

75.59-TpSemaforos

Nicolás Longo - Padrón N 98271

17 de junio 2020

1 Ejercicio 1

1.1 Introducción: análisis y explicación del problema

Este trabajo aborda la preparación de una pizza en 4 pasos secuenciales.

- Preparar la masa
- Preparar los ingredientes que lleve
- Rallar el queso
- Hornear

El proceso en sí no tiene nada de especial, hasta que planteamos lo siguiente: qué pasa si cada una de estas tareas es realizada por una entidad y, si bien el proceso de preparación de una pizza resulta ser secuencial, podemos abordar concurrentemente el proceso de preparación de varias pizzas a la vez.

Para llevar adelante esta tarea, contamos con una serie de entidades trabajadoras y un horno, en el que podemos cocinar hasta 5 pizzas en simultáneo. Este último punto es crucial, pues se transforma en nuestro cuello de botella: no importa cuán rápido nuestra Fábrica de Pizzas prepare la masa, o los ingredientes, o ralle el queso, si el horno tarda más tiempo este será el tiempo mínimo que tardará una pizza en ser preparada.

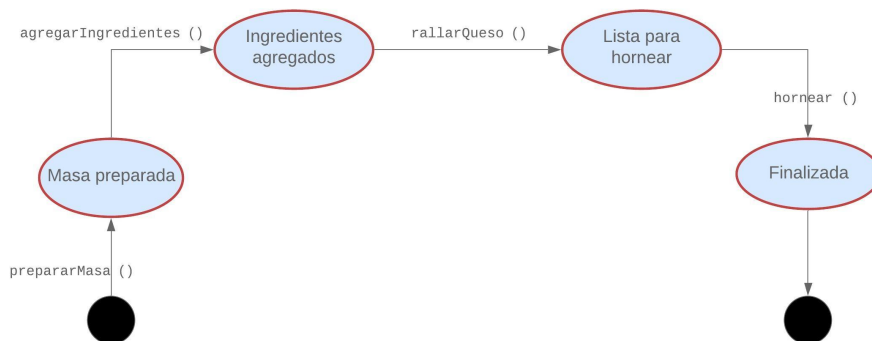
Desde ese punto de vista, podemos modelar la preparación de pizzas en ésta Fábrica como un problema de Productor(es) y Consumidor: nuestras entidades trabajadoras van realizando (de manera secuencial) la preparación de una pizza hasta que llega el momento de meterla en el horno, si es que hay para entonces un espacio disponible.

1.2 Implementación

Una Fábrica prepara en un día de trabajo una determinada cantidad de Pizzas. Para ello cuenta con un Fabricante de Masa, una Herramienta de Ingredientes, un Rallador de Queso y el Horno que se menciona en la sección anterior. El Fabricante de Masa y la Herramienta de Ingredientes realizan su trabajo de manera ininterrumpida: estos preparan cada uno su parte de la Pizza y van

dejándola a disposición para que el pipeline del producto pueda continuar. Distinto es el caso del Rallador de Queso, quien se encarga de poner la pizza en el horno (luego de agregarle el queso, encima de los ingredientes especiales) y sólo tiene en lista de espera, como máximo, una pizza con el queso rallado lista para meter al horno. Cuando la pizza está horneada, es el Horno mismo el que se encarga de sacarla, momento a partir del cual la Pizza está finalizada.

A continuación, se adjunta un diagrama de los estados por los que atraviesa la Pizza desde que arranca el proceso hasta que ya está lista



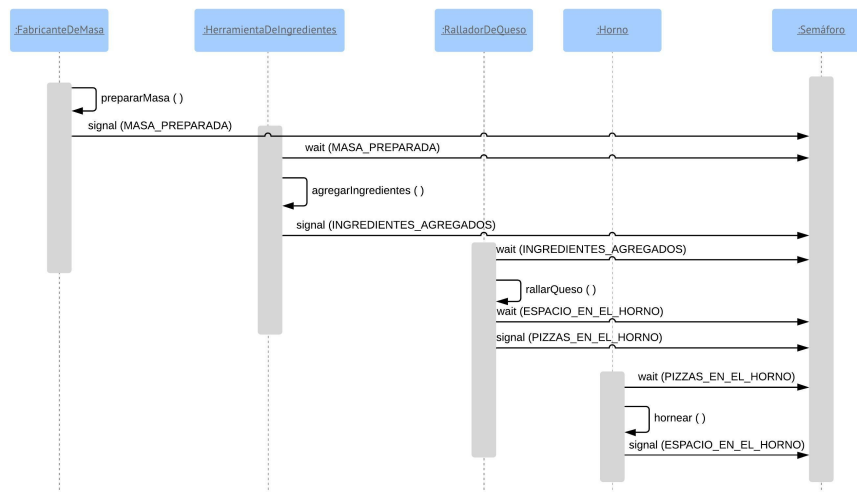
Nuestros procesos utilizan como herramienta de comunicación (IPC) un semáforo que sigue el estándar System V. Y al respecto de ésta herramienta hay que mencionar una serie de cosas.

