



Actividades UD 4. Interbloqueos



Concurrencia y Sistemas Distribuidos



Actividad 1

- ▶ Indique si las siguientes afirmaciones son V/F:
- ▶ Las condiciones de Coffman...

	... son 4: exclusión mutua, progreso, espera limitada y “no expulsión”.
	... son solamente cuatro, que incluyen “retención y espera” y “determinismo”.
	... son condiciones suficientes; es decir, si se cumplen todas simultáneamente se puede garantizar que ha ocurrido un interbloqueo.
	... son condiciones necesarias. Basta con evitar una de ellas para que no pueda generarse un interbloqueo.
	... son necesarias y suficientes. Si se cumplen todas, entonces hay interbloqueo. Si hay interbloqueo, entonces seguro que se cumplen todas.
	... proporcionan la base para el diseño de estrategias de prevención de interbloqueos.
	Una de las condiciones implica que “mientras un recurso está asignado a un hilo, otros no pueden usarlo”.
	Una de las condiciones implica que “un recurso asignado sólo lo puede liberar su dueño”.



Actividad 1

- ▶ Indique si las siguientes afirmaciones son V/F:
- ▶ Las condiciones de Coffman...

F	... son 4: exclusión mutua, progreso, espera limitada y “no expulsión”.
F	... son solamente cuatro, que incluyen “retención y espera” y “determinismo”.
F	... son condiciones suficientes; es decir, si se cumplen todas simultáneamente se puede garantizar que ha ocurrido un interbloqueo.
V	... son condiciones necesarias. Basta con evitar una de ellas para que no pueda generarse un interbloqueo.
F	... son necesarias y suficientes. Si se cumplen todas, entonces hay interbloqueo. Si hay interbloqueo, entonces seguro que se cumplen todas.
V	... proporcionan la base para el diseño de estrategias de prevención de interbloqueos.
V	Una de las condiciones implica que “mientras un recurso está asignado a un hilo, otros no pueden usarlo”.
V	Una de las condiciones implica que “un recurso asignado sólo lo puede liberar su dueño”.



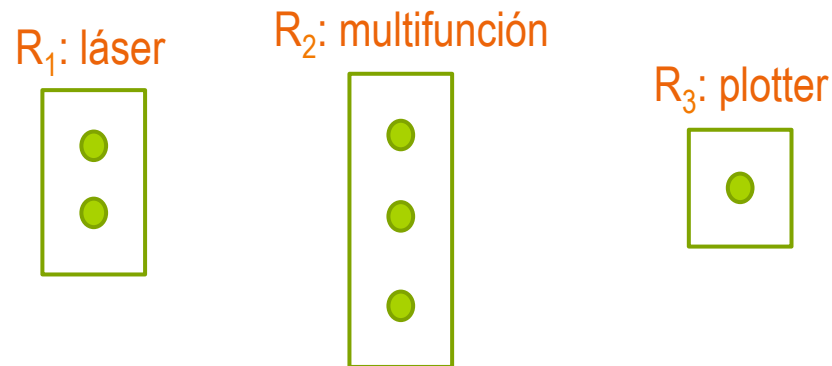
Actividad 2

- ▶ En una determinada empresa dedicada a la edición y maquetación de documentos existen algunas impresoras, equipos multifunción y plotters, con las siguientes características:
 - ▶ 2 impresoras láser monocromas, que manejan papel de formato A4.
 - ▶ 3 equipos multifunción (impresora + escáner) de chorro de tinta (negro / color) que manejan papel de formato A4.
 - ▶ 1 plotter monocromo para trazado de documentos capaz de utilizar papel de formato A2. Este dispositivo no se llegará a utilizar para impresión de documentos de texto y/o gráficos que puedan ser manejados por el resto de equipos.
- ▶ Las peticiones que podrán generar las aplicaciones son de los siguientes tipos:
 - 1) Imprimir documentos en monocromo en un formato de papel A4.
 - 2) Imprimir documentos en color en un formato de papel A4.
 - 3) Escanear documentos en color en un formato A4.
 - 4) Trazar planos/mapas en formato A2.
- ▶ Discuta qué recursos tendrán que considerarse y cuántas instancias de cada uno de ellos habrá a la hora de modelar este sistema mediante un Grafo de Asignación de Recursos (GAR).



