

Obligatorio Primer Semestre 2021

Segunda Entrega



Universidad
Católica del
Uruguay

Ingeniería Informática

Sistemas Operativos

Gastón Cabrera, Francisco Hernández, Gonzalo Sempol

ÍNDICE:

Identificación de conceptos y problemas similares de sistemas operativos que puedan aprovechar para el diseño de la solución.	3
Posibles implementaciones prácticas:	4
Opción 1:	4
Opción 2:	5
Opción 3:	6
Justificación de la alternativa a implementar:	7
Un bosquejo de la simulación que se efectuará:	7
Descripción del escenario de la simulación	8
Entradas y Salidas:	9
Entradas:	9
Salidas esperadas:	9
Para las Personas:	9
Para el Vacunador:	9
Mediciones que se realizarán:	10

Identificación de conceptos y problemas similares de sistemas operativos que puedan aprovechar para el diseño de la solución.

Se implementará un sistema con múltiples entradas, donde las mismas se harán desde distintos canales, todos hacia el mismo servidor. Aquí se identifican una comunicación entre procesos y distintos hilos de ejecución que deberán encargarse de las I/Os asociadas. A su vez cada uno de estos hilos deberá respetar ciertas condiciones de sincronización para evitar errores de escritura y lectura en memoria. Es decir no se permitirá acceder a 2 hilos a su sección crítica a la vez por lo que se respetará la exclusión mutua, y para esto se utilizarán distintos semáforos para los procesos de entrada y salida.

Por ejemplo, para poner en espera de agendar a 2 personas en un mismo departamento no deberían poder hacerlo a la vez, pues podrían provocar una colisión. Para evitar esto se implementará un semáforo que controla la escritura de una nueva persona en la cola de espera de cada departamento, sin importar el medio por el cual esta se anote.

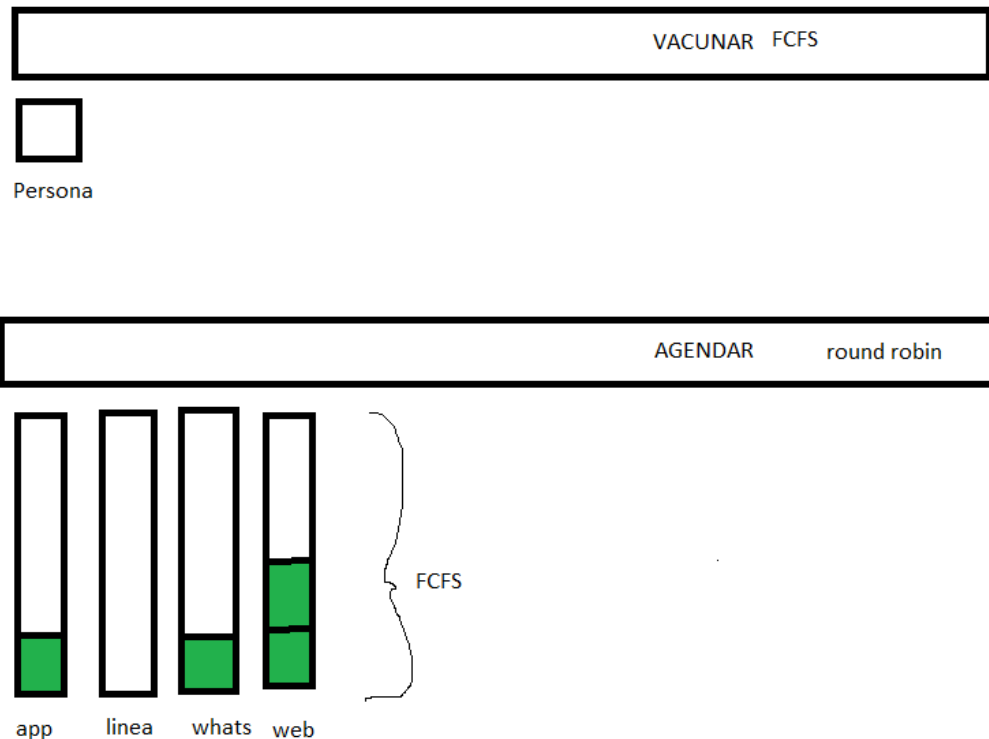
A nivel más general se identifica la necesidad de implementar un planificador de múltiples colas que presente distintos criterios de planificación, a fin de lograr que las distintas partes del sistema se ejecuten dentro de los tiempos esperados.

