



UNIVERSIDAD
NACIONAL
DE LA PLATA



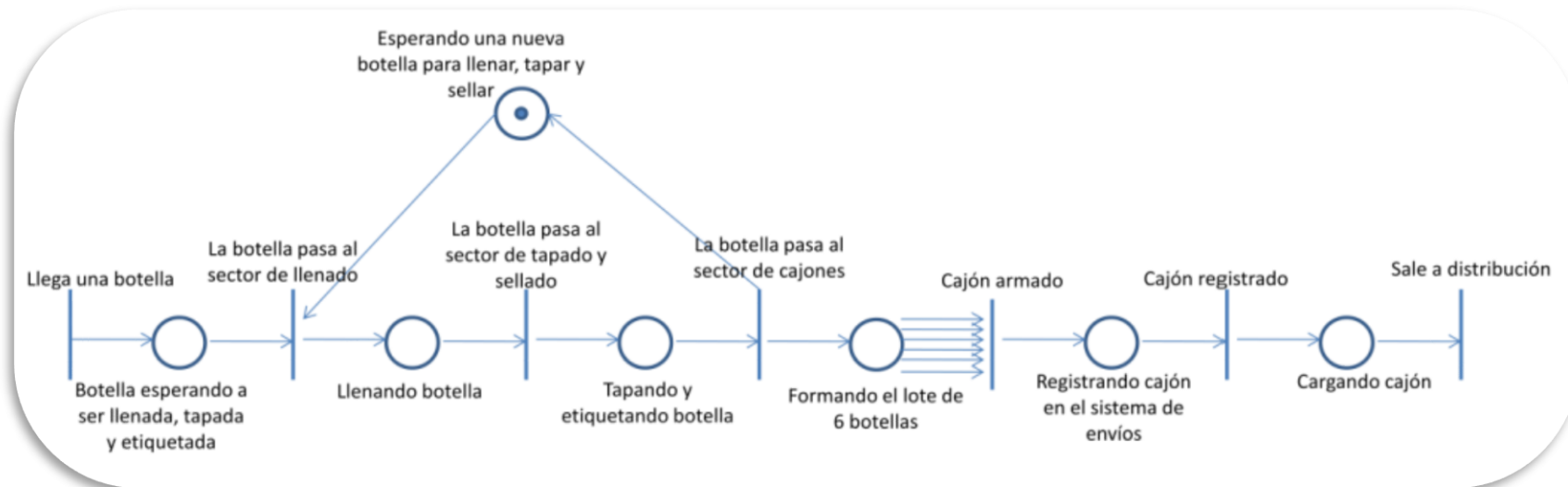
Ingeniería de Software I - 2021

Redes De Petri

Red de petri

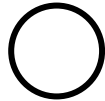
Una Red de Petri permite modelar sistemas dinámicos y concurrentes mediante una representación gráfica de eventos discretos.

La red es un grafo dirigido con 3 componentes principales: sitios, transiciones y arcos.



Red de petri - componentes

Sitio (Place):



Modela un estado o condición (dos puntos de vistas de la interpretación)

Transiciones :



Modela un evento o una acción (dos puntos de vistas de la interpretación)

Arco:



Relaciona un sitio con una transición o una transición con un sitio. Es unidireccional. La dirección se indica con una flecha.
NUNCA sitio con sitio ni transición con transición

Marca (Token):



Su función es habilitar/deshabilitar transiciones para controlar la ejecución de la red. Se colocan en los sitios. Pueden haber más de uno en un sitio.

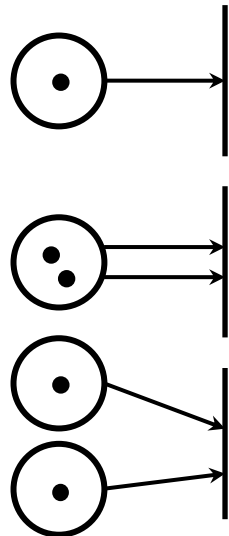


Red de petri - Funcionamiento

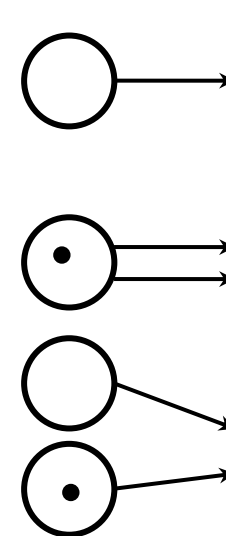
Transición habilitada:

una transición se encuentra habilitada cuando al menos hay un token por cada arco que llega a la transición.

Ejemplo de transiciones habilitadas



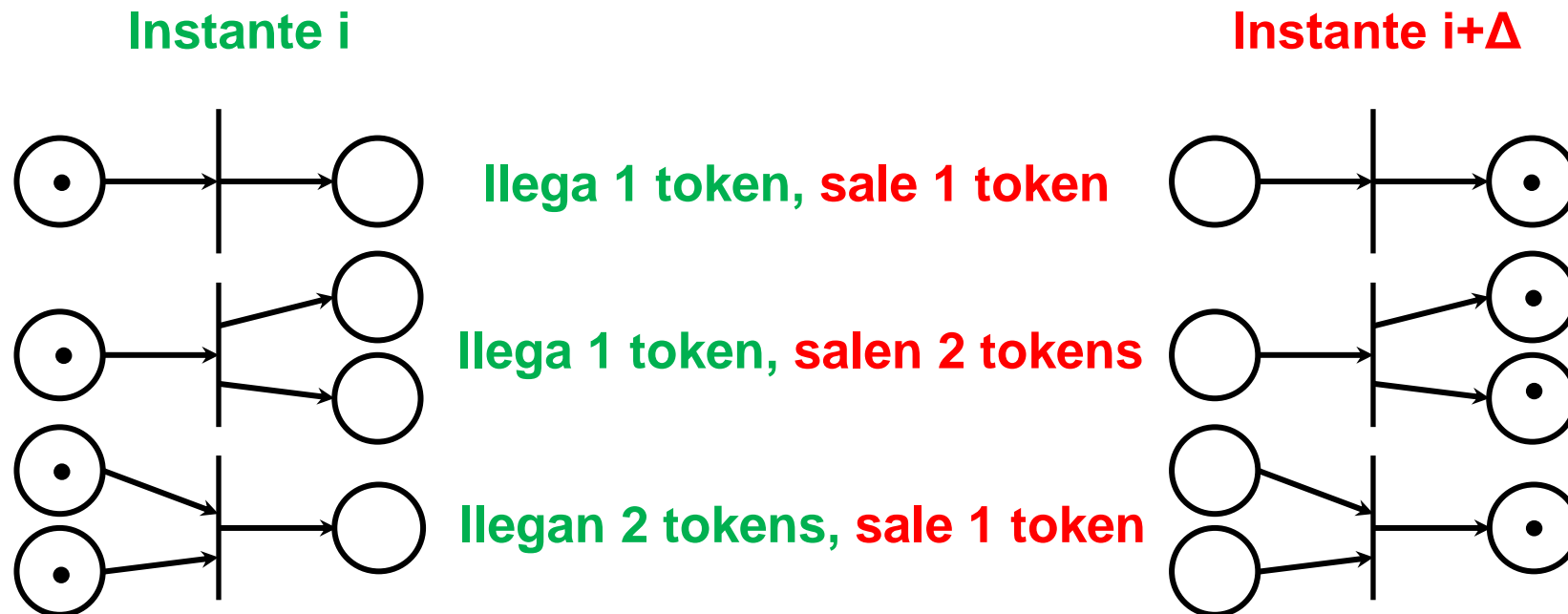
Ejemplo de transiciones deshabilitadas



Red de petri - Funcionamiento

Propagación de tokens:

