

## Programación de servicios y procesos - 2324

[Área personal](#)

[Mis cursos](#)

[FORMACIÓN PROFESIONAL 23/24](#)

[CIFP CARLOS III](#)

[CGS Desarrollo de Aplicaciones Multiplataforma 23/24](#)

[Segundo\(DAM\)](#)

[Programación de servicios y procesos - 2324](#)

[1ª CONVOCATORIA](#)

[Tarea para PSP02. \(hasta el 5/11/23 23:55\)](#)

[Editar entrega](#)

Tarea para PSP02. (hasta el 5/11/23 23:55)

## Tarea para PSP02.

### Detalles de la tarea de esta unidad.

Enunciado.

Programa en Java las siguientes aplicaciones con hilos. No utilices un jdk superior al 17.

### Ejercicio 1

Una biblioteca pequeña de un pueblo va a ser visitada el sábado por 8 usuarios. Todos los usuarios van a llevarse un libro de la biblioteca. Como la sala dónde se encuentran los libros es muy pequeña, sólo pueden entrar 3 usuarios a la vez a la zona de libros. Una vez los usuarios han escogido sus libros se dirigen a la zona de préstamo y se van. En la zona de préstamo existen 2 mostradores dónde pueden atenderles. Ese día además irá el limpiador de la biblioteca. El funcionamiento de la biblioteca que tienes que implementar es el siguiente:

- Los usuarios y el limpiador son hilos.
- Se espera la llegada de 8 usuarios ese día.
- Sólo pueden entrar en la zona de libros 3 usuarios a la vez.
- Una vez el usuario sale de la zona de libros se dirige a la zona de préstamos.
- Existen 2 mostradores en la zona de préstamos. Cada uno de los mostradores atiende a los usuarios de uno en uno. Los usuarios se dirigirán al primer mostrador que quede libre. Como máximo tendremos a 2 usuarios en la zona de préstamos.
- El tiempo que el usuario pasa mirando libros en la zona de libros es random, pero suele ser entre 1 y 3 segundos.
- El tiempo que pasa el usuario en el mostrador de préstamo es de 1 segundo siempre.
- Para simular los tiempos que pasan los usuarios en la biblioteca utiliza un sleep.
- El limpiador va a ir también ese día a limpiar. El limpiador necesita que tanto la zona de libros como los mostradores estén vacíos para poder entrar a limpiarlos. El limpiador limpia primero la zona de libros y después el mostrador. El limpiador tarda 2 segundos en limpiar la zona de libros y 2 segundos en limpiar los mostradores.
- Implementa mediante **semáforos** la entrada y salida a la zona de libros y a la zona de préstamos tanto de los usuarios como del limpiador.
- Una vez los usuarios y el limpiador se han marchado, la biblioteca cuelga el cartel de "BIBLIOTECA CERRADA". Para implementar esto haz uso del **CountDownLatch**.
- El programa termina cuando se cierra la biblioteca.

Consideraciones sobre el ejercicio:

- Debes usar los mismos mensajes en castellano que te doy en la salida de ejemplo.
- Suponemos que el limpiador llega a la biblioteca después del usuario 2. Es decir, el orden de los start() es Usuario1, Usuario2, Limpiador, Usuario3, Usuario4, Usuario5, Usuario6, Usuario7, Usuario8.
- La salida que te doy a continuación, es orientativa, los hilos pueden ejecutarse en otro orden pero siguiendo las consideraciones del ejercicio descritas anteriormente.

