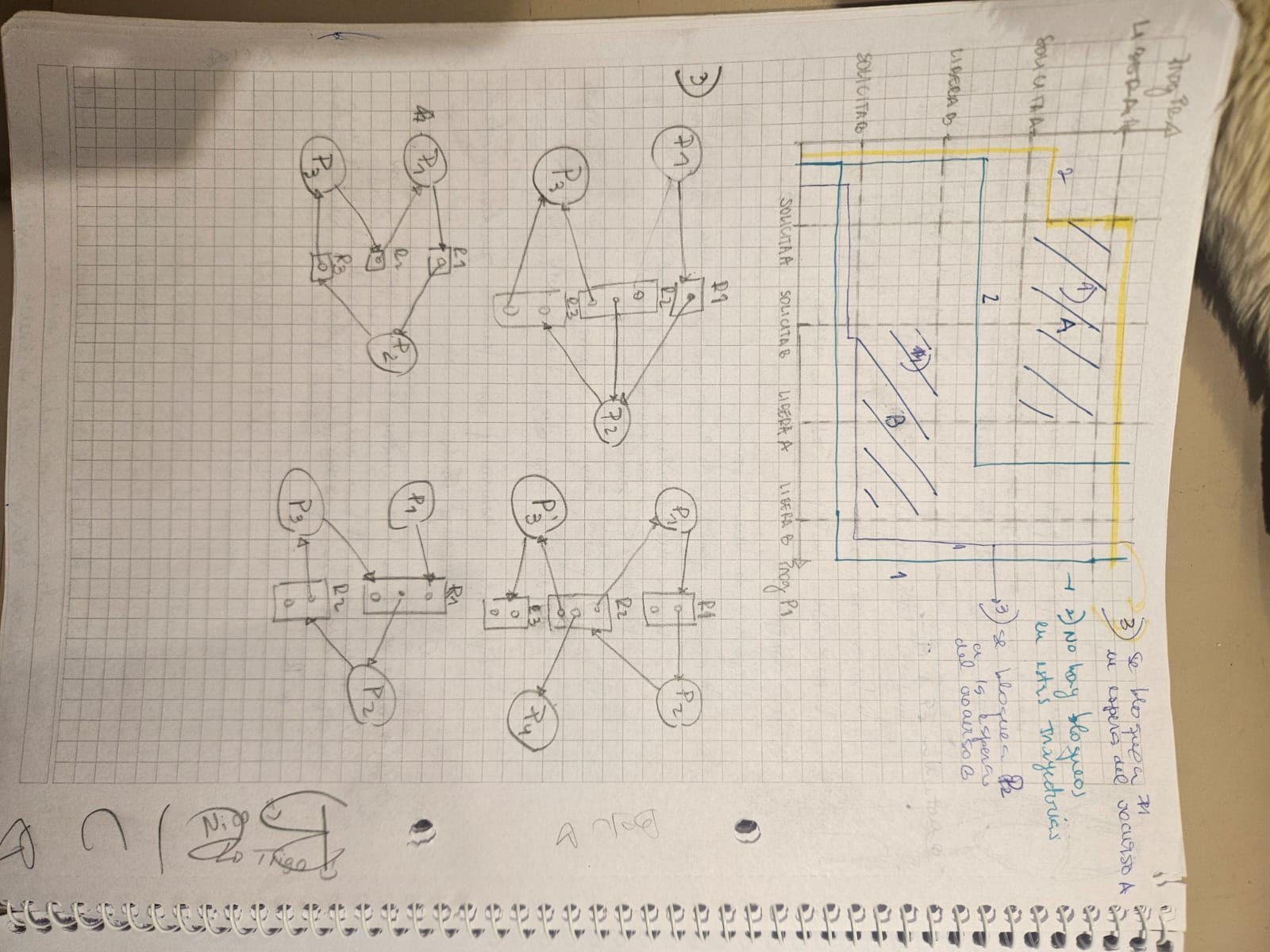


No hay situación de abrazo mortal ya que no hay ningún ciclo en el grafo de asignación de recursos que representa una posible situación de abrazo mortal.