Por ejemplo uno de los criterios a utilizar podría ser el de Round Robin para los múltiples canales de entrada y salida, esto permite ser equitativo a la hora de tratar con los usuarios que se conectan al sistema por los distintos canales disponibles.

Lo anterior nos permite ver que todas las partes del sistema se encuentran particionadas en distintos hilos que ejecutan en simultáneo lo que denota la existencia de un sistema multihilo. Todos estos hilos se encuentran cargados en memoria lo que denota multiprogramación.

Posibles implementaciones prácticas:

Opción 1:

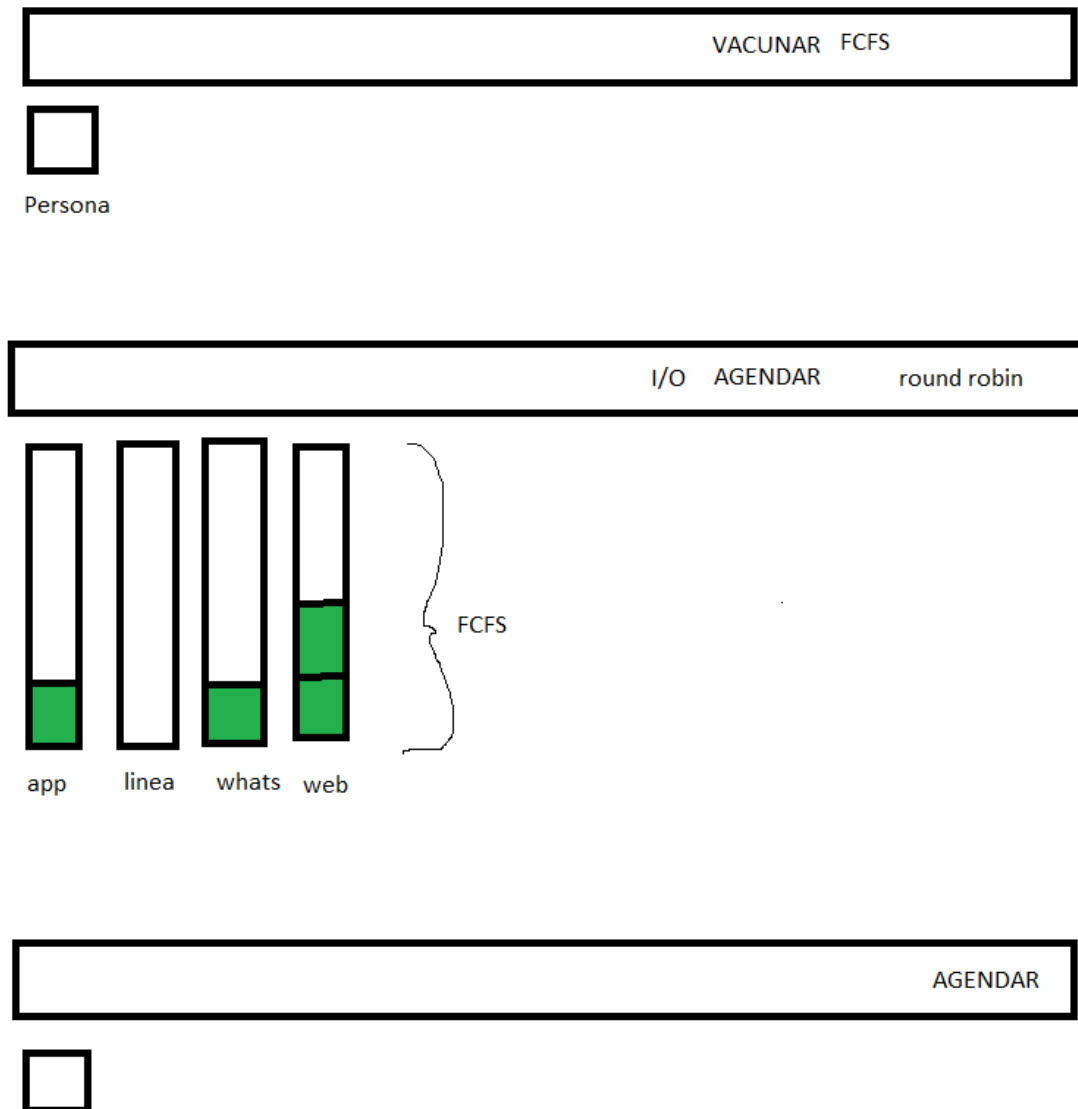


Para la primera opción se implementará un planificador de múltiples colas, donde en el primer nivel tendremos una cola de prioridad con tres procesos. El proceso de mayor prioridad es vacunar, seguido por el proceso de agendar.

Los hilos del proceso vacunar se modelarán como una cola FCFS donde cada elemento será una persona al vacunarse. Al terminar cada uno, se marcará la persona como vacunada.

El proceso de agendar, se modelará como 4 colas, una por cada método de agendación, a través de la APP, por línea telefónica, etc. Las colas se seleccionarán con un algoritmo de Round