Actividad 2

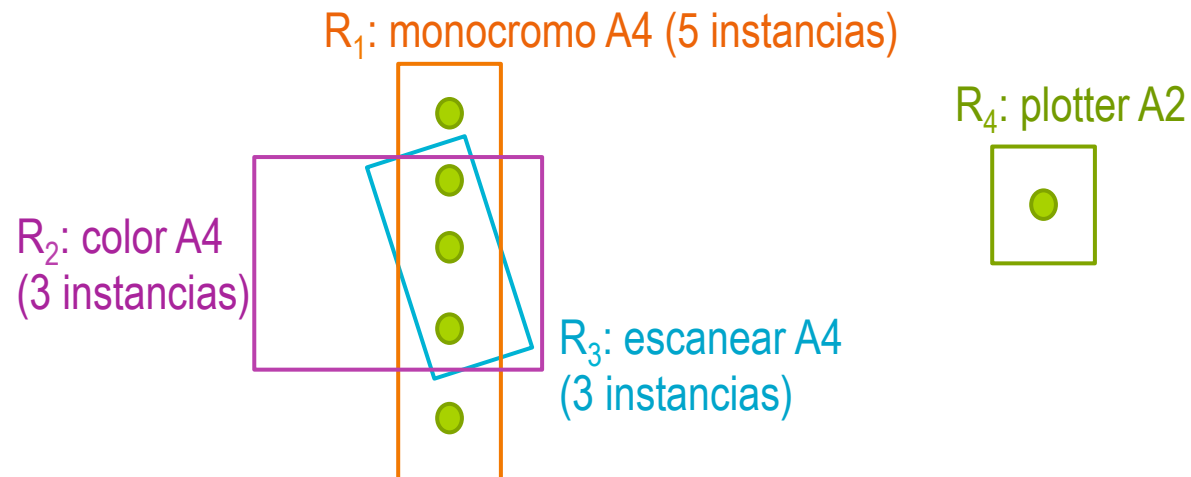
- ▶ Alternativa 1 (centrada en los dispositivos):



- ▶ Esta alternativa está orientada a lo que suele ocurrir en los ordenadores personales donde primero seleccionas el dispositivo desde la aplicación y luego seleccionas las propiedades.

Actividad 2

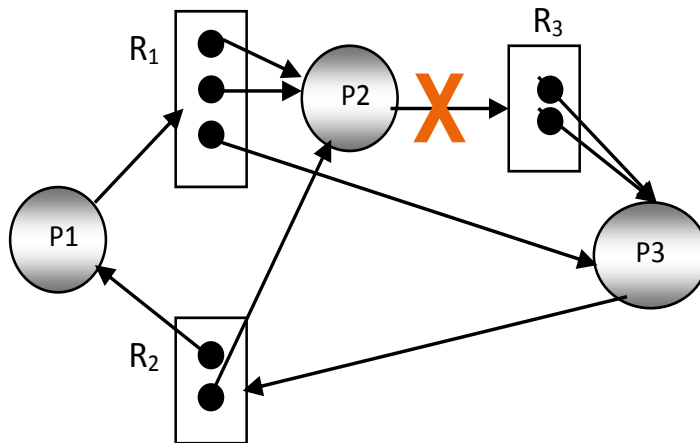
- ▶ Alternativa 2 (centrada en las peticiones):



- ▶ Cuando el servicio de impresión nos da una interfaz en donde nos obliga a escoger entre las alternativas de impresión 1), 2), 3) y 4).

Actividad 3

- Dado el siguiente estado de un sistema y suponiendo que tras esto ningún proceso realiza ninguna petición más y no liberan los recursos hasta su terminación:



Hay que eliminar alguna arista de petición.
Por ejemplo: eliminando la petición $P_2 \rightarrow R_3$
Secuencia de terminación:

- P_2 finaliza y libera lo que tiene asignado
- P_1 puede finalizar, liberando R_2
- P_3 puede finalizar.