1. Los semáforos en System V no tienen implementada la creación y la inicialización de manera atómica, pero nuestra clase Semáforo realiza todo en el método constructor, para evitar problemas de *race condition*.
2. Los semáforos System V en C modelan en verdad un set o conjunto de semáforos, y se utilizó para desarrollar este trabajo un conjunto de 4 semáforos:
 - Un semáforo que lleva la cuenta de cuántos recursos del tipo masa preparada hay a disposición en lo que va de la jornada laboral. Este semáforo fue inicializado en 0, y es utilizado por la Herramienta de Ingredientes
 - Un semáforo que lleva la cuenta de cuántos recursos del tipo ingredientes agregados (a la masa) hay a disposición en lo que va de la jornada laboral. Este semáforo también fue inicializado en 0 y es consultado por el Rallador de Queso.
 - Un último semáforo inicializado en 0 que lleva la cuenta de la cantidad de pizzas en el horno. Este semáforo es utilizado por el Rallador de Quesos y la entidad Horno, que lo incrementan y decrementan respectivamente

- Y finalmente un semáforo inicializado en 5 que modela la cantidad de lugares disponibles en el horno. Este semáforo es también utilizado por el Rallador de Quesos y la entidad Horno, que lo decrementan e incrementan a medida que van metiendo y sacando las pizzas.

3. Como se utilizó el flag SEM UNDO, cada proceso espera a que los semáforos en los que realizó alguna operación alcancen un estado determinado. De ésta manera, evita que se reviertan los cambios aplicados, pero sin descuidarse ante una eventual finalización abrupta del proceso en cuestión.

A continuación se incluye un diagrama de secuencia que refleja el flujo del programa a lo largo de la preparación de una pizza



1.3 Ejecuciones

Veamos la primera ejecución del programa, en la que se fabrica un total de 10 pizzas.

```

Masa fabricada pizza número 1
Masa fabricada pizza número 2
Agregados los ingredientes a la pizza número 1
Masa fabricada pizza número 3
Agregados los ingredientes a la pizza número 2
Rallé el queso de la pizza número 1
Metí la pizza número 1 en el horno
Agregados los ingredientes a la pizza número 3
Masa fabricada pizza número 4
Agregados los ingredientes a la pizza número 4
Masa fabricada pizza número 5
  
```

Rallé el queso de la pizza número 2
Metí la pizza número 2 en el horno
Agregados los ingredientes a la pizza número 5
Horneé la pizza número 1 y hay un espacio más!
Rallé el queso de la pizza número 3
Metí la pizza número 3 en el horno
Masa fabricada pizza número 6
Masa fabricada pizza número 7
Rallé el queso de la pizza número 4
Metí la pizza número 4 en el horno
Masa fabricada pizza número 8
Rallé el queso de la pizza número 5
Agregados los ingredientes a la pizza número 6
Masa fabricada pizza número 9
Metí la pizza número 5 en el horno
Horneé la pizza número 2 y hay un espacio más!
Agregados los ingredientes a la pizza número 7
Rallé el queso de la pizza número 6
Masa fabricada pizza número 10
Metí la pizza número 6 en el horno
Agregados los ingredientes a la pizza número 8
Horneé la pizza número 3 y hay un espacio más!
Rallé el queso de la pizza número 7
Metí la pizza número 7 en el horno
Agregados los ingredientes a la pizza número 9
Rallé el queso de la pizza número 8
Metí la pizza número 8 en el horno
Agregados los ingredientes a la pizza número 10
Rallé el queso de la pizza número 9
Horneé la pizza número 4 y hay un espacio más!
Metí la pizza número 9 en el horno
Rallé el queso de la pizza número 10
Horneé la pizza número 5 y hay un espacio más!
Metí la pizza número 10 en el horno
Horneé la pizza número 6 y hay un espacio más!
Horneé la pizza número 7 y hay un espacio más!
Horneé la pizza número 8 y hay un espacio más!
Horneé la pizza número 9 y hay un espacio más!
Horneé la pizza número 10 y hay un espacio más!

Lo primero que se debe aclarar es que, aunque cada parte del proceso lleva un tiempo variable según el tipo de pizza, el horneado es la operación que más tarda. Esto se ve reflejado ya con una cantidad baja de pedidos.