Robin, para ser equitativo. Cada elemento de cada cola implicará que la persona ingresará su CI, y un departamento, y el sistema determinará su edad, para definir si se encuentra en el rango habilitado, posteriormente procederá a asignarle un vacunatorio, una fecha y un horario via email y teléfono, completando el proceso de agendado.

Opción 2:

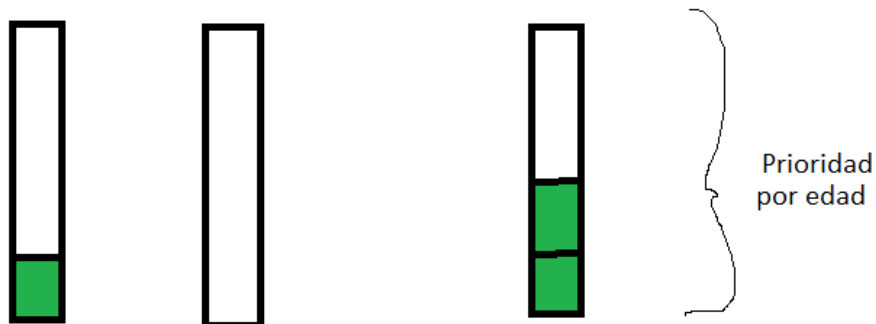
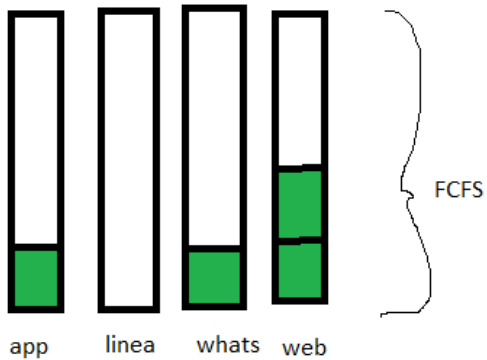


Para la segunda opción se implementará un planificador de múltiples colas, donde en el primer nivel tendremos una cola de prioridad con cuatro procesos. En este caso se divide, Agendar en 2, por una lado se atiende la I/O y se encola a las personas y por otro el proceso mismo de buscar un vacunatorio disponible, y la agendarlas. Esto permite disminuir la carga a las operaciones que entren al sistema desacoplando la parte más compleja del cálculo. La parte de Agendar per se, se coloca en la cola con menor prioridad para que el sistema solo agende cuando tenga tiempo de CPU, priorizando así la respuesta al usuario, y sentirán que el sistema es más responsivo. Mientras agendar se procese el usuario esperará ser notificado, este proceso puede demorar mucho tiempo.