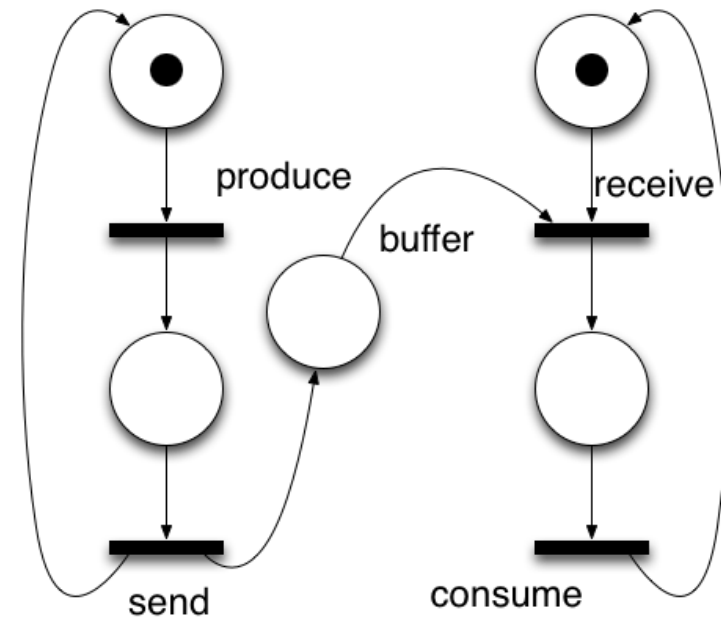
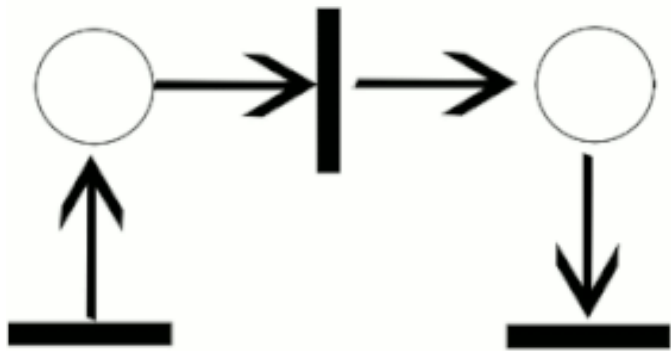
Cuando una transición se encuentra habilitada, en un instante de tiempo i absorberá tantos tokens como arcos llegan y producirá tantos tokens como arcos salen en el instante $i+\Delta$.



Red de petri - Funcionamiento

Propagación de tokens:

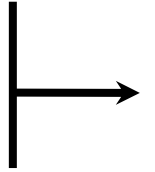
Cuando una transición se encuentra habilitada, en un instante de tiempo i absorberá tantos tokens como arcos llegan y producirá tantos tokens como arcos salen en el instante $i+\Delta$.



Red de petri - Convenciones

Convención de inicio

Para indicar que se pueden generar una cantidad ilimitada de tokens se utiliza una transición sin entradas (transición fuente). Puede haber más de una en la Red de Petri.



No bloquear la red

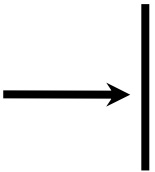
Toda transición debe tener oportunidad de ser habilitada alguna vez.

Nombres obligatorios y expresados en el diagrama

Todos los estados y transiciones deben tener nombres distintos. Además las transiciones pueden llamarse según la etapa anterior (porque termina) o siguiente (porque empieza).

Convención de fin

Una transición sin lugares de salida elimina tokens de la Red de Petri (transición final o de salida). Puede haber más de una.



Red de petri - Ejemplo

Modelar un campeonato de tenis amateur en un club de la ciudad.

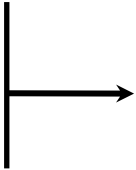
Las personas interesadas en participar llegan al club y forman una única fila esperando para abonar la inscripción al campeonato. Hay un solo cobrador que puede atender de a una persona por vez. Una vez abonado el torneo, la persona (jugador) espera por algún otro jugador para disputar su partido. Luego, ambos jugadores pasan a jugar su partido a cualquiera de las dos canchas que posee el club (en una cancha sólo se puede jugar un partido por vez). Si las dos canchas se encuentran ocupadas deben esperar a que se libere alguna de ellas para poder jugar su partido. Una vez finalizado el partido se retiran a la zona de vestuarios para finalmente retirarse del club.



LAS PERSONAS INTERESADAS EN PARTICIPAR LLEGAN AL CLUB Y FORMAN UNA ÚNICA FILA ESPERANDO PARA ABONAR LA INSCRIPCIÓN AL CAMPEONATO.

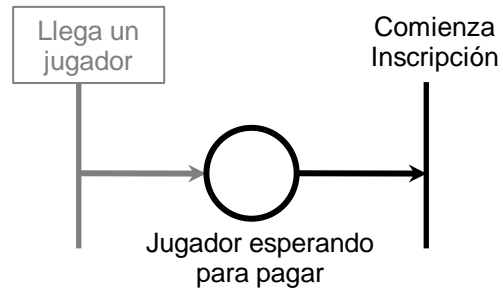
Modelamos la llegada de una cantidad arbitraria de jugadores con una transición

Llega un jugador



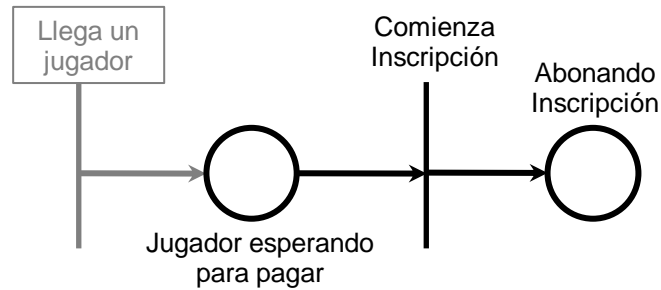
LAS PERSONAS INTERESADAS EN PARTICIPAR LLEGAN AL CLUB Y FORMAN UNA ÚNICA FILA ESPERANDO PARA ABONAR LA INSCRIPCIÓN AL CAMPEONATO.