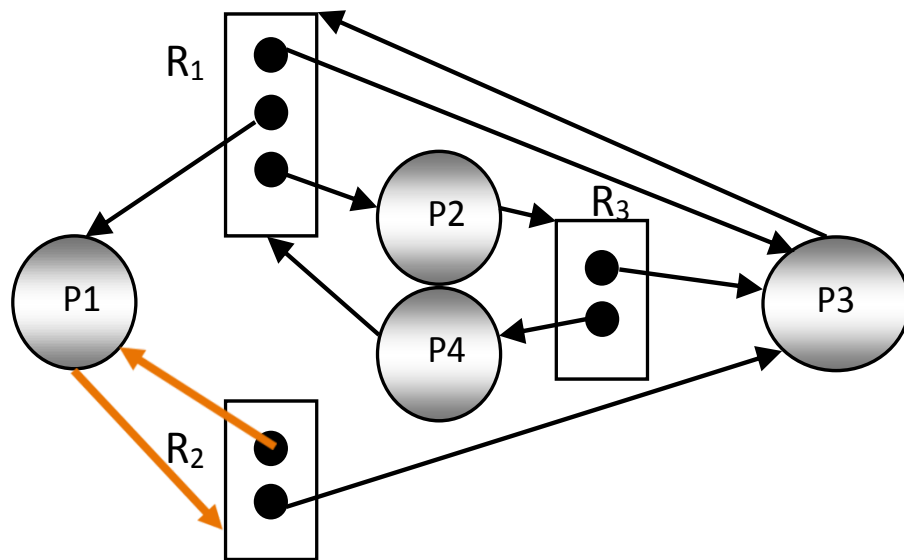
- a) Indique si hay o no interbloqueo. Si lo hay, diga qué procesos están interbloqueados y si no lo hay, dé al menos una secuencia de terminación.

Todos los procesos están en un **interbloqueo**. Todos esperan que se libere alguna instancia.

- b) Redibuje el estado del sistema anterior, para que presente un estado del sistema totalmente contrario al indicado en el apartado anterior. Es decir, si en el apartado “a” el sistema presenta interbloqueo, ahora añada/elimine las flechas necesarias para que el sistema no presente interbloqueo. Y viceversa.

Actividad 4

- ▶ Dado el siguiente estado de un sistema y suponiendo que a los procesos P1 y P4 les queda pendiente todavía realizar una petición cada uno de una instancia de R2 (se desconoce el orden en que efectúan tales peticiones) y que tras esto ningún proceso realiza ninguna petición más y no liberan los recursos hasta su terminación:



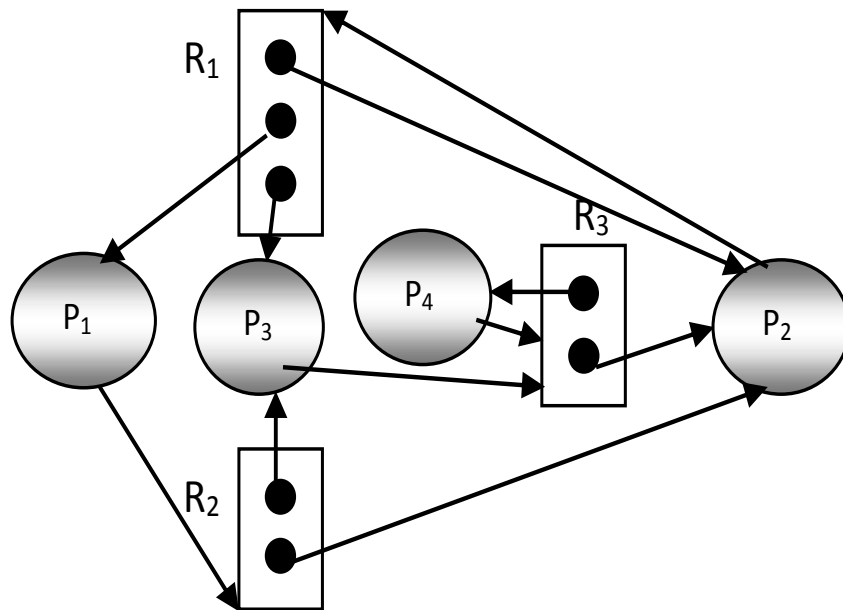
- P1 solicita R₂
- Como hay una instancia libre se le asigna, P4 está esperando una instancia de R₁

- Secuencia de terminación:
- P1 puede finalizar, liberando R₁ y R₂
- P4 podría continuar usando R1, al solicitar R2 se le asignaría y finalizaría liberando R1, R2 y R3
- P2 podría finalizar liberando R1 y R3
- P3 podría finalizar
- Por tanto **NO** hay interbloqueo

- ▶ Indique si hay o no interbloqueo. Si lo hay, diga qué procesos están interbloqueados y si no lo hay, dé al menos una secuencia de terminación.
- ▶ **Nota:** El proceso P4 realizará la petición a la instancia de R2 una vez haya obtenido la instancia de R1 previamente solicitada (según se muestra en el GAR).