Veamos ahora lo que sucede con un total de 25 pizzas pedidas

Masa fabricada pizza número 1

Masa fabricada pizza número 2
Agregados los ingredientes a la pizza número 1
Masa fabricada pizza número 3
Masa fabricada pizza número 4
Agregados los ingredientes a la pizza número 2
Agregados los ingredientes a la pizza número 3
Rallé el queso de la pizza número 1
Masa fabricada pizza número 5
Metí la pizza número 1 en el horno
Agregados los ingredientes a la pizza número 4
Masa fabricada pizza número 6
Rallé el queso de la pizza número 2
Metí la pizza número 2 en el horno
Agregados los ingredientes a la pizza número 5
Masa fabricada pizza número 7
Rallé el queso de la pizza número 3
Agregados los ingredientes a la pizza número 6
Metí la pizza número 3 en el horno
Horneé la pizza número 1 y hay un espacio más!
Masa fabricada pizza número 8
Rallé el queso de la pizza número 4
Metí la pizza número 4 en el horno
Agregados los ingredientes a la pizza número 7
Rallé el queso de la pizza número 5
Metí la pizza número 5 en el horno
Masa fabricada pizza número 9
Agregados los ingredientes a la pizza número 8
Rallé el queso de la pizza número 6
Metí la pizza número 6 en el horno
Masa fabricada pizza número 10
Rallé el queso de la pizza número 7
Masa fabricada pizza número 11
Horneé la pizza número 2 y hay un espacio más!
Metí la pizza número 7 en el horno
Agregados los ingredientes a la pizza número 9
Masa fabricada pizza número 12
Masa fabricada pizza número 13
Rallé el queso de la pizza número 8
Horneé la pizza número 3 y hay un espacio más!
Metí la pizza número 8 en el horno
Agregados los ingredientes a la pizza número 10
Masa fabricada pizza número 14
Masa fabricada pizza número 15
Rallé el queso de la pizza número 9
Agregados los ingredientes a la pizza número 11
Agregados los ingredientes a la pizza número 12

Masa fabricada pizza número 16
Masa fabricada pizza número 17
Horneé la pizza número 4 y hay un espacio más!
Metí la pizza número 9 en el horno
Agregados los ingredientes a la pizza número 13
Masa fabricada pizza número 18
Agregados los ingredientes a la pizza número 14
Rallé el queso de la pizza número 10
Masa fabricada pizza número 19
Agregados los ingredientes a la pizza número 15
Horneé la pizza número 5 y hay un espacio más!
Metí la pizza número 10 en el horno
Masa fabricada pizza número 20
Agregados los ingredientes a la pizza número 16
Rallé el queso de la pizza número 11
Agregados los ingredientes a la pizza número 17
Agregados los ingredientes a la pizza número 18
Masa fabricada pizza número 21
Horneé la pizza número 6 y hay un espacio más!
Metí la pizza número 11 en el horno
Rallé el queso de la pizza número 12
Agregados los ingredientes a la pizza número 19
Masa fabricada pizza número 22
Horneé la pizza número 7 y hay un espacio más!
Metí la pizza número 12 en el horno
Rallé el queso de la pizza número 13
Agregados los ingredientes a la pizza número 20
Masa fabricada pizza número 23
Agregados los ingredientes a la pizza número 21
Masa fabricada pizza número 24
Horneé la pizza número 8 y hay un espacio más!
Metí la pizza número 13 en el horno
Masa fabricada pizza número 25
Rallé el queso de la pizza número 14
Agregados los ingredientes a la pizza número 22
Agregados los ingredientes a la pizza número 23
Agregados los ingredientes a la pizza número 24
Horneé la pizza número 9 y hay un espacio más!
Metí la pizza número 14 en el horno
Agregados los ingredientes a la pizza número 25
Rallé el queso de la pizza número 15
Horneé la pizza número 10 y hay un espacio más!
Metí la pizza número 15 en el horno
Rallé el queso de la pizza número 16
Horneé la pizza número 11 y hay un espacio más!
Metí la pizza número 16 en el horno

Rallé el queso de la pizza número 17
Horneé la pizza número 12 y hay un espacio más!
Metí la pizza número 17 en el horno
Rallé el queso de la pizza número 18
Horneé la pizza número 13 y hay un espacio más!
Metí la pizza número 18 en el horno
Rallé el queso de la pizza número 19
Horneé la pizza número 14 y hay un espacio más!
Metí la pizza número 19 en el horno
Rallé el queso de la pizza número 20
Horneé la pizza número 15 y hay un espacio más!
Metí la pizza número 20 en el horno
Rallé el queso de la pizza número 21
Horneé la pizza número 16 y hay un espacio más!
Metí la pizza número 21 en el horno
Rallé el queso de la pizza número 22
Horneé la pizza número 17 y hay un espacio más!
Metí la pizza número 22 en el horno
Rallé el queso de la pizza número 23
Horneé la pizza número 18 y hay un espacio más!
Metí la pizza número 23 en el horno
Rallé el queso de la pizza número 24
Horneé la pizza número 19 y hay un espacio más!
Metí la pizza número 24 en el horno
Rallé el queso de la pizza número 25
Horneé la pizza número 20 y hay un espacio más!
Metí la pizza número 25 en el horno
Horneé la pizza número 21 y hay un espacio más!
Horneé la pizza número 22 y hay un espacio más!
Horneé la pizza número 23 y hay un espacio más!
Horneé la pizza número 24 y hay un espacio más!
Horneé la pizza número 25 y hay un espacio más!