Opción 3:



Persona



En este caso se mejora sobre el caso anterior, modelando Agendar como múltiples colas ordenadas por rango de edad , cada una de estas se ordena por FCFS para respetar el orden de llegada de las personas, de esta forma se procesarán en orden y si en el futuro alguien de un rango anterior se anota fuera de fecha, ejemplo, alguien de 80 se anota cuando está abierto para 60, se le dará prioridad al de 80.

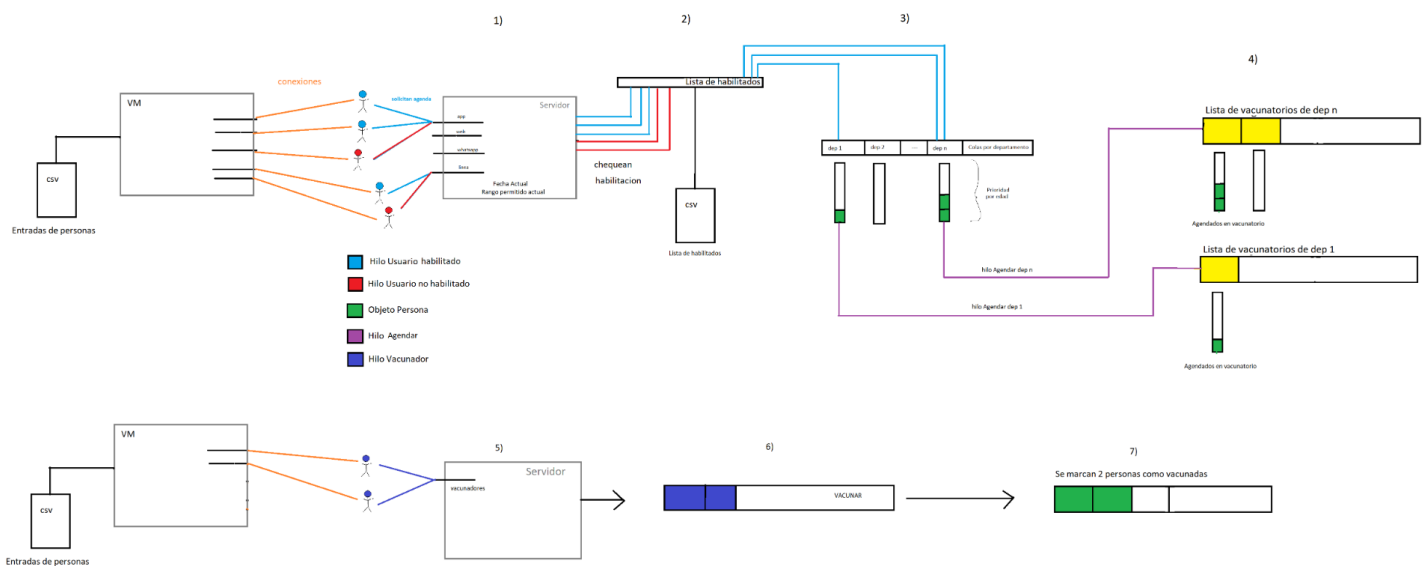
Justificación de la alternativa a implementar:

Hemos decidido implementar la opción número 3.

La decisión de haber elegido esta opción se debe a que se busca darle prioridad a los procesos de I/O de cara al usuario, empezando con marcar como vacunado, y seguido de la agenda por medio de los 4 métodos posibles para agendarse, esto hace que el usuario perciba al sistema como responsivo. Además esta solución desacopla los cálculos pesados de las solicitudes asociadas a ellos, maximizando la cantidad de usuarios que pueden ser atendidos en simultáneo, sin que se degrade la calidad del servicio. Es decir, permite dar un tiempo de respuesta correcto.

También esta implementación permitirá que al abrir distintos rangos, además de poder inscribirse las personas correspondientes, se permite que personas de rangos anteriormente abiertos a inscripción se inscriban y a su vez, se manejen prioritariamente frente a los otros.