Modelamos una cola con un sitio y una transición (por el momento)



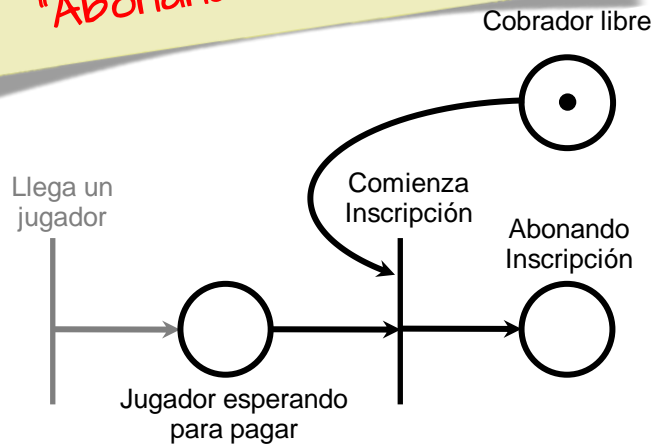
LAS PERSONAS INTERESADAS EN PARTICIPAR LLEGAN AL CLUB Y FORMAN UNA ÚNICA FILA ESPERANDO PARA ABONAR LA INSCRIPCIÓN AL CAMPEONATO.

Cuando el jugador sale de la fila se lo inscribe y se le cobra



HAY UN SOLO COBRADOR QUE PUEDE ATENDER DE A UNA PERSONA POR VEZ.

Esto impone una restricción:
No puede haber 2 jugadores
"Abonando Inscripción"



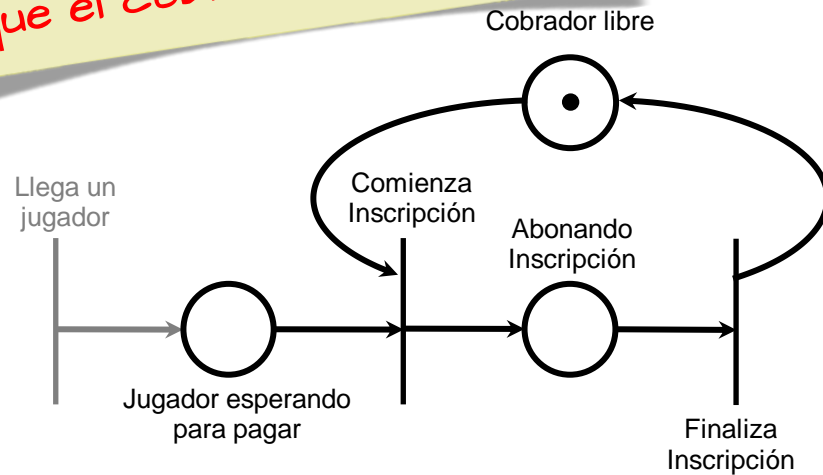
Lo modelado puede interpretarse
como: para comenzar la inscripción
del jugador se requiere **un jugador en
la fila y el cobrador libre**

Debemos impedir que la transición “Comienza Inscripción” esté habilitada para más de un jugador. Modelamos con un sitio y un token el “Cobrador libre” y lo asociamos a la transición “Comienza Inscripción”.

De esta manera si hay más de un jugador en “Jugador esperando para pagar” solo podrá pasar 1, ya que solo hay un “Cobrador libre”.

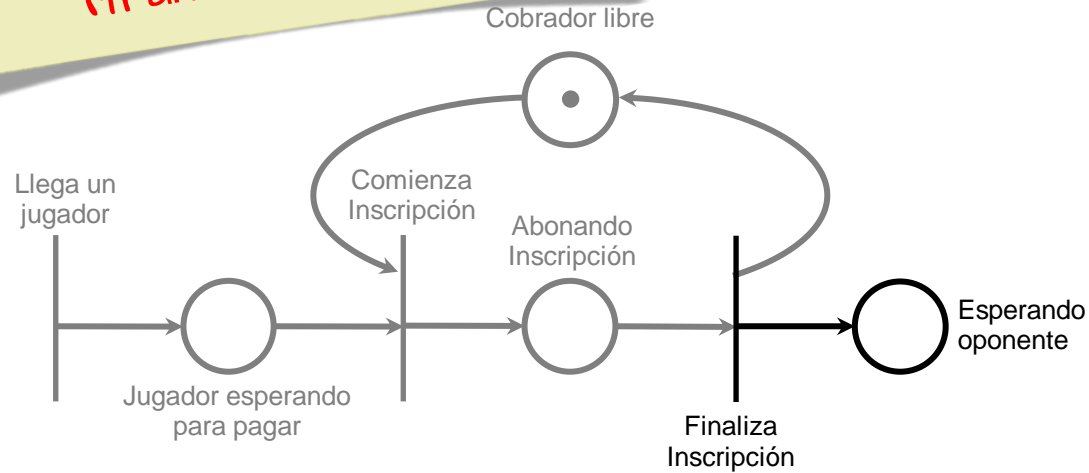
UNA VEZ ABONADO EL TORNEO, LA PERSONA (JUGADOR) ESPERA POR ALGÚN OTRO JUGADOR PARA DISPUTAR SU PARTIDO. LUEGO, AMBOS JUGADORES PASAN A JUGAR SU PARTIDO A CUALQUIERA DE LAS DOS CANCHAS QUE POSEE EL CLUB (EN UNA CANCHA SÓLO SE PUEDE JUGAR UN PARTIDO POR VEZ).