Actividad 5

- ▶ Dado el siguiente estado de un sistema y suponiendo que los procesos ya no realizarán más peticiones durante el resto de su ejecución y no liberarán los recursos hasta su terminación:



- Todos los procesos están esperando a usar algún recurso que está asignado a otro

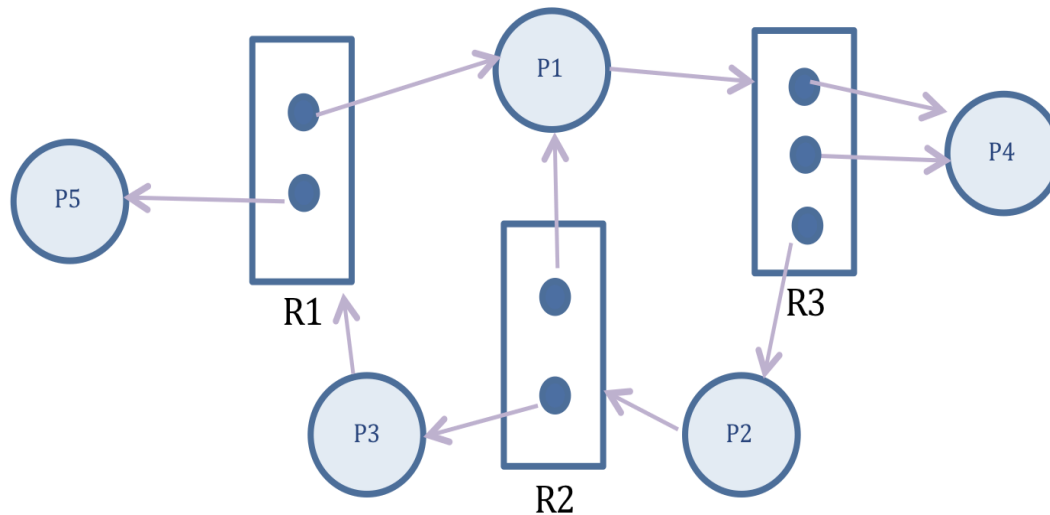


- Hay un **Interbloqueo** en el que intervienen todos los procesos.
- No es posible encontrar una secuencia de terminación de los procesos.

- ▶ Indique si hay o no interbloqueo. Si lo hay, diga qué procesos están interbloqueados y si no lo hay, dé al menos una secuencia de terminación.

Actividad 6

- Dado el siguiente grafo de asignación de recursos, indique si las afirmaciones son V/F:



1. Existe más de un ciclo.	V
2. Podemos encontrar una secuencia segura.	V
3. El proceso P1 forma parte de un interbloqueo (o más de uno).	F
4. Es un grafo incorrecto, pues tenemos dos recursos de instancia doble, pero hay un recurso con 3 instancias.	F
5. El proceso P4 no puede tener dos instancias de un mismo recurso asignadas a la vez.	F



Actividad 7

- En un sistema se encuentran en ejecución cinco procesos: P0, P1, P2, P3 y P4 que utilizan los recursos R0, R1, R2, R3, R4 y RC. Inicialmente la cantidad de instancias de cada tipo es la indicada en la siguiente tabla:

Recurso	R0	R1	R2	R3	R4	RC
Cantidad	1	1	1	1	1	5

- El perfil de ejecución de un proceso P_i es distinto para los procesos pares e impares y es el indicado en la tabla siguiente:

