



Trabajo Práctico N°1

75.59 - Técnicas de Programación Concurrentes I

Facultad de Ingeniería de la Universidad de Buenos Aires

2do. Cuatrimestre 2017

Federico Baliña, *Padrón Nro. 96.945*

`federicobalina@gmail.com`

Pablo Rodrigo Ciruzzi, *Padrón Nro. 95.748*

`p.ciruzzi@hotmail.com`

Tomás Mussi, *Padrón Nro. 94.485*

`tomasmussi@gmail.com`

12 de octubre de 2017

Índice

1. Hipótesis	3
2. División en Procesos	3
2.1. Main	3
2.2. Beach Manager	3
2.3. Team Maker	3
2.4. Court Manager	4
2.5. Results Reporter	4
2.6. Match Processes	4
2.7. Timer	4
3. Esquema de Comunicación	4
3.1. Main	4
3.2. Main - Beach Manager	4
3.3. Main - Team Maker	4
3.4. Beach Manager - Team Maker	5
3.5. Team Maker - Court Manager	5
3.6. Court Manager	5
3.7. Court Manager - Results Reporter	5
3.8. Court Manager - Team Maker	5
3.9. Match Process - Court Manager	5
3.10. Timer - Beach Manager	5
3.11. Timer - Court Manager	5
4. Mecanismos de concurrencia	6
4.1. Señales	6
4.2. Locks	6
4.3. Memoria Compartida	6
4.4. Pipes y Fifos	7
4.5. Semáforos	7
5. Diagramas	8
5.1. Diagrama de procesos e IPC	8
5.2. Diagramas de clases	8
5.3. Diagrama de transición de estados de un jugador	10
6. Instrucciones	10

1. Hipótesis

- Un jugador puede abandonar la playa voluntariamente cuando ingresa a ella, ya sea porque recién ingresa o porque volvió de un partido. El jugador decidirá aleatoriamente con una probabilidad de irse fijada.
- Un jugador que ya haya jugado sus k partidos, se retira y no vuelve a entrar jamás.
- Los partidos pueden durar un tiempo aleatorio entre 1 y 5 segundos.
- Cuando una cancha se inunda, el partido se suspende y es como si nunca se hubiese jugado. De todas formas, ambas parejas quedan esperando para volver a jugar, aunque no necesariamente jueguen entre sí.
- Si una vez comenzado el torneo, en algún momento del mismo, no comienzan partidos en más de 20 segundos, se da por finalizado el mismo.

2. División en Procesos

Para modelar el torneo de manera que cumpla con las funcionalidades requeridas, se optó por asignar un proceso a cada actor/rol que debería ejercer una función independiente y en simultáneo a los demás.

2.1. Main

Es el punto de entrada del proyecto. Recibe los parámetros de configuración: dimensiones de la playa (cantidad de canchas), capacidad máxima y cantidad máxima de partidos por jugador. Además, recibe el nivel de log, que por default es en modo INFO.

Su rol básicamente consiste en la creación de los procesos principales (todos menos los Match Processes) y las facilidades de IPC requeridas. Además, es el encargado de recibir a las personas que llegan al predio, mediante el New Player Handler.

2.2. Beach Manager

Si pensáramos a este proceso como una persona, sería el que está parado en la puerta del predio y va anotando a las personas a medida que van llegando. Por esto, también es el encargado de dar comienzo al torneo una vez que llegaron al menos 10 personas. Básicamente, es una suerte de “patovica”.

También, dado que es él quien permite el ingreso de las personas, debe estar atento a que no se sobrepase la capacidad máxima.

2.3. Team Maker

Una vez que un jugador ya ingresó en el predio y se encuentra compitiendo en el torneo, el Team Maker es el encargado de armar las parejas teniendo en cuenta que las mismas no se pueden repetir durante el torneo. Es decir, que cada participante puede jugar en pareja con cada uno de los participantes restantes como máximo una sola vez.

Una persona que ingresa a la playa (Recién entrado o porque vuelve de un partido) puede decidir retirarse voluntariamente. El Team Maker deberá estar atento a esto para que el Beach Manager pueda dejar ingresar jugadores acorde a la capacidad máxima.