Cuando finaliza la inscripción debe modelarse que el cobrador queda libre



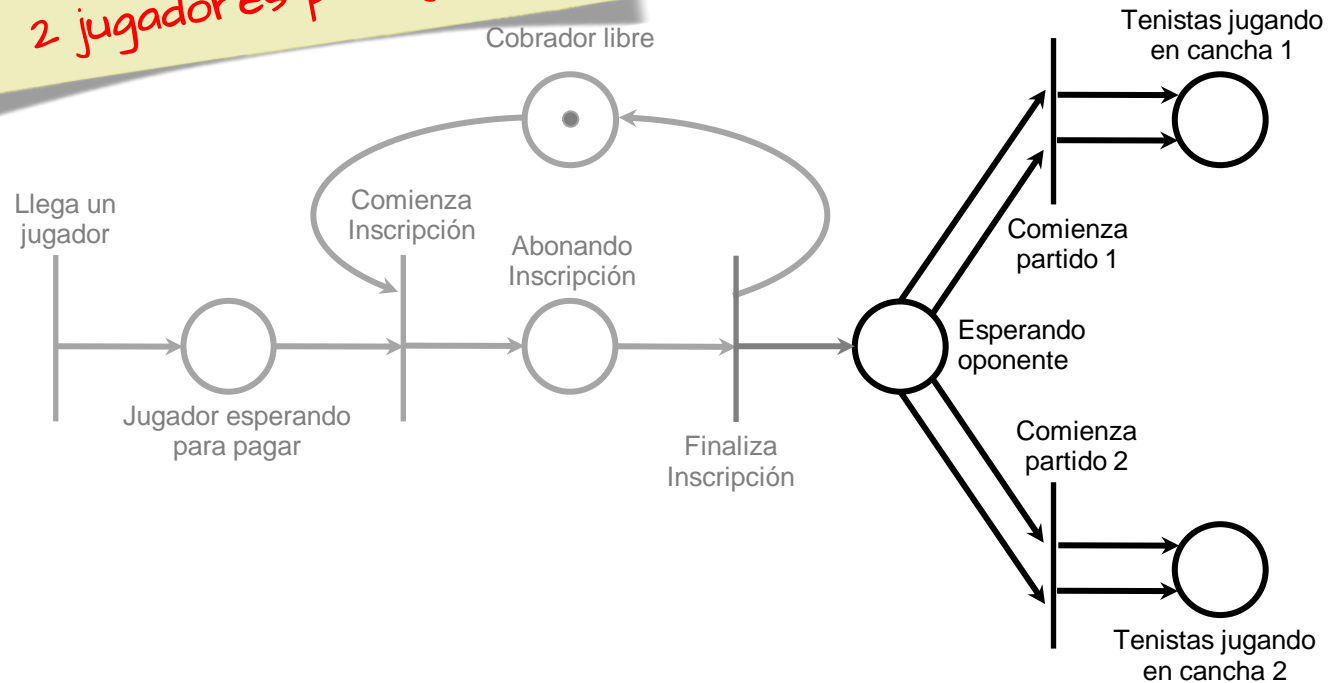
UNA VEZ ABONADO EL TORNEO, LA PERSONA (JUGADOR) ESPERA POR ALGÚN OTRO JUGADOR PARA DISPUTAR SU PARTIDO. LUEGO, AMBOS JUGADORES PASAN A JUGAR SU PARTIDO A CUALQUIERA DE LAS DOS CANCHAS QUE POSEE EL CLUB (EN UNA CANCHA SÓLO SE PUEDE JUGAR UN PARTIDO POR VEZ).

Modelamos otra cola para esperar un oponente (transición y sitio)



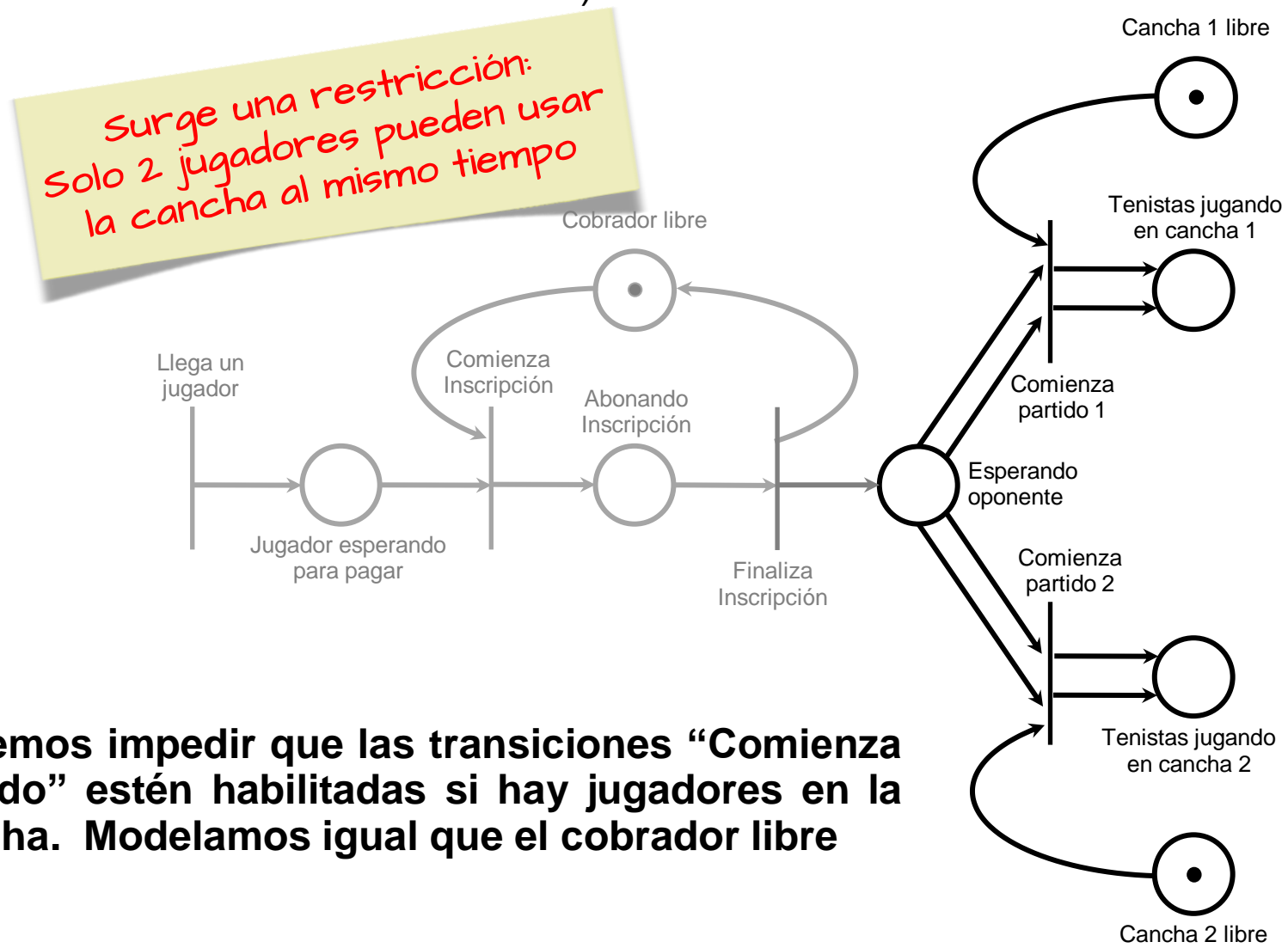
UNA VEZ ABONADO EL TORNEO, LA PERSONA (JUGADOR) ESPERA POR ALGÚN OTRO JUGADOR PARA DISPUTAR SU PARTIDO. LUEGO, AMBOS JUGADORES PASAN A JUGAR SU PARTIDO A CUALQUIERA DE LAS DOS CANCHAS QUE POSEE EL CLUB (EN UNA CANCHA SÓLO SE PUEDE JUGAR UN PARTIDO POR VEZ).

Debemos modelar el uso de 2 canchas y la sincronización de 2 jugadores para jugar



Si hay 2 jugadores “Esperando Oponente” en la cola, deben dirigirse JUNTOS (sincronizados) a una de las canchas. Luego pasan juntos a jugar.