1. P = {P1, P2, P3, P4}

R = {R1, R1, R2, R2, R2, R3, R3}

E = {P1R1, R1P2, P2R2, R2P1, R2P4, R2P3, P3R3}

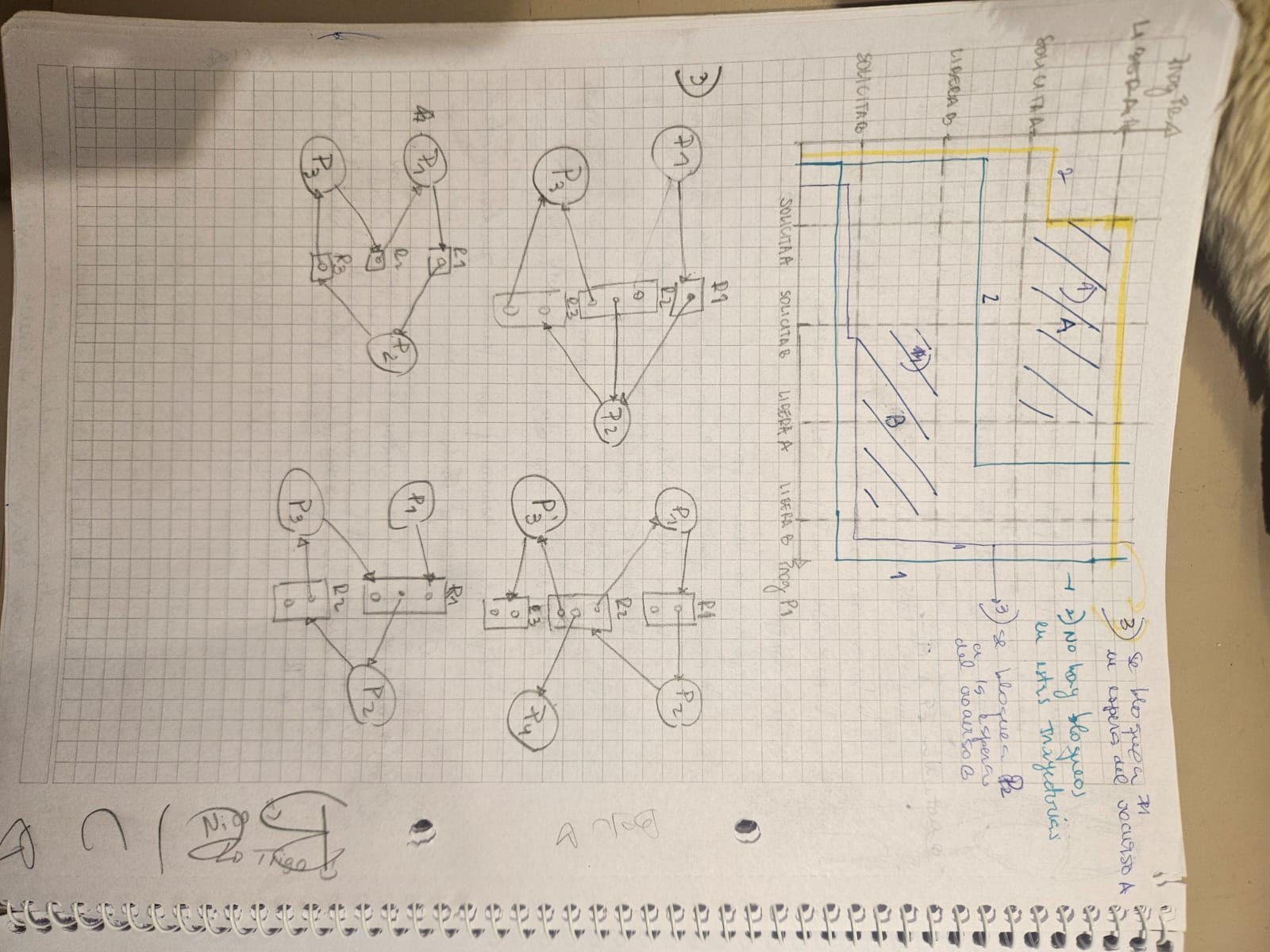


Hay un ciclo {P1 -> R1, R1 -> P2, P2 -> R2, P2 ->P1} que podría indicar una situación de abrazo mortal. Sin embargo, el ciclo no es irresoluble entonces esto no llevará a una situación de abrazo mortal porque P4 es un proceso que puede seguir su ejecución ya que tiene los recursos asignados. Luego, este liberará los recursos al terminar su ejecución. Como P2 solicita R2, recibirá el recurso que ha sido liberado por P4 y romperá el ciclo evitando el abrazo mortal. También, existe una instancia de R1 libre que puede ser asignada a P1 por el sistema operativo que rompería con el ciclo.