2.4. Court Manager

Como su nombre lo indica, es el encargado de las canchas. Dado que se posee un número acotado de canchas, este proceso es el encargado de definir qué parejas jugarán una contra la otra y en qué cancha.

Este proceso será el que cree los Match Process correspondientes a los partidos despachados.

2.5. Results Reporter

Es el encargado de llevar un detalle de todos los partidos jugados en el torneo y de llevar la tabla de posiciones actualizada e informarla periódicamente. Además, una vez finalizado el torneo, es el encargado de informar quién fue el campeón.

2.6. Match Processes

Cada Match Process durante su ciclo de vida representa un partido que se está jugando en el torneo. Es el encargado de simular dicho partido con un tiempo aleatorio, así como definir el resultado del mismo.

2.7. Timer

El Timer es el encargado de tomar el tiempo de cuánto tiempo pasó desde que comenzó el último partido. En caso de excederse de cierto tiempo, éste dará por finalizado el torneo.

3. Esquema de Comunicación

3.1. Main

El Main debe poder ser advertido del arribo de un jugador al torneo.

3.2. Main - Beach Manager

El Main debe poder informarle al Beach Manager cuando ingresa una persona al predio.

3.3. Main - Team Maker

Dado que las personas que se van voluntariamente pueden volver a entrar, el Team Maker debe compartir la información de quiénes se fueron con el Main para que puedan eventualmente volver a entrar al torneo.

3.4. Beach Manager - Team Maker

El Beach Manager debe ir informándole al Team Maker las personas que él ya anotó (que ingresaron en el predio) y que van a empezar a jugar el torneo.

Ambos deben compartir la información de cuántos jugadores se encuentran jugando en el torneo, dado que hay una capacidad máxima que no se puede sobrepasar.

3.5. Team Maker - Court Manager

El Team Maker debe informarle al Court Manager las parejas que armó para que luego este último pueda organizar los partidos.

3.6. Court Manager

El Court Manager debe poder ser advertido de cambios en la marea (ya sea subida o bajada), ya que en dicho caso debe cancelar partidos o saber que tiene más canchas disponibles.

3.7. Court Manager - Results Reporter

El Court Manager debe poder informarle al Results Reporter los partidos que se juegan y sus resultados, para que este último lleve la tabla de posiciones.

3.8. Court Manager - Team Maker

Así como el Beach Manager le debía informar al Team Maker las personas que comenzarían a jugar en el torneo, cuando termina un partido, el Court Manager es el encargado de avisarle al Team Maker que estas personas ya están disponibles para conseguirles otra pareja.

3.9. Match Process - Court Manager

Cada Match Process debe poder avisarle al Court Manager que su partido ha finalizado y su resultado, para que este último sepa que hay una nueva cancha disponible.

3.10. Timer - Beach Manager

Dado que el Timer debe comenzar a correr una vez que empezó el torneo, el Beach Manager debe indicarle de alguna manera que esto sucedió.

3.11. Timer - Court Manager

Como el Timer se fija hace cuanto tiempo no comienza un partido, es importante que el Court Manager pueda avisarle cuando esto sucedió, para que el Timer vuelva a comenzar.

4. Mecanismos de concurrencia

4.1. Señales

Se utilizaron las siguientes señales para señalar eventos asincrónicos:

- SIGINT (2): Para terminar la ejecución del programa, se le envía esta señal al Main, el cual a su vez se la envía a sus procesos hijos. Al recibirla, Court Manager también se la envía a sus Match Process hijos en curso. Cada proceso la maneja para terminar de manera correcta.
- SIGUSR1 (10): Esta señal se le envía al Main para señalar el arribo de un jugador (puede ser un jugador nuevo o uno que se haya retirado voluntariamente antes). Por otro lado, también la utiliza el Court Manager para avisarle a los Match Process correspondientes que deben terminar debido a una inundación de la cancha.
- SIGCHLD (17): Esta señal se utiliza como mecanismo de comunicación entre un Match Process y el Court Manager. Al finalizar un Match Process, este realiza un `exit()` con un código asociado al resultado de dicho partido. Al finalizar dicho proceso, Court Manager (el padre) recibe esta señal, obtiene el código correspondiente y actúa en consecuencia.
- SIGUSR2 (12) / SIGUNUSED (31): Se utilizan respectivamente para señalar al Court Manager sobre la subida y bajada de la marea. Este se encargará de manejarla como corresponda.
- SIGUSR2 (12): Esta señal la utiliza el Court Manager para avisarle al Timer que se ha despachado un nuevo partido. De esta manera, el Timer empieza a contar de nuevo.