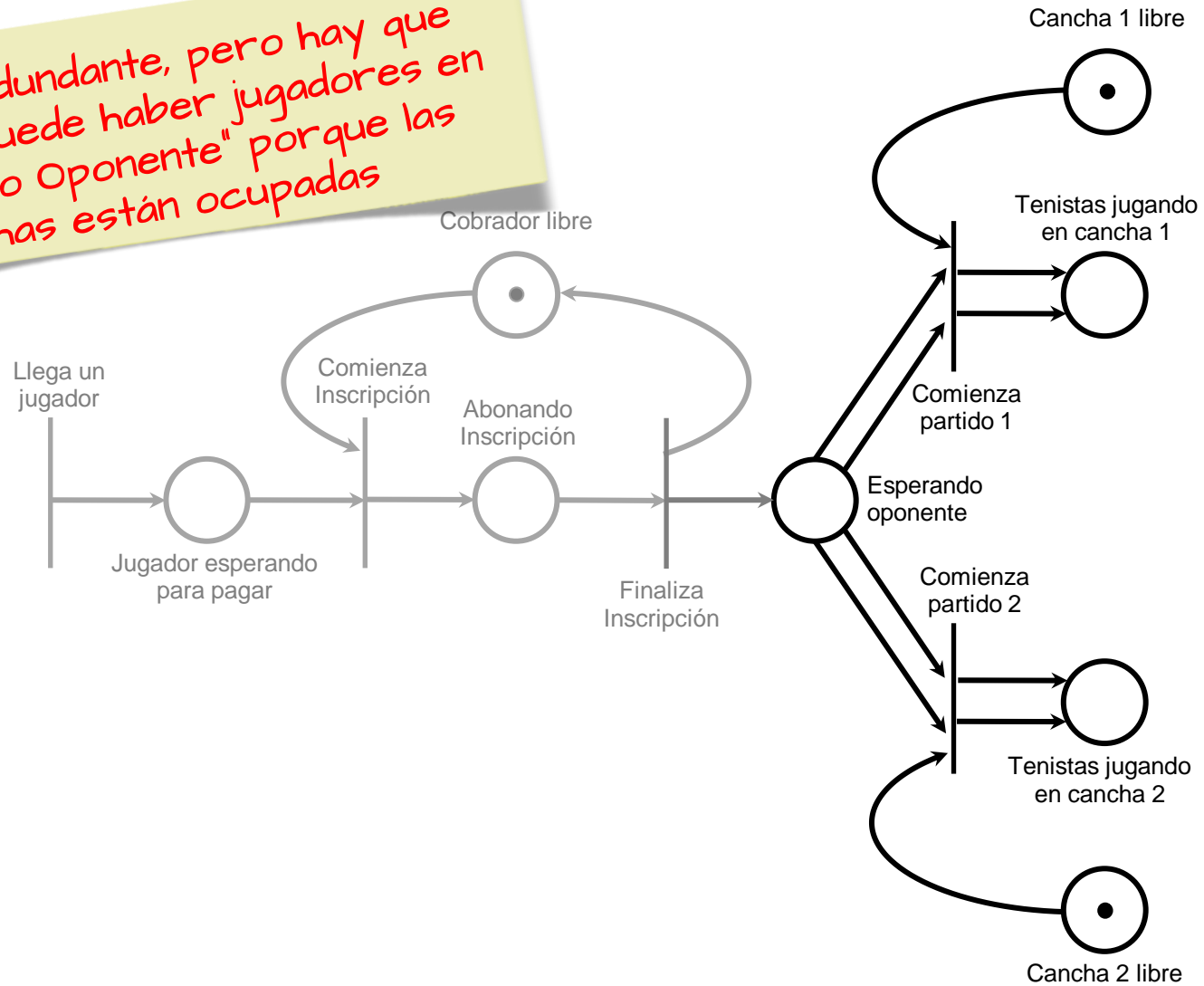
UNA VEZ ABONADO EL TORNEO, LA PERSONA (JUGADOR) ESPERA POR ALGÚN OTRO JUGADOR PARA DISPUTAR SU PARTIDO. LUEGO, AMBOS JUGADORES PASAN A JUGAR SU PARTIDO A CUALQUIERA DE LAS DOS CANCHAS QUE POSEE EL CLUB (EN UNA CANCHA SÓLO SE PUEDE JUGAR UN PARTIDO POR VEZ).



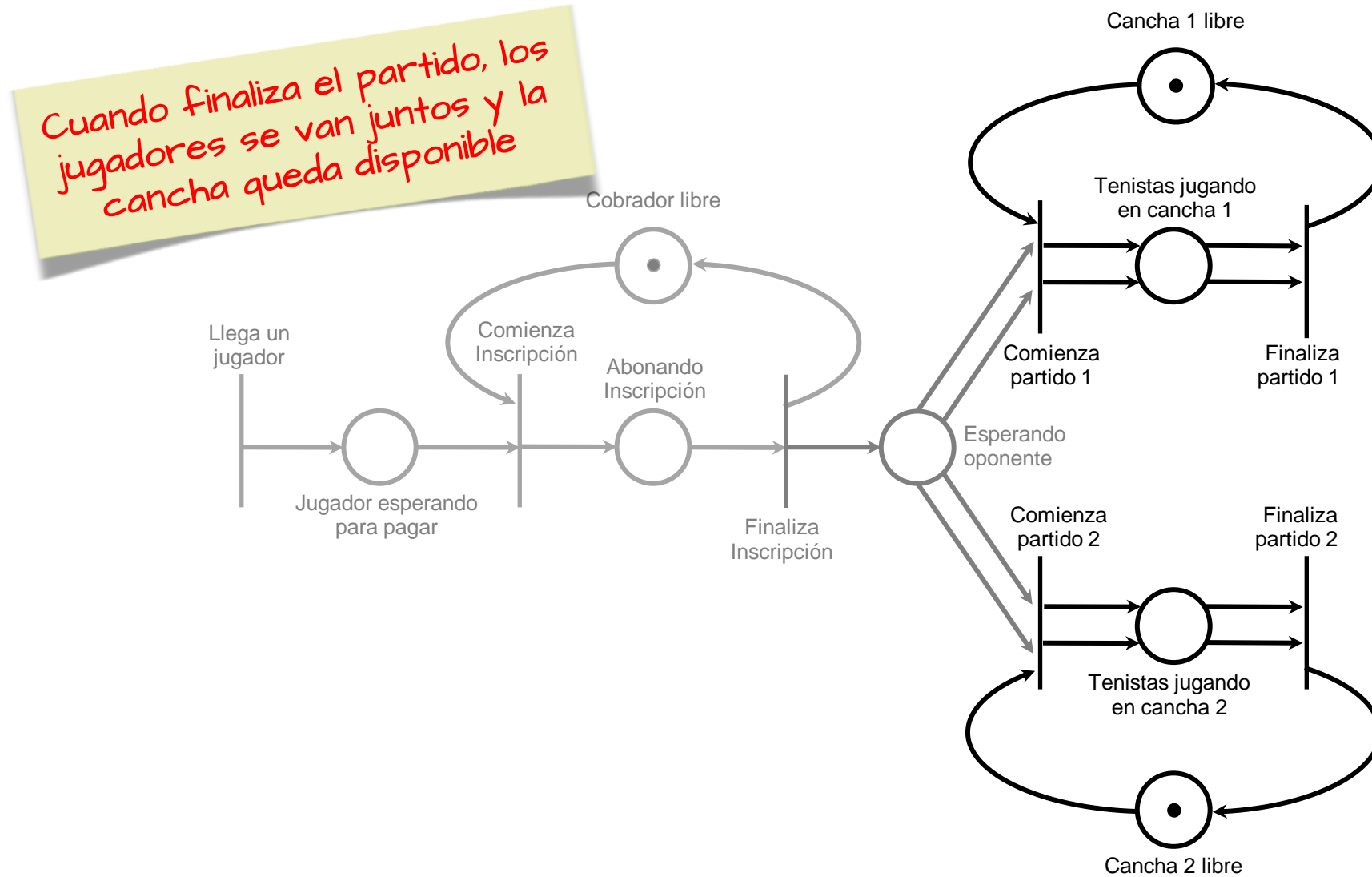
Debemos impedir que las transiciones “Comienza partido” estén habilitadas si hay jugadores en la cancha. Modelamos igual que el cobrador libre