1. P = {P1, P2, P3}

R = {R1, R2, R3}

E = {P1R1, R1P2, P2R3, R3P3, P3R2, R2P1}

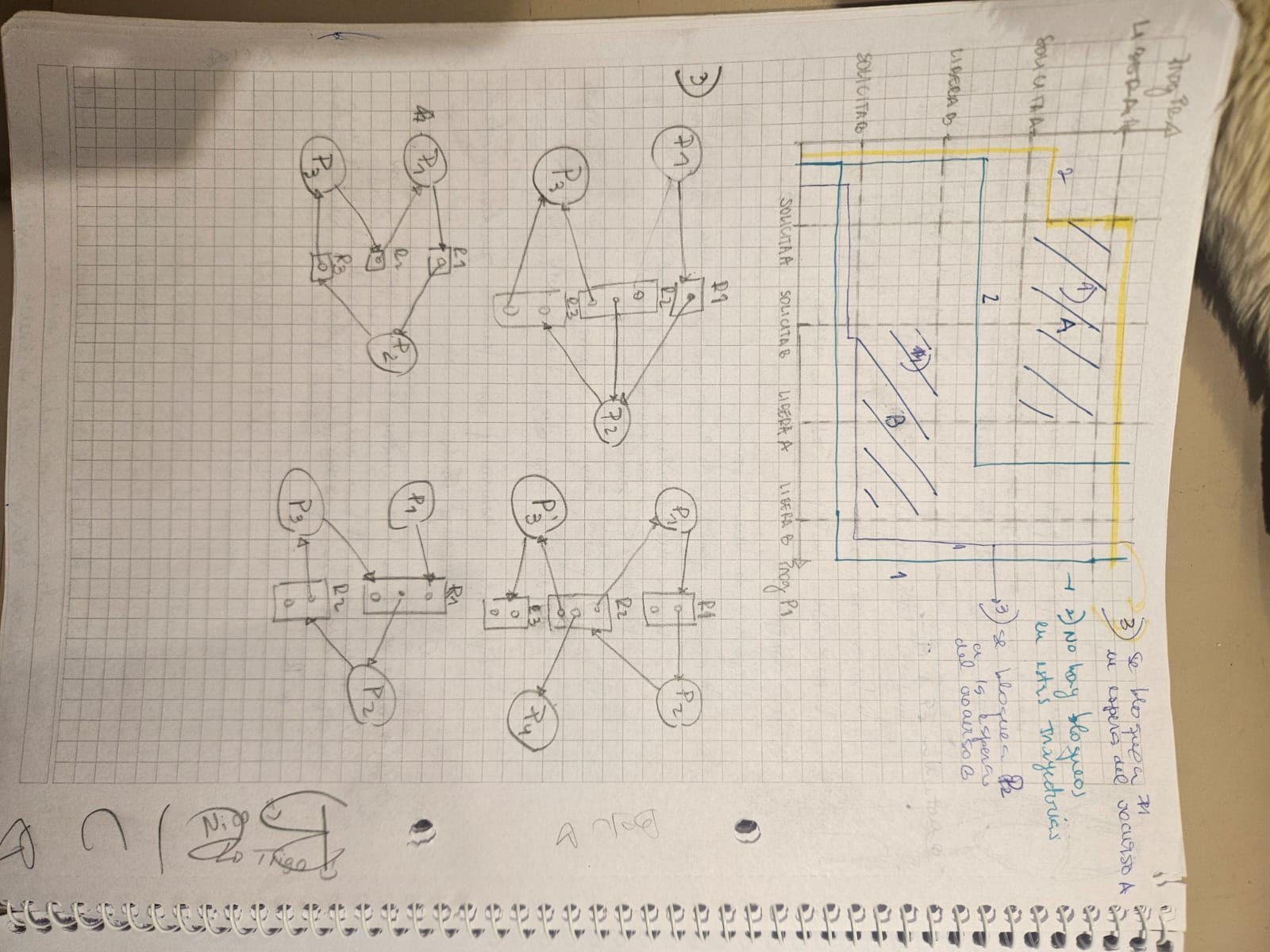


Hay un ciclo {P1 -> R1, R1 -> P2, P2 -> R3, P3 -> R2, R2 -> P1} por lo que hay posibilidades de abrazo mortal. Además, podemos afirmar que este ciclo es irresoluble ya que todos los procesos están bloqueados a la espera de recursos asignados a otros procesos. Como ninguno puede seguir su ejecución no se liberarán ningún recurso y se mantendrán indefinidamente bloqueados. Por lo tanto, hay abrazo mortal.