4.2. Locks

Los locks fueron utilizados en dos situaciones:

- Para administrar la lectura y escritura de memoria compartida.
- Para administrar las escrituras del Logger. Dado que todos los procesos utilizan el Logger para escribir sobre un mismo archivo, era necesario un mecanismo para controlar estos accesos.

Cabe destacar que en vez de Locks se podrían haber usado semáforos binarios que son equivalentes. Pero se optó por los locks dado que su uso es más natural para estos casos y no tienen el problema de la inicialización que sí tienen los semáforos, además de que esta funcionalidad ya se había desarrollado al momento de conocer los semáforos.

4.3. Memoria Compartida

Se utilizó una única memoria compartida entre el Main (Más precisamente para el New Player Handler) y el Team Maker para manejar los jugadores que abandonaron el predio voluntariamente. El Team Maker escribe en ella los jugadores que abandonan el predio voluntariamente y el Main lee de ella para reingresarlos.

Se podría pensar que esto se podría haber realizado mediante un fifo, pero en un fifo si no hubiera habido nada para leer, el Main habría quedado bloqueado hasta que el Team Maker escribiera y esto no era deseable para este caso (Ya que en el caso de que no haya jugadores que se hayan ido, debe ingresar uno completamente nuevo).

4.4. Pipes y Fifos

En este trabajo práctico se utilizaron Fifos, no Pipes. Esto se debe a que se considera que los Fifos son una evolución de los Pipes, por lo que nos dan las mismas prestaciones y un poco más.

Se utilizaron los siguientes Fifos:

- Main - Beach Manager: canal de comunicación en el cual el Main escribe enteros (int) representando id's de jugadores que están entrando al predio y el Beach Manager los lee.
- Beach Manager y Court Manager - Team Maker: canal de comunicación en el cual el Main escribe personas (Person) representando a jugadores que ya entraron al predio y quieren empezar a jugar; mientras que el Court Manager también escribe en el mismo a las personas que terminaron de jugar un partido. Team Maker las lee para generar parejas.
- Team Maker - Court Manager: canal de comunicación en el cual el Team Maker escribe equipos (Team) representando las parejas que va armando y el Court Manager los lee. Asimismo, el Court Manager también lo utiliza para escribir equipos que hayan terminado su partido por una inundación, de forma que vuelvan a estar esperando para jugar sin pasar por el Team Maker.
- Court Manager - Results Reporter: canal de comunicación en el cual el Court Manager escribe partidos (Match) representando los partidos que van terminando y el Results Reporter los lee.

4.5. Semáforos

Los semáforos son contadores que representan la cantidad de recursos disponibles. En nuestro caso utilizamos los siguientes:

- Beach Manager - Team Maker: se utiliza un semáforo que representa el cupo disponible de personas para que entre al torneo. Se inicializa con la capacidad máxima y Beach Manager lo disminuye cuando va a ingresar un jugador y Team Maker lo aumenta cuando se retira uno. De esta manera, podemos estar seguros que no se va a superar la capacidad máxima de la playa.
- Court Manager: se utiliza un semáforo que representa la cantidad de canchas disponibles. Se inicializa como la cantidad de filas por columnas y él mismo irá decrementándolo a medida que se crean partidos. A su vez, este semáforo se aumenta cuando un proceso termina un partido (Se recibe la señal de finalización) y se libera la cancha. Aunque esté todo dentro del mismo proceso, se utilizó un semáforo para que el proceso quede bloqueado hasta que haya alguna cancha libre.