SI LAS DOS CANCHAS SE ENCUENTRAN OCUPADAS DEBEN ESPERAR A QUE SE LIBERE ALGUNA DE ELLAS PARA PODER JUGAR SU PARTIDO. UNA VEZ FINALIZADO EL PARTIDO SE RETIRAN A LA ZONA DE VESTUARIOS PARA FINALMENTE RETIRARSE DEL CLUB.

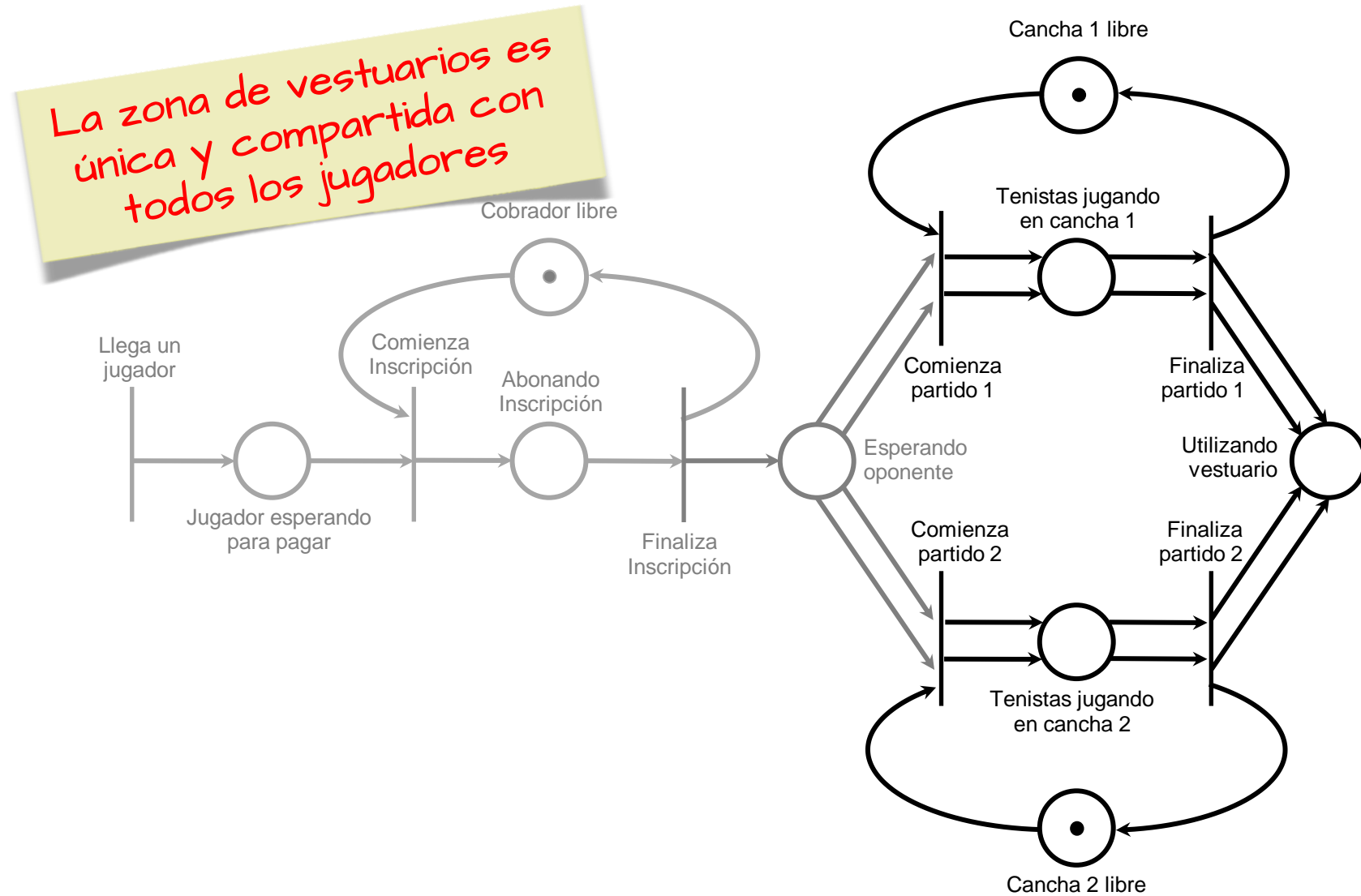
Esto es redundante, pero hay que notar que puede haber jugadores en "Esperando Oponente" porque las canchas están ocupadas



SI LAS DOS CANCHAS SE ENCUENTRAN OCUPADAS DEBEN ESPERAR A QUE SE LIBERE ALGUNA DE ELLAS PARA PODER JUGAR SU PARTIDO. UNA VEZ FINALIZADO EL PARTIDO SE RETIRAN A LA ZONA DE VESTUARIOS PARA FINALMENTE RETIRARSE DEL CLUB.