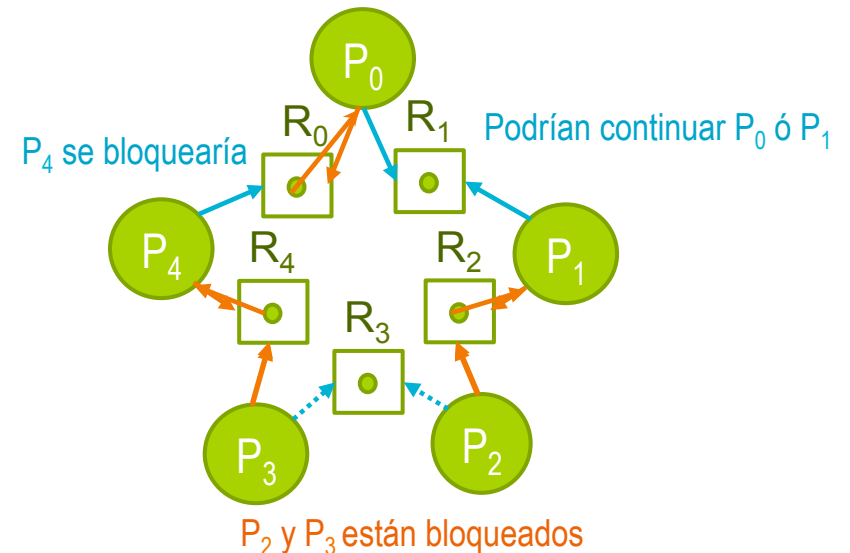
Perfil de los procesos pares	Perfil de los procesos impares
<pre>while (TRUE) { Petición(RC); Petición(Ri); Petición(R((i+1) % 5)); UsoDeLosRecursos(); Libera(Ri); Libera(R((i+1) % 5)); Libera(RC); SecciónRestante(); };</pre>	<pre>while (TRUE) { Petición(RC); Petición(R((i+1) % 5)); Petición(Ri); UsoDeLosRecursos(); Libera(Ri); Libera(R((i+1) % 5)); Libera(RC); SecciónRestante(); };</pre>

Nota: Cada petición de recursos solicita un solo ejemplar del recurso en concreto y bloquea al proceso solicitante si el recurso no está disponible.

Actividad 7

- ▶ ¿Es posible que en el sistema se produzca un interbloqueo? Razone la respuesta y muestre una traza (o varias) para justificarlo.

- ▶ No es posible que se produzca un interbloqueo.
- ▶ No se puede dar la condición de la espera circular, pues los procesos pares e impares solicitan los recursos en orden diferente, de forma que siempre habrá algún proceso que obtendrá los dos recursos que necesita y no esperará por ninguno.



- ▶ ¿Y si $RC=4$, pero todos ellos tienen el mismo perfil (sin distinguir entre pares e impares)?
 - ▶ Tampoco es posible que se produzca un interbloqueo.
 - ▶ Limitando a 4 el n° de procesos en la sección crítica, siempre habrá un proceso que pueda obtener los dos recursos que necesita. No se puede dar la condición de la espera circular entre todos ellos, ni retener y esperar para todos ellos. Siempre hay uno que no retiene nada.



Actividad 8

- Sea el siguiente monitor (que sigue la variante de Lampson/Redell):

```
monitor ResourceController {  
    private int availableItems;  
    private condition c;  
    private int k, retrying, toBeNotified;  
  
    public ResourceController(int N) {  
        availableItems = N;  
        k = retrying = toBeNotified = 0;  
    }  
  
    entry void request(int p) {  
        if (p < availableItems) {  
            availableItems -= p;  
        } else {  
            retrying++;  
            retry(p);  
        }  
    }  
  
    entry void release(int p) {  
        availableItems += p;  
        toBeNotified = retrying;  
        c.notify();  
    }  
  
    private void retry(int p) {  
        c.wait();  
        toBeNotified--;  
        if (toBeNotified > 0)  
            c.notify();  
        k++;  
        if (p < availableItems) {  
            retrying--;  
            availableItems -= p;  
        } else {  
            retry(p);  
        }  
    }  
}
```

- A) Diga cuál será el valor final de los atributos **k** y **availableItems**, suponiendo que en el constructor se ha utilizado N=15 y que los hilos H1, H2 y H3 realizan las siguientes invocaciones secuenciales:

H1 invoca request(11)

H2 invoca request(10)

H3 invoca request(8)

H1 invoca release(3)

H1 invoca release(2)