Un bosquejo de la simulación que se efectuará:



Descripción del escenario de la simulación

Tendremos 1 máquina virtual cliente, que establecerá conexiones y por cada una, creará un hilo, luego mandará una CI y un departamento al servidor y esperará por el mensaje de que está en cola para recibir una fecha y hora, o que no está habilitado para inscribirse por alguna razón.

1) Servidor:

El servidor va a tener 5 puertos escuchando, uno para cada canal de entrada de datos posible; 81: web, 82: app, 83: chatbot, 84: línea, 85: vacunadores.

Hay 5 hilos escuchando en el servidor, (web, app, chatbot, línea, vacunadores), 5 usuarios intentan conectarse al sistema por las siguientes vías, 3 por app, y 2 por línea, lo que conlleva a la creación de 5 hilos usuarios.

2) Cada hilo usuario verificará en la lista de usuarios habilitados si le corresponde la agenda, si no recibirá uno de los siguientes errores:

- a) Su edad no coincide con el rango habilitado.
- b) El Usuario ya está en cola.
- c) El Usuario ya recibió día y hora.
- d) El Usuario ya recibió las vacunas.

Se contará con un semáforo por cada persona para controlar si esta ya ingresó por otro canal o no.

3) De los 5 que entraron, solo 3 estaban habilitados, cada uno solicitó el semáforo al departamento para poder insertarse en la cola, aquí se ven, distribuidos por departamento, ingresados en cada cola en orden según su edad, esperando a ser atendidos por el hilo Agendar de su departamento.(Se atenderán por prioridad de rangos).

4) El hilo agendar de cada departamento solicita el semáforo y toma las personas en cola en orden según su edad, y los agenda en cada vacunatorio según disponibilidad de fechas. (No se agendará en un vacunatorio en una fecha hasta no completar el anterior) Los hilos terminan su ejecución, se solicita el semáforo de la persona y se la marcará como agendada.

Proceso Vacunar:

Se cuenta con una máquina virtual para simular el accionar de los vacunadores.

Ésta establecerá conexiones con el servidor, enviando CI del vacunado.

- 5) Existe un hilo vacunador escuchando en el servidor, 2 vacunadores intentan conectarse, lo que conlleva a la creación de 2 hilos vacunador.
- 6) Estos se encolan para posteriormente ser atendidos en la cola Vacunar.
- 7) Ambos hilos terminaron su ejecución generando como salida 2 personas vacunadas. (Marcan en el sistema a cada persona como vacunada)

Entradas y Salidas:

Entradas:

El cliente (VM) tendrá un .csv, que cargará a memoria, con un listado de cédulas de identidad, válidas y no válidas, seguido de su departamento y número de puerto por el que se quiere conectar. Estos serán los datos que se enviarán.

Ejemplo:

```
1000000;Montevideo;84
1000001;Montevideo;84
1000002;Montevideo;83
1000003;Montevideo;84
1000004;Montevideo;81
```

A su vez habrá otro .csv para los vacunadores, que contendrá la cédula del vacunador, y la cédula de la persona a la que pretende vacunar.

```
555555 1111111
444444 2222222
```

Salidas esperadas:

Para las Personas:

1. CORRECTO: Su solicitud será procesada.
2. Error: Su edad no coincide con el rango habilitado.
3. Error: El Usuario ya está esperando a ser agendado.
4. Error: El Usuario ya recibió día y hora.
5. Error: El Usuario ya recibió las vacunas.

Para el Vacunador:

1. CORRECTO: Vacunación de ci XXXXXX exitosa, le quedan Y dosis.
2. Error: La persona no está agendada.
3. Error: La persona ya tiene 2 dosis.

Mediciones que se realizarán:

Se medirá el tiempo con un timestamp, desde que se inicia la conexión en el cliente, hasta que se recibe una respuesta del servidor, ya sea que está en cola esperando ser agendado, como un error de que no está habilitado en el rango de edad. Esto permitirá saber si el sistema es responsivo.

Del lado del servidor se podrá medir la cantidad de transacciones por minuto que el servidor puede procesar.

Se generará un reporte con gráfico de barras, de los vacunados por rango de edad para ver a cuales se les dio prioridad.