1. P = {P1, P2, P3}

R = {R1, R1, R1, R2, R2}

E = {P1R1, R1P2, P2R2, R2P3, P3R1}

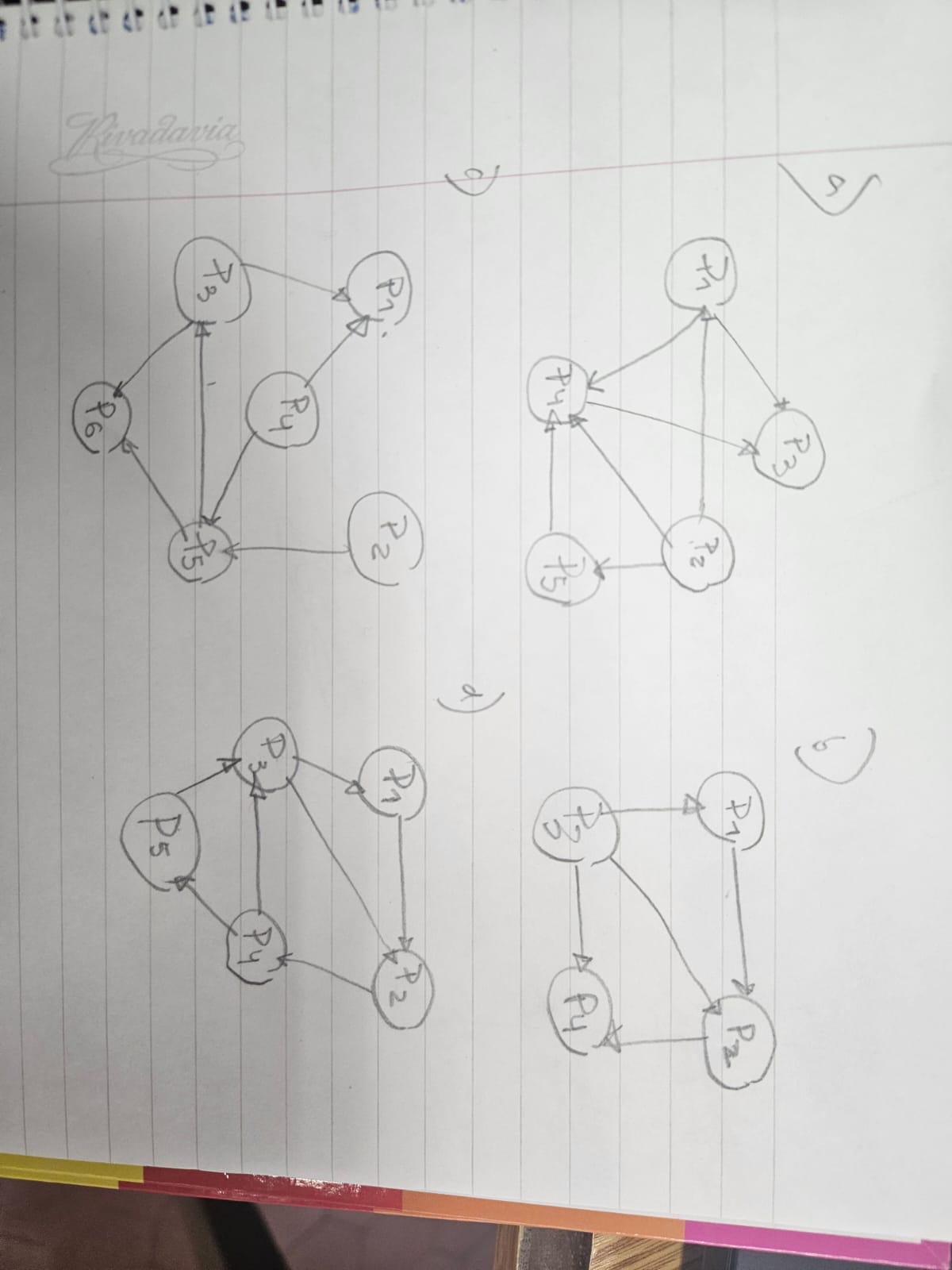


Hay un ciclo {R1 -> P2, P2 -> R2, R2 -> P3, P3 -> R1} por lo que hay posibilidades de abrazo mortal. Además, podemos afirmar que este ciclo resoluble ya que hay instancias de los recursos libres que pueden ser asignadas a los procesos por el sistema operativo y que rompan con el ciclo. Por lo que no hay situación de abrazo mortal.

**4.-** Analice los siguientes grafos “wait-for”, definidos a partir de sus conjuntos de vértices y arcos, e indique si los mismos representan o no, una situación de abrazo mortal.