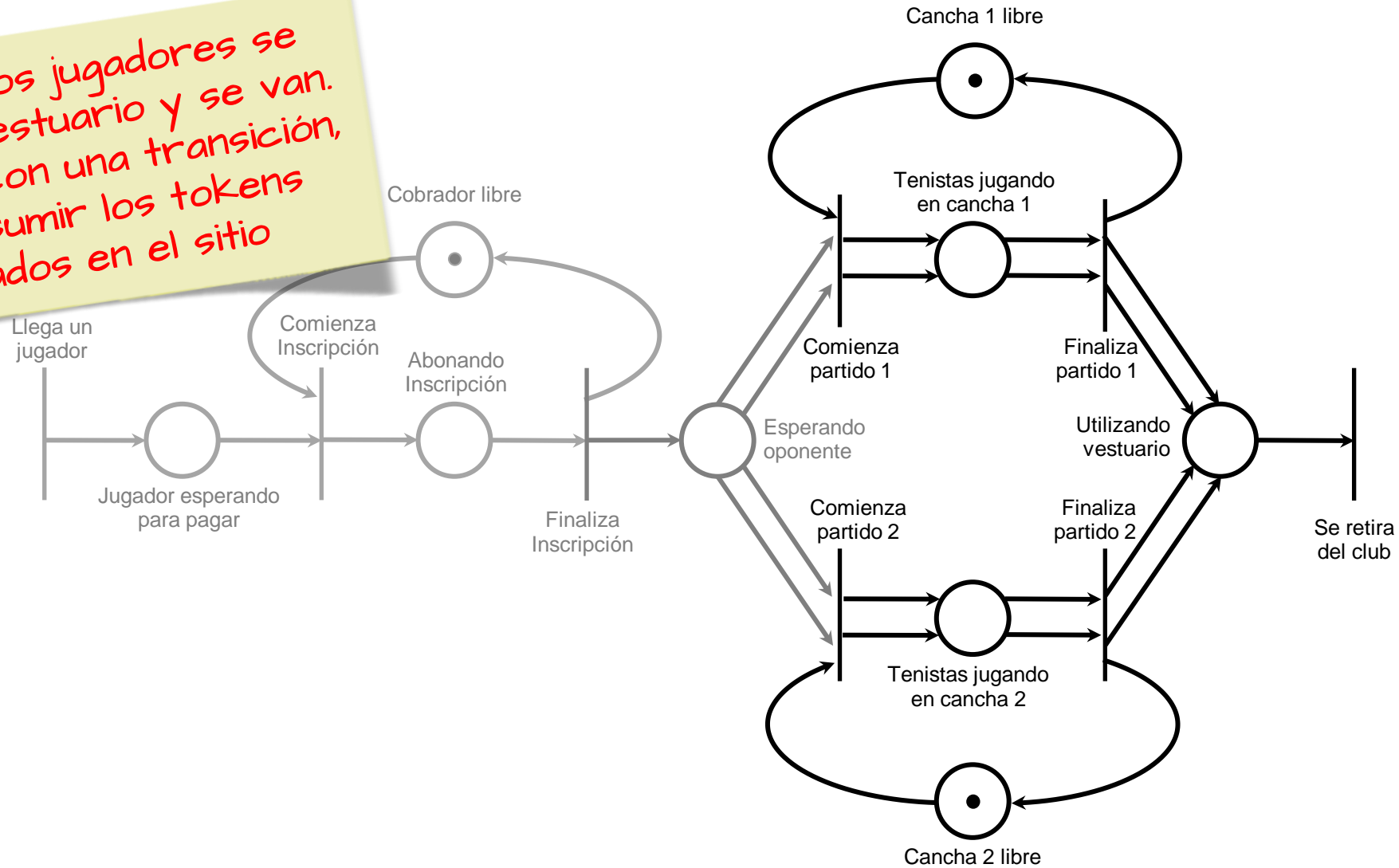


SI LAS DOS CANCHAS SE ENCUENTRAN OCUPADAS DEBEN ESPERAR A QUE SE LIBERE ALGUNA DE ELLAS PARA PODER JUGAR SU PARTIDO. UNA VEZ FINALIZADO EL PARTIDO SE RETIRAN A LA ZONA DE VESTUARIOS PARA FINALMENTE RETIRARSE DEL CLUB.



SI LAS DOS CANCHAS SE ENCUENTRAN OCUPADAS DEBEN ESPERAR A QUE SE LIBERE ALGUNA DE ELLAS PARA PODER JUGAR SU PARTIDO. UNA VEZ FINALIZADO EL PARTIDO SE RETIRAN A LA ZONA DE VESTUARIOS PARA FINALMENTE RETIRARSE DEL CLUB.

Finalmente los jugadores se retiran del vestuario y se van. Modelamos con una transición, para consumir los tokens acumulados en el sitio