- Timer - Beach Manager: se utiliza para mostrar que el torneo ha comenzado (Ya que el timer debe comenzar a correr recién en ese momento). Inicialmente se encuentra en 0, hasta que el Beach Manager lo aumenta indicando que comenzó el torneo, para que el Timer lo decremente y empiece a contar el tiempo. Este semáforo se utiliza una sola vez y, al ser binario, podría haber sido reemplazado por un lock.

5. Diagramas

5.1. Diagrama de procesos e IPC

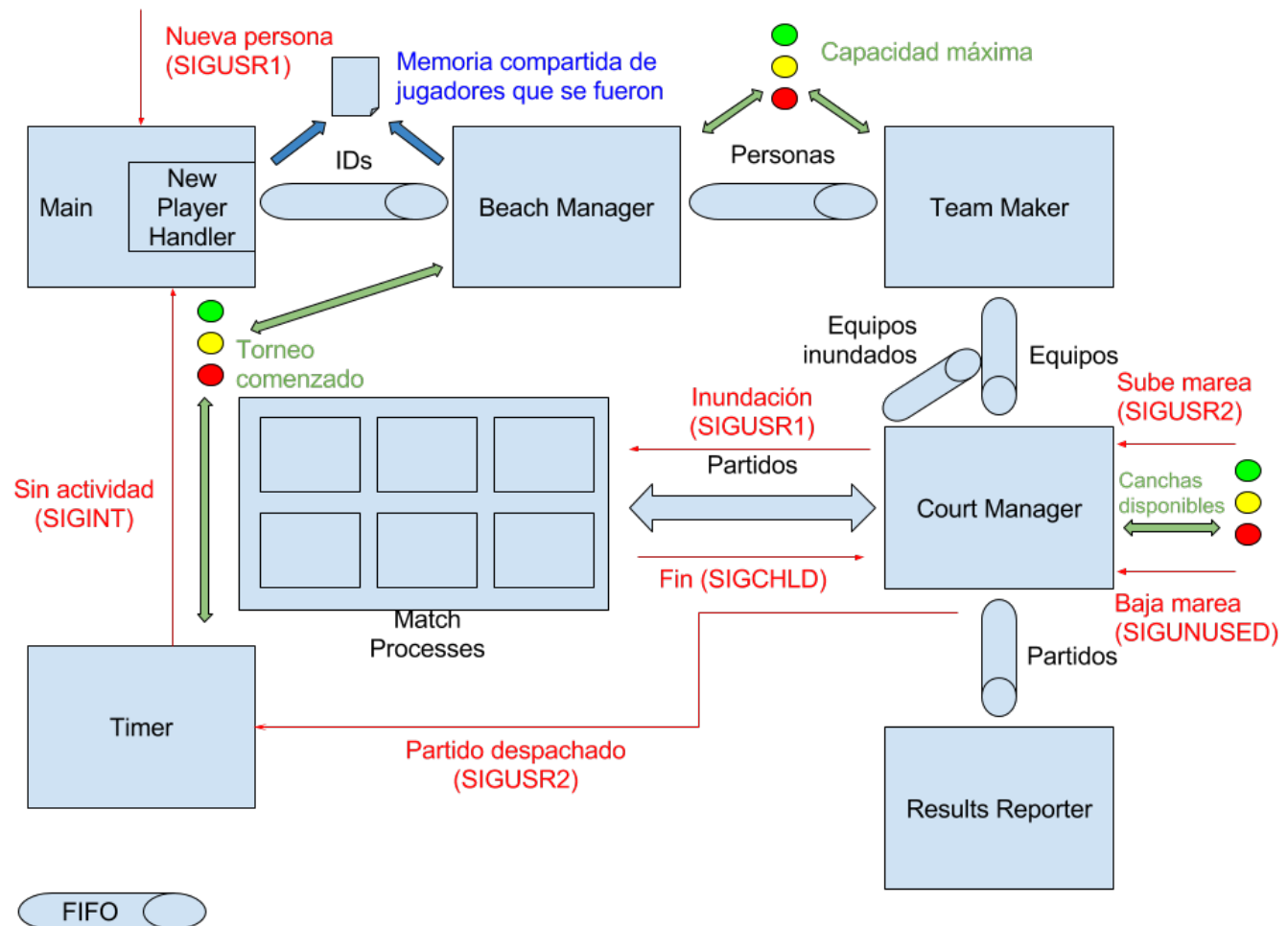


Figura 1: Diagrama de procesos e IPC

5.2. Diagramas de clases

A continuación mostramos el diagrama de clases del programa. Se obviaron los IPCs, ya que no aportaban claridad, sino que es más simple verlos en el diagrama de procesos e IPC. Tampoco se muestran las clases *Person*, *Team* y *Match*, ya que son simples contenedores.