1. P = {P1, P2, P3, P4, P5}

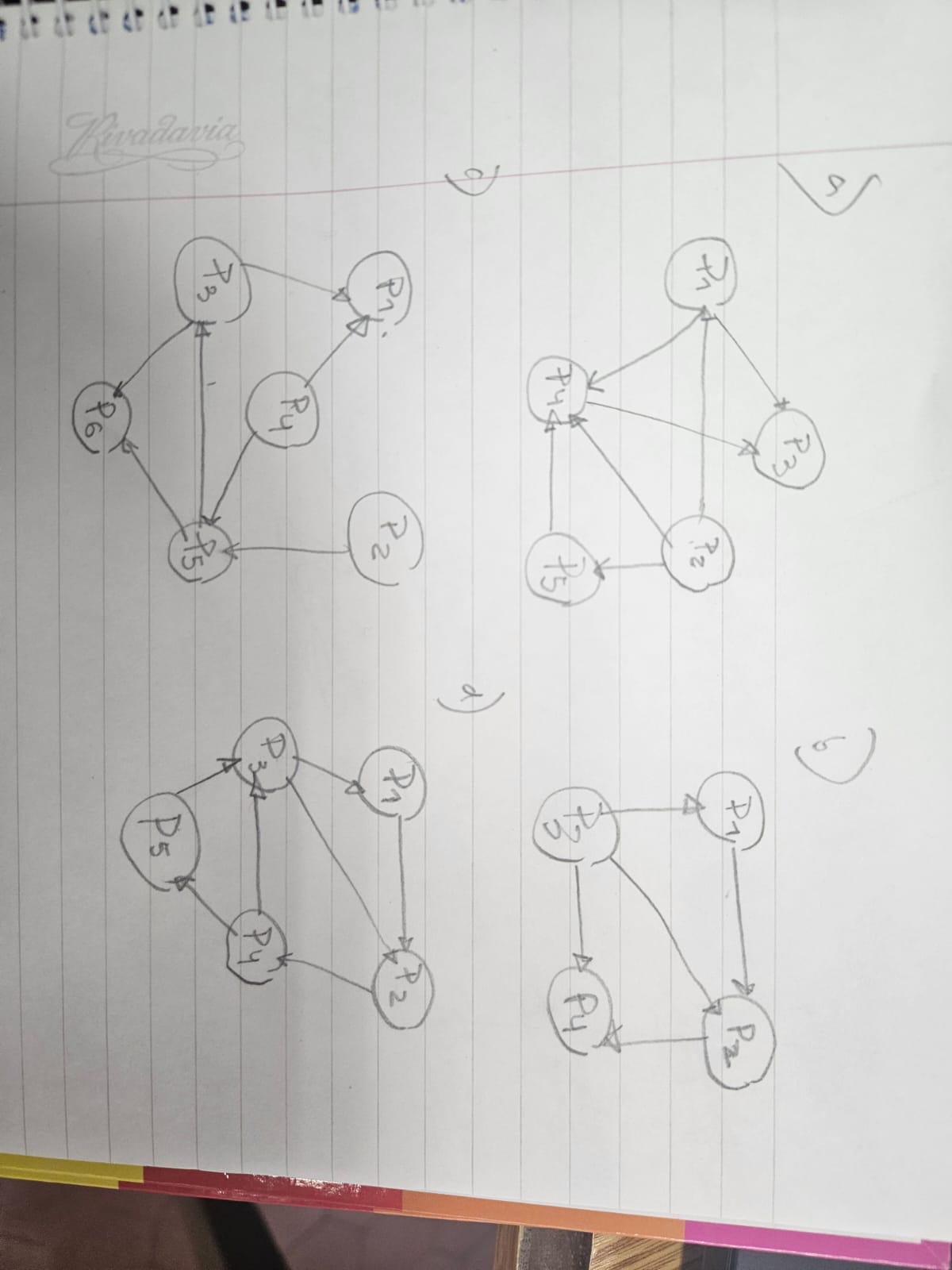
E = {P1P3, P1P4, P2P1, P2P4, P2P5, P4P3, P5P4}



El grafo no presenta ningún ciclo por lo que no se representa una situación de abrazo mortal.