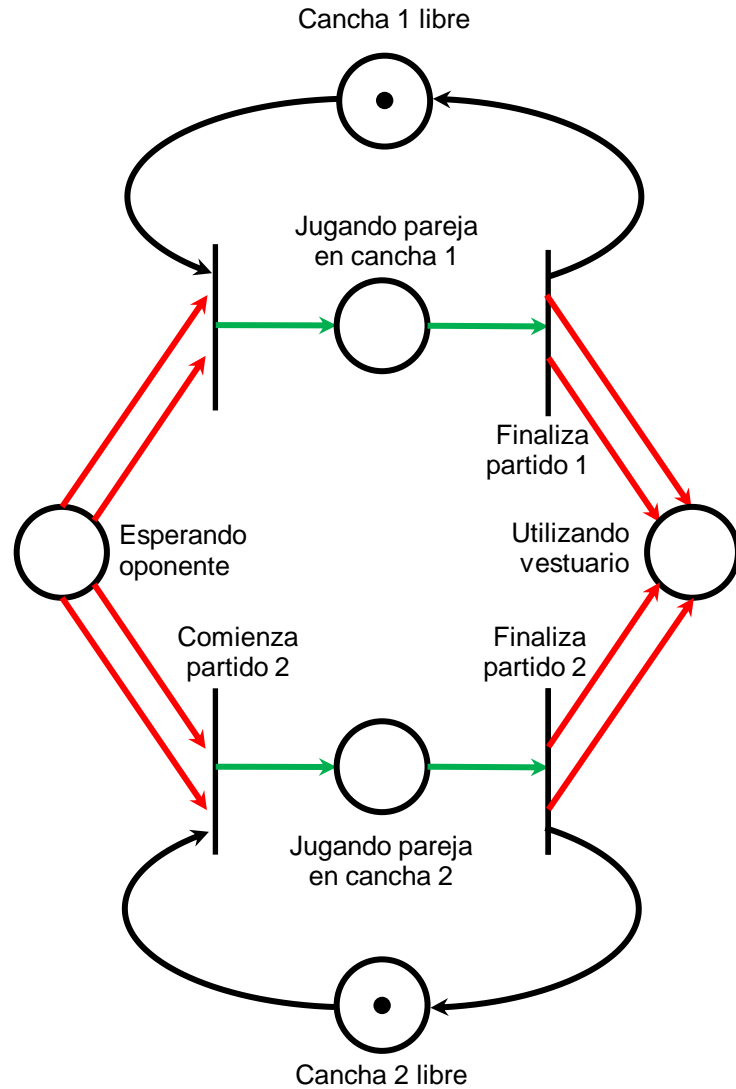


VARIANTE

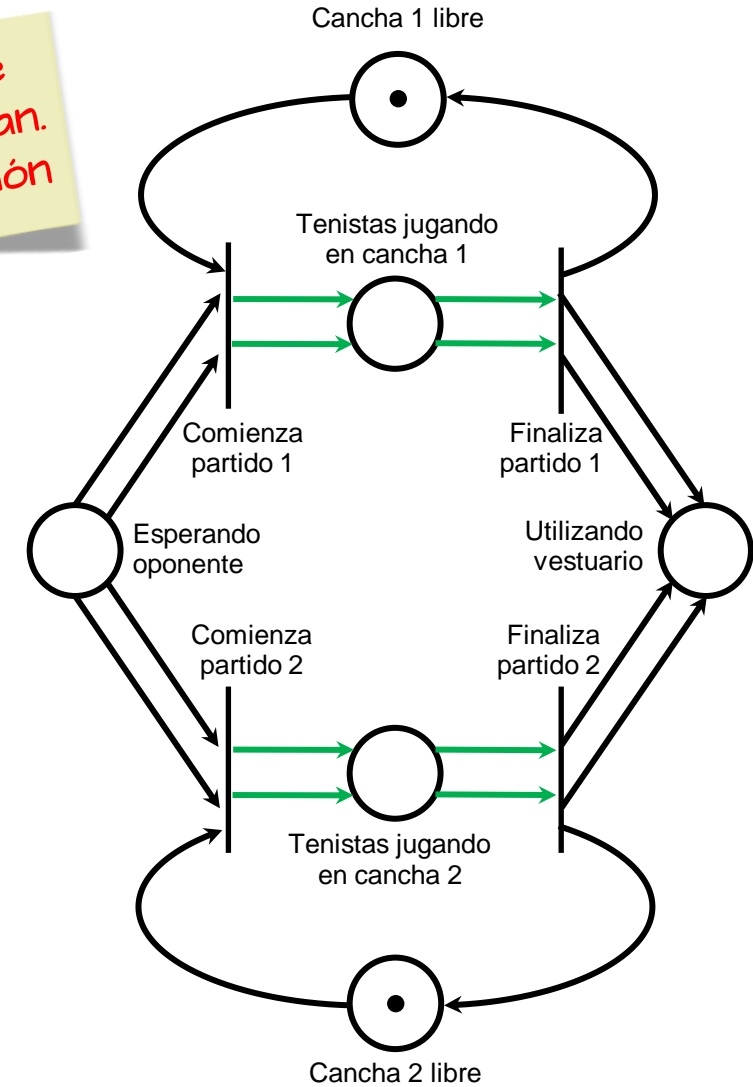
- Podría haber una variante respecto del modelado de los tenistas jugando en las canchas.
- En vez de modelar dos tenistas jugando podría modelarse una pareja jugando.
- De ser así, el bloque de modelado de la cancha tendría un arco en vez de dos.
- Es importante remarcar que si se usa esta aproximación, cuando se retiran los jugadores del vestuario no lo hacen en pareja. Deben modelarse con dos arcos.
- Dependiendo del problema, esta forma de modelar puede ser alternativa (este caso) u obligatoria (ej: embasar 6 botellas en un pack).



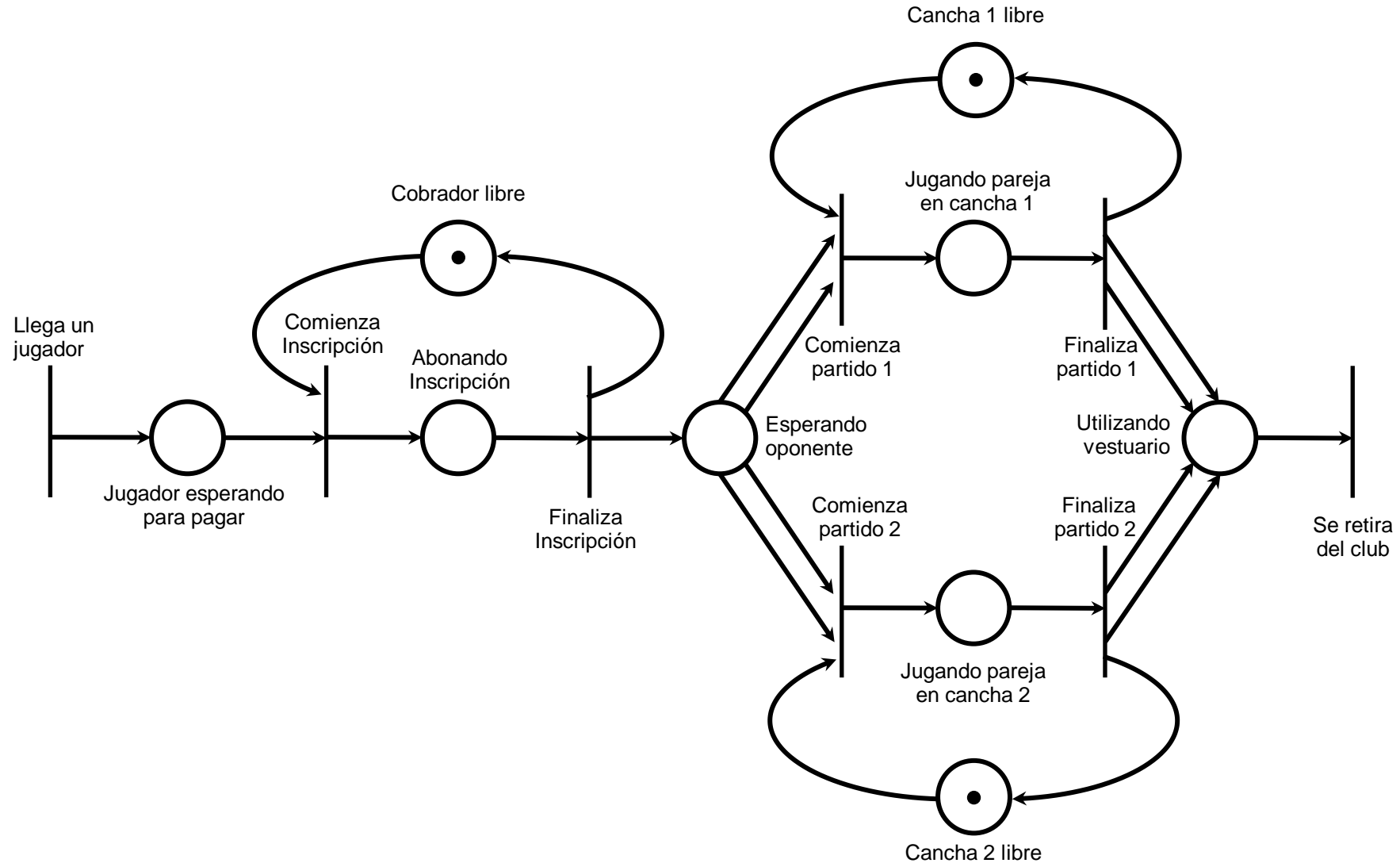
VARIANTE – Tenistas como pareja



Finalmente los jugadores se retiran del vestuario y se van. Modelamos con una transición



VARIANTE – Tenistas como pareja



EJEMPLO DESARROLLADO - SOLUCIÓN

Solución Final