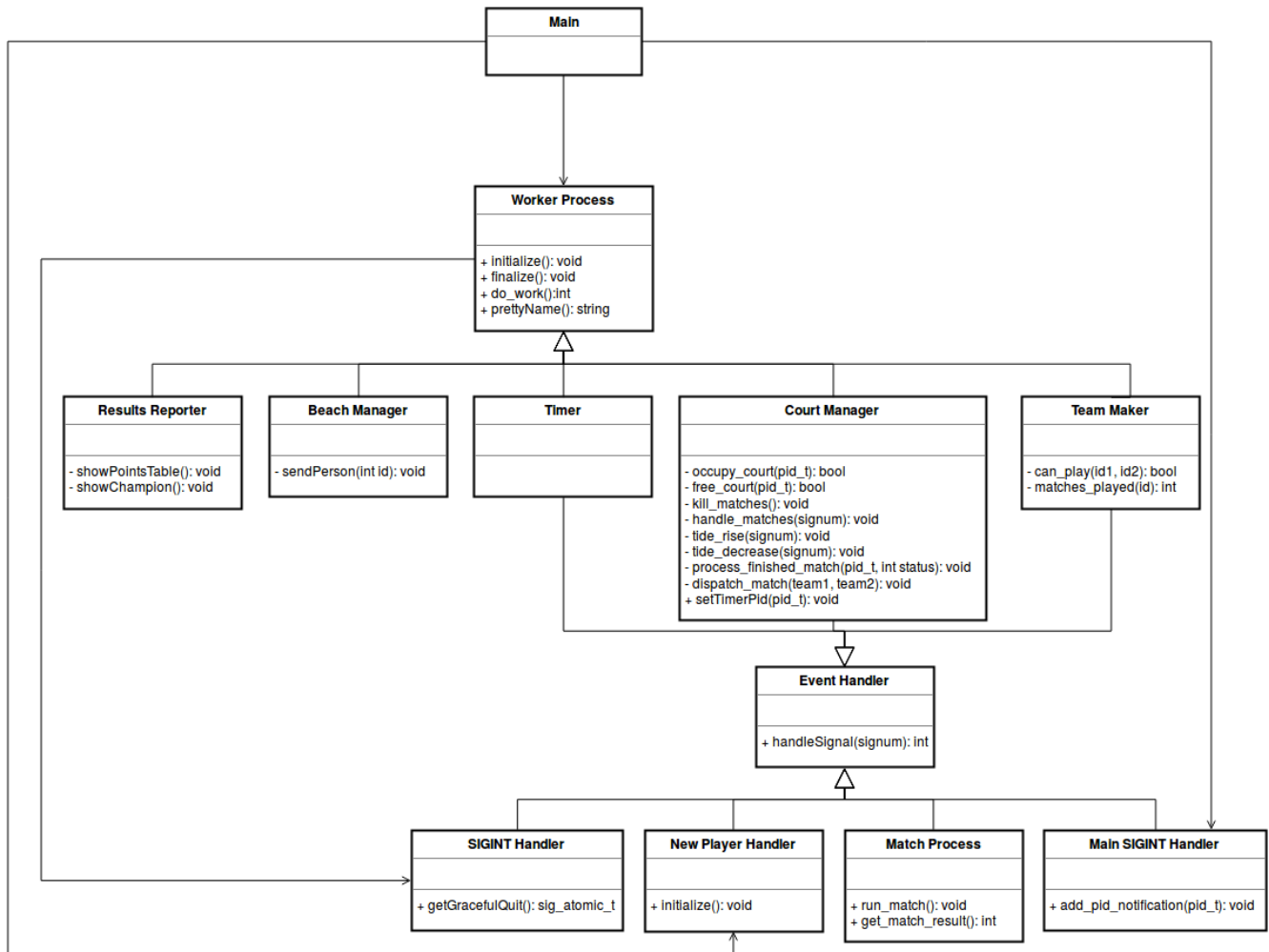


Figura 2: Diagrama de clases del TP

5.3. Diagrama de transición de estados de un jugador

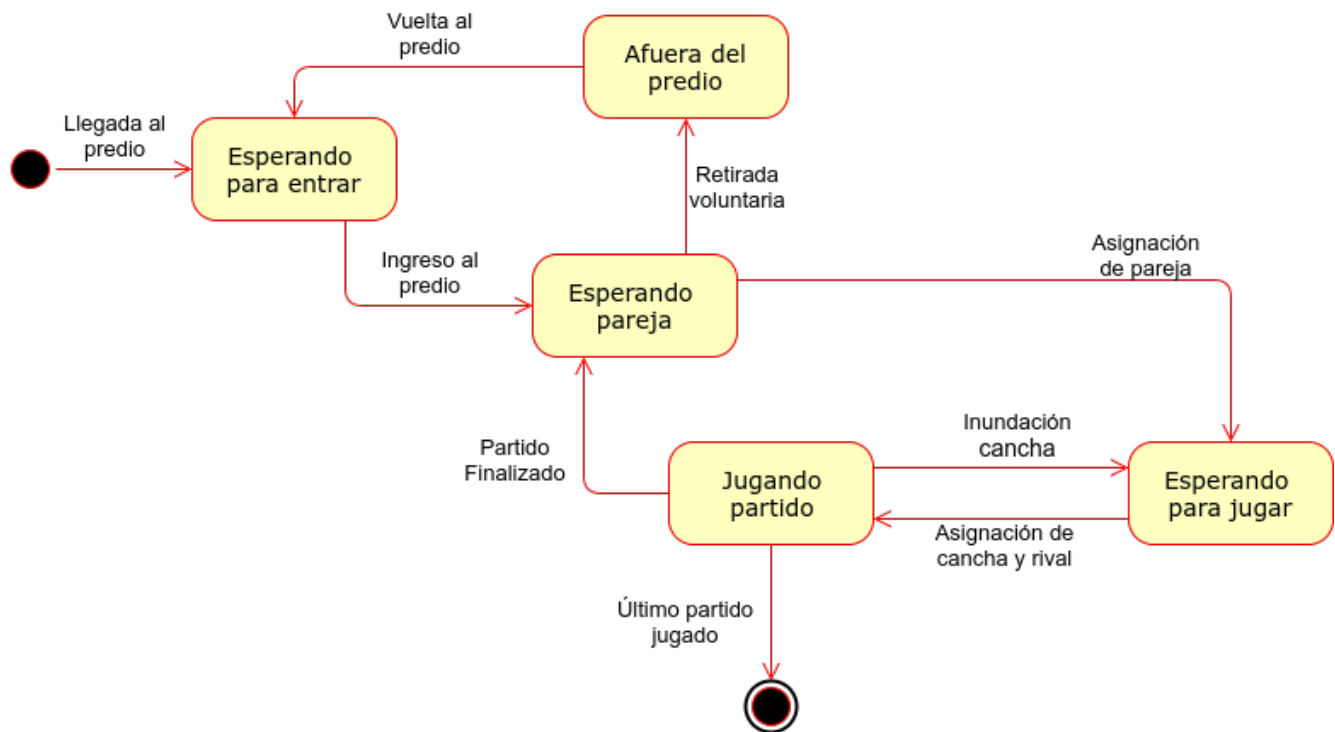


Figura 3: Diagrama de transición de estados de un jugador

6. Instrucciones

En el código fuente del programa se puede encontrar un archivo README.md, en donde se explicita cómo correr la simulación.