1. P = {P1, P2, P3, P4}

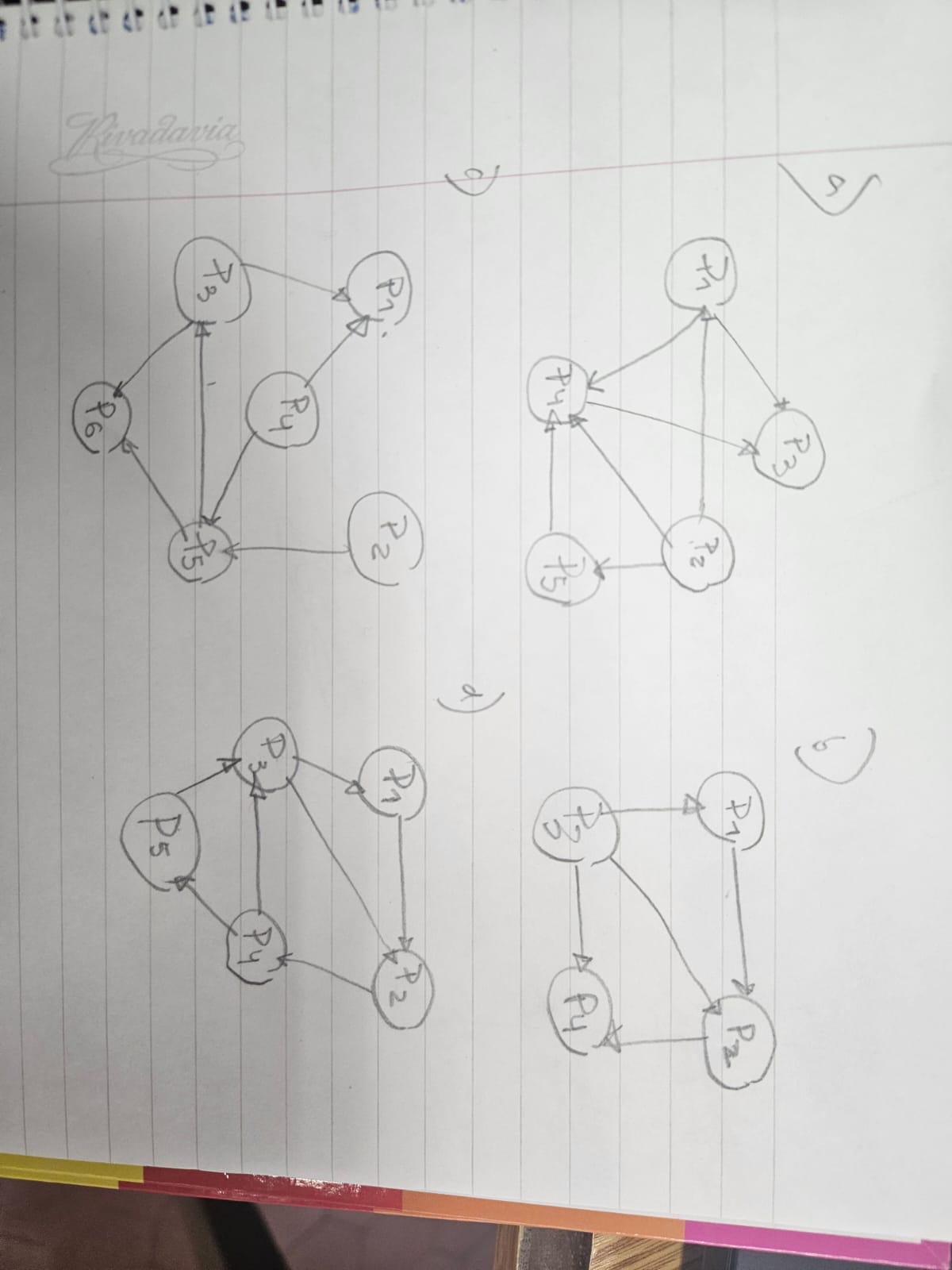
E = {P1P2, P2P4, P3P1, P3P2, P3P4 }



El grafo no presenta ningún ciclo por lo que no se representa una situación de abrazo mortal.