Con esta cantidad mayor queda claro cómo el Fabricante de Masa y la Herramienta de Ingredientes terminan su trabajo más rápidamente que el Rallador de Queso, el cual queda constantemente supeditado a la espera de un nuevo espacio disponible para hornear la pizza, una vez que ya hay cinco pizzas cocinándose.

Con este informe se adjunta una carpeta de ejecuciones donde se encontrará el resultado de las corridas con 50 y 250 pizzas

2 Ejercicio 2

Un lock es un mecanismo de sincronismo de acceso a un determinado archivo o recurso. Los locks pueden ser compartidos o exclusivos. Un lock exclusivo permite a una serie de procesos o threads, por ejemplo, abrir el mismo archivo en modo escritura y, tomando el lock antes de ejecutar la escritura en sí, evitar problemas de sincronismo causados por el interleaving de las ejecuciones del programa concurrente.

Un semáforo es también un mecanismo de acceso a un determinado archivo o recurso. Dentro de su estructura interna, el semáforo posee un contador. Si ese contador es mayor que cero, por ejemplo N, quiere decir que hay "N recursos disponibles". Si ese contador es cero, entonces no hay recursos disponibles. El contador de un semáforo no debería ser nunca negativo. Particularmente, un semáforo binario es aquel que puede adoptar, en su contador interno, los valores 0 o 1. Esto da como resultado un mecanismo de IPC casi idéntico al lock exclusivo. La diferencia crucial entre este último y un semáforo binario (amen de que conceptualmente son entidades diferentes) es que un lock solo puede ser devuelto por el proceso o thread que lo tomó. En cambio, un semáforo puede recibir las operaciones wait() o signal() desde procesos o threads diferentes.

La siguiente situación modela un caso particular que puede ser resuelto con un semáforo binario y no con un lock exclusivo.

Supongamos una represa hidroeléctrica que posee dos grandes fases de control del agua. Cada una de estas fases está manejada por un proceso único, y estos dos procesos se ejecutan de manera concurrente. Una de las situaciones más críticas es el tema de abrir y cerrar la compuerta principal. Como las fases se encuentran en serie, el nivel de agua de la primera fase luego se transporta a la segunda fase. Observemos el siguiente fragmento de código correspondiente a la Fase 1:

```
while(gracefulQuit()) {
    // ... //
    if(nivelDeAguaEsBajo()) {
        semaforo->p();
        abrirCompuertas();
    }
    // ... //
}
```

En primer lugar, se utiliza para modelar la disponibilidad de las compuertas un Semáforo Binario, *sem*. Si la Fase 1 detecta que el nivel de agua es bajo, entonces espera a tener el control de las compuertas ejecutando la función wait sobre un semaforo, y entonces las abre. Esto permite el ingreso del agua que se transformará en electricidad dentro de la usina. Veamos un fragmento del proceso correspondiente a la Fase 2.


```

while(gracefulQuit()) {
    // ... //
    if(nivelDeAguaEsAlto()) {
        if(compuertasAbiertas()) {
            cerrarCompuestas();
            semaforo->v();
        }
    }
    // ... //
}

```

Como podemos ver, la Fase 2 en algún momento de su ejecución verifica si el nivel de agua es alto. Puede ser que esto suceda por una serie de factores, por eso a continuación chequea si las compuertas están abiertas, y en caso de estarlo las cierra y deja a disposición este recurso nuevamente ejecutando `signal()` o `v()` sobre el semáforo. Esta situación no puede solucionarse con un Lock Exclusivo porque si el proceso de la Fase 1 decide tomarlo, entonces este mismo es el encargado de devolverlo, cuando acá podemos ver que claramente cada Fase se encarga de abrir o cerrar la compuerta y hacer o no disponible el recurso.

Una aclaración extremadamente importante es que, para que todo este escenario tenga sentido y funcione, nuestro semáforo fue creado e inicializado atómicamente con un valor inicial de 1, y luego cada proceso de cada Fase lo recibió ya listo para ser utilizado.

```

Semaforo* sem = new Semaforo(FILE, 1);
// ... //

```