K= 4, availableItems=1



Actividad 8 -traza

- Sea el siguiente monitor (que sigue la variante de Lampson/Redell):

```
monitor ResourceController {
    private int availableItems;
    private condition c;
    private int k, retrying, toBeNotified;

    public ResourceController(int N) {
        availableItems = N; 15
        k = retrying = toBeNotified = 0;
    }

    entry void request(int p) {
        if (p < availableItems) {
            availableItems -= p;
        } else {
            retrying++; 2
            retry(p);
        }
    }

    entry void release(int p) {
        availableItems += p; 9
        toBeNotified = retrying; 2
        c.notify();
    }

    private void retry(int p) {
        c.wait(); H2 H3
        toBeNotified--; 0
        if (toBeNotified > 0)
            c.notify();
        k++; 2
        if (p < availableItems) {
            retrying--;
            availableItems -= p;
        } else {
            retry(p);
        }
    }
}
```

- A) Diga cuál será el valor final de los atributos **k** y **availableItems**, suponiendo que en el constructor se ha utilizado N=15 y que los hilos H1, H2 y H3 realizan las siguientes invocaciones secuenciales:

H1 invoca request(11)

H2 invoca request(10)

H3 invoca request(8)

H1 invoca release(3)

H1 invoca release(2)

K= 4, availableItems=1



Actividad 8 -traza

- Sea el siguiente monitor (que sigue la variante de Lampson/Redell):

```
monitor ResourceController {  
    private int availableItems;  
    private condition c;  
    private int k, retrying, toBeNotified;  
  
    public ResourceController(int N) {  
        availableItems = N;  
        k = retrying = toBeNotified = 0;  
    }  
  
    entry void request(int p) {  
        if (p < availableItems) {  
            availableItems -= p;  
        } else {  
            retrying++;  
            retry(p);  
        }  
    }  
  
    entry void release(int p) {  
        availableItems += p;  
        toBeNotified = retrying;  
        c.notify();  
    }  
  
    private void retry(int p) {  
        c.wait();  
        toBeNotified--;  
        if (toBeNotified > 0) {  
            c.notify();  
            k++;  
            if (p < availableItems) {  
                retrying--;  
                availableItems -= p;  
            } else {  
                retry(p);  
            }  
        }  
    }  
}
```

- A) Diga cuál será el valor final de los atributos **k** y **availableItems**, suponiendo que en el constructor se ha utilizado $N=15$ y que los hilos H1, H2 y H3 realizan las siguientes invocaciones secuenciales:

H1 invoca request(11)

H2 invoca request(10)

H3 invoca request(8)

H1 invoca release(3)

H1 invoca release(2)

K= 4, availableItems=1



Actividad 8

- Sea el siguiente monitor (que sigue la variante de Lampson/Redell):

```
monitor ResourceController {
    private int availableItems;
    private condition c;
    private int k, retrying, toBeNotified;

    public ResourceController(int N) {
        availableItems = N;
        k = retrying = toBeNotified = 0;
    }

    entry void request(int p) {
        if (p < availableItems) {
            availableItems -= p;
        } else {
            retrying++;
            retry(p);
        }
    }

    entry void release(int p) {
        availableItems += p;
        toBeNotified = retrying;
        c.notify();
    }

    private void retry(int p) {
        c.wait();
        toBeNotified--;
        if (toBeNotified > 0)
            c.notify();
        k++;
        if (p < availableItems) {
            retrying--;
            availableItems -= p;
        } else {
            retry(p);
        }
    }
}
```

- B) Indique si puede introducir interbloqueos. Para ello, tome como referencia la siguiente secuencia de invocaciones (a partir de la llamada a su constructor con N=15):

```
H1 invoca request(8)
H2 invoca request(6)
H3 invoca request(8)
H1 invoca request(6)
H2 invoca request(4)
```

- En caso afirmativo indique el conjunto de hilos interbloqueados

H1, H2 y H3 quedan interbloqueados, todos esperan en c.wait y ninguno hará notify.



Actividad 9

► Indique si las siguientes afirmaciones son V/F:

V	1. Las situaciones de interbloqueo pueden evitarse si el sistema monitoriza las peticiones de recursos, denegando las que generen peligro de interbloqueo.
V	2. Las situaciones de interbloqueo pueden prevenirse asignando los recursos de manera que nunca se genere una espera circular.
F	3. En muchos sistemas operativos, como Unix o Windows, se utilizan grafos de asignación de recursos para evitar los interbloques.
F	4. Las condiciones de Coffman permiten diseñar sistemas que cumplan con todas ellas, para así garantizar que no se producirán interbloques.
F	5. Las condiciones de Coffman se utilizan para diseñar algoritmos de detección de interbloques.