1. P = {P1, P2, P3, P4, P5, P6}

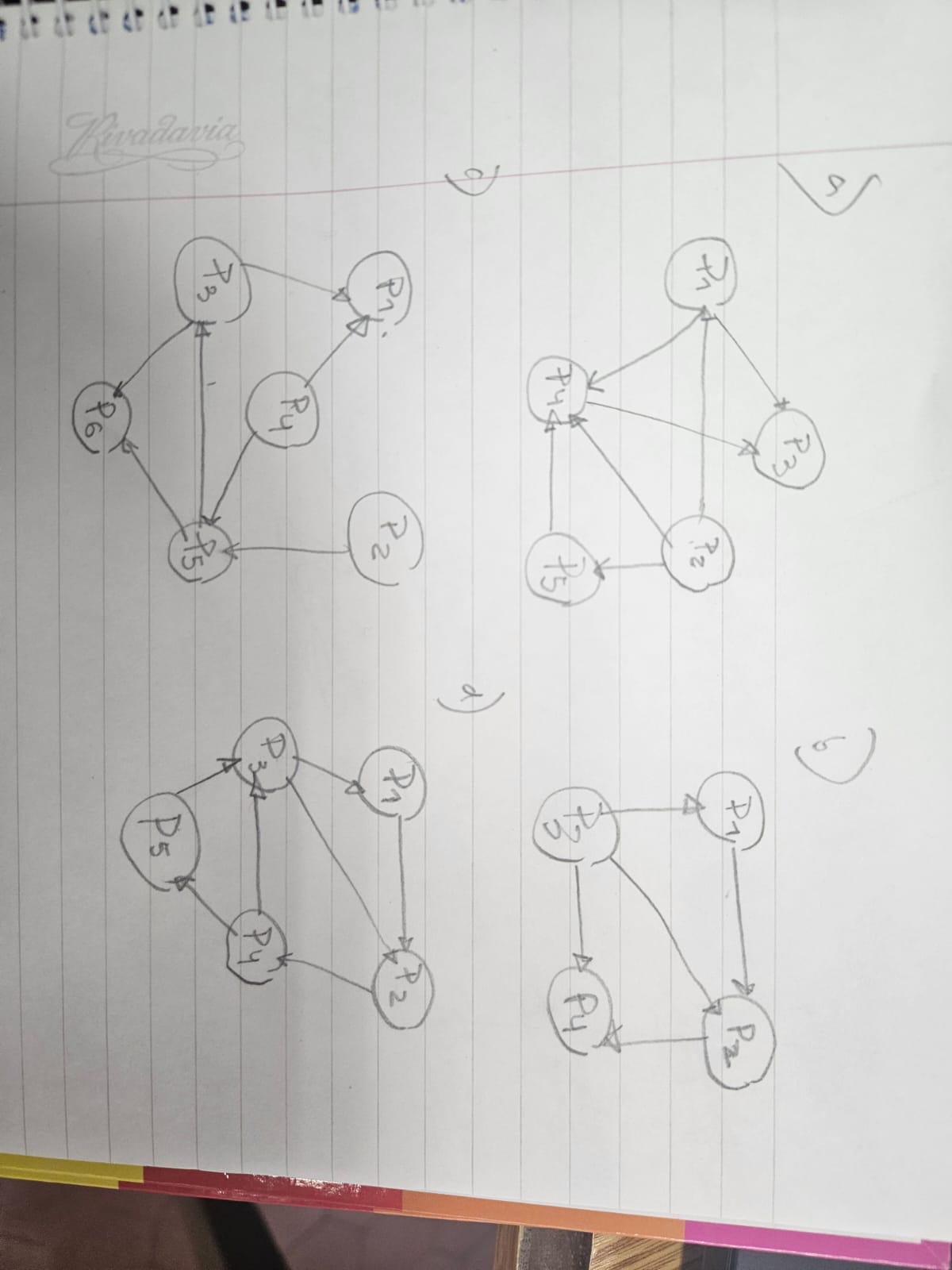
E = {P2P5, P3P1, P3P6, P4P1, P4P5, P5P3, P5P6}



El grafo no presenta ningún ciclo por lo que no se representa una situación de abrazo mortal.

1. P = {P1, P2, P3, P4, P5}

E = {P1P2, P2P4, P3P1, P3P2, P4P3, P4P5, P5P3}



Hay dos ciclos {P1P2, P2P4, P3P1, P4P3} y {P1P2, P2P4, P3P1, P4P5, P5 -> P3}. En los grafos wait-for esto es condición suficiente para afirmar que hay situación de abrazo mortal.

**5.-** ¿Es posible tener abrazo mortal en el cual participe un solo proceso?

No puede suceder. La situación de abrazo mortal implica un conjunto de procesos que se encuentran en situación de interbloqueo permanente.

**6.-** ¿Qué dificultad puede surgir cuando un proceso debe realizar un “rollback” como resultado

de un abrazo mortal?

* **Pérdida de trabajo ya realizado**
* Todo lo ejecutado desde el último checkpoint se descarta.
* Si los checkpoints son poco frecuentes, la pérdida de trabajo puede ser significativa y costosa en términos de CPU y tiempo de ejecución.
* **Repetición innecesaria de operaciones**
* El proceso puede volver a ejecutar operaciones costosas (como cálculos largos, accesos a disco o consultas a base de datos).
* Esto afecta la eficiencia del sistema.
* **Posible corrupción de datos**
* Si el proceso había modificado archivos, bases de datos o estructuras compartidas y esos cambios no fueron revertidos correctamente al estado del checkpoint, puede quedar el sistema en un **estado inconsistente**.
* Ejemplo: una transacción bancaria que acreditó dinero en una cuenta pero que al hacer rollback no revierte la operación opuesta.
* **Posible repetición del interbloqueo**
* Tras volver a un checkpoint, el proceso podría nuevamente solicitar el recurso que lo llevó al deadlock y volver a caer en el mismo abrazo mortal.
* En ese caso, se entra en un **ciclo de rollback sin progreso**.
* **Sobrecarga del sistema**
* Guardar checkpoints consume memoria y almacenamiento.
* Hacer rollback frecuente puede ralentizar el sistema completo, afectando a procesos que no estaban involucrados en el interbloqueo.

**7.-** ¿Cómo es posible detectar que existen en el sistema procesos en estado de inanición?

¿Cómo se puede detectar que existen procesos en abrazo mortal?

**Inanición (starvation):**

Un proceso está en inanición cuando permanece demasiado tiempo en espera de un recurso que siempre es otorgado a otros.

Se detecta si un proceso lleva esperando un tiempo excesivo sin avanzar, mientras otros sí lo hacen. En algunos sistemas, se mide el tiempo de espera en colas y si supera cierto umbral, se considera inanición.

**Abrazo mortal (deadlock):**

Se detecta utilizando grafos de asignación de recursos o grafos wait-for. Si en el grafo no hay ciclos, no existe interbloqueo.

Si existe un ciclo, puede haber interbloqueo. Si ese ciclo representa una espera circular irresoluble, entonces efectivamente hay abrazo mortal. En el caso de grafos wait-for, la existencia de un ciclo es condición suficiente para afirmar que existe una situación de abrazo mortal

El sistema operativo puede ejecutar periódicamente un algoritmo que busque ciclos en el grafo wait-for para confirmar la existencia de un deadlock.

**8.-** Enuncie las condiciones necesarias para la existencia de abrazo mortal.

Las condiciones necesarias para la existencia de abrazo mortal son:

* **Exclusión mutua**: Sólo un proceso puede utilizar un recurso en cada

momento. Ningún proceso puede acceder a una unidad de un recurso que se

ha asignado a otro.

* **Retención y espera**: Un proceso puede mantener los recursos asignados

mientras espera la asignación de otros recursos.

* **Sin expropiación:** No se puede forzar la expropiación de un recurso a un

proceso que lo posee.

* **Espera circular**: Existe una lista cerrada de procesos, de tal manera que cada

uno posee al menos un recurso necesitado por el siguiente proceso de la lista.

**9.-** Considere el siguiente plan de asignación de recursos: la solicitud y liberación de recursos son permitidas en cualquier momento. Si una solicitud por recursos no puede ser satisfecha (debido a que no están disponibles), se chequea a los procesos que están bloqueados esperando por algún recurso. Si alguno de estos procesos posee los recursos solicitados, se les quitan y se le otorgan a los procesos que lo solicitan. (A los procesos bloqueados, se les agregan dentro de los recursos que están esperando, estos recursos que les fueron quitados).

Por ejemplo, considere un sistema con clases de recursos I, II y III. La clase I consta de cuatro unidades, La clase II consta de dos unidades; y la clase III, de dos unidades.

Si el proceso A solicita dos unidades de la clase I, dos de II y uno de III, y las obtiene. Si B solicita por una unidad de I y una de III, y también las obtiene. Si a continuación A solicita por una unidad de la clase III y pasa un estado de bloqueo (recurso no disponible). Si ahora C solicita por dos unidades de la clase I, toma una de las disponibles y toma otra del proceso A (ya que A esta bloqueado).

1. ¿Puede ocurrir un abrazo mortal en este sistema? Si su respuesta es sí, muestre un ejemplo. Si no es así, ¿qué condición necesaria no puede ocurrir?

No puede ocurrir una situación de abrazo mortal ya que en este sistema no se cumple la condición necesaria de sin expropiación ya que aquí cuando un proceso solicita el recurso y este está ocupado por un proceso bloqueado, será expropiado. Por lo tanto, cuando exista un ciclo de espera se romperá quitando el recurso a algún proceso bloqueado.

1. ¿Puede ocurrir una situación de bloqueo indefinido?

Puede pasar que un proceso entre en inanición si éste está bloqueado a la espera de recursos y cada vez que los está por obtener, le son expropiados para asignarlos a otro proceso.

Por ejemplo, si el proceso A necesita 2 unidades de recurso I, pero cada vez que está bloqueado, el sistema le quita lo que tenía y se lo da a otros procesos que llegan después. De esta manera, nunca obtendrá los recursos y estará bloqueado indefinidamente. Al favorecer siempre a otro proceso por sobre A, éste último podrá encontrarse en estado de inanición.