*ABRE LA BIBLIOTECA*

*Usuario1 esperando poder entrar a la zona de libros*

*-- ENTRA a la zona de libros Usuario1*

*Usuario2 esperando poder entrar a la zona de libros*

*-- ENTRA a la zona de libros Usuario2*

*Limpiador esperando poder entrar a la zona de libros*

*Usuario3 esperando poder entrar a la zona de libros*

*-- ENTRA a la zona de libros Usuario3*

*Usuario4 esperando poder entrar a la zona de libros*

*Usuario5 esperando poder entrar a la zona de libros*

*Usuario6 esperando poder entrar a la zona de libros*

Usuario7 esperando poder entrar a la zona de libros

Usuario8 esperando poder entrar a la zona de libros

-- SALE de la zona de libros Usuario2

Usuario2 esperando turno para el mostrador de préstamos

\*\* ENTRA mostrador de préstamos Usuario2

-- SALE de la zona de libros Usuario3

Usuario3 esperando turno para el mostrador de préstamos

\*\* ENTRA mostrador de préstamos Usuario3

\*\* SALE mostrador de préstamos Usuario2

-- SALE de la zona de libros Usuario1

&&&&& EMPIEZA A LIMPIAR la zona de libros Limpiador

Usuario1 esperando turno para el mostrador de préstamos

\*\* ENTRA mostrador de préstamos Usuario1

\*\* SALE mostrador de préstamos Usuario3

\*\* SALE mostrador de préstamos Usuario1

&&&&& TERMINA DE LIMPIAR la zona de libros Limpiador

-- ENTRA a la zona de libros Usuario4

-- ENTRA a la zona de libros Usuario5

-- ENTRA a la zona de libros Usuario6

Limpiador esperando para limpiar los mostradores de préstamos

&&&&& EMPIEZA A LIMPIAR los mostradores de préstamos Limpiador

-- SALE de la zona de libros Usuario4

-- ENTRA a la zona de libros Usuario7

Usuario4 esperando turno para el mostrador de préstamos

-- SALE de la zona de libros Usuario7

-- ENTRA a la zona de libros Usuario8

Usuario7 esperando turno para el mostrador de préstamos

&&&&& TERMINA DE LIMPIAR los mostradores de préstamos Limpiador

\*\* ENTRA mostrador de préstamos Usuario4

\*\* ENTRA mostrador de préstamos Usuario7

-- SALE de la zona de libros Usuario5

Usuario5 esperando turno para el mostrador de préstamos

-- SALE de la zona de libros Usuario6

Usuario6 esperando turno para el mostrador de préstamos

\*\* SALE mostrador de préstamos Usuario4

\*\* ENTRA mostrador de préstamos Usuario5

\*\* SALE mostrador de préstamos Usuario7

\*\* ENTRA mostrador de préstamos Usuario6

-- SALE de la zona de libros Usuario8

Usuario8 esperando turno para el mostrador de préstamos

\*\* SALE mostrador de préstamos Usuario5

\*\* ENTRA mostrador de préstamos Usuario8

\*\* SALE mostrador de préstamos Usuario6

\*\* SALE mostrador de préstamos Usuario8

BIBLIOTECA CERRADA

## Ejercicio 2

Crea un programa en Java que controla el acceso a una carretera que sólo tiene una dirección en dos sentidos: este y oeste. Vas a tener en cuenta lo siguiente:

- Los coches que llegan por el oeste y los coches que lleguen por el este son hilos. En total ese día van circular 20 coches, 10 que vienen del oeste y otros 10 que vienen del este.
- Cuando un coche que viene por el oeste llega a la carretera sólo cruzará si no existe otro en sentido contrario cruzando la carretera. Varios coches que llegan por el oeste pueden cruzar uno detrás de otro. Lo mismo ocurre con los coches que llegan por el este.
- Cada coche tarda en cruzar la carretera un tiempo aleatorio de 1 a 3 segundos.
- Deben implementarse **monitores** para los métodos que controlan el acceso y salida de la carretera en ambos sentidos y los métodos wait(), notify() y notifyAll() de la clase Object para sincronizar los hilos.
- Vamos a crear un **pool de hilos (Executor)** para que sólo puedan haber 5 hilos en ejecución a la vez.
- El orden de los start() será intercalado, oeste, este, oeste, este, oeste, este...
- Utilizar los mismos mensajes de salida en la aplicación o al menos parecidos, que pueda verse el nombre del coche, cuando intenta entrar, cuando entra y cuando sale.
- La salida que te doy a continuación, es orientativa, los hilos pueden ejecutarse en otro orden pero siguiendo las consideraciones del ejercicio descritas anteriormente.

### COCHES OESTE



### COCHES ESTE



### COCHES OESTE



### COCHES ESTE

Posible salida del programa

Quiere entrar por el oeste OESTE0

--> Entra por el oeste el coche OESTE0

Quiere entrar por el este ESTE0

Quiere entrar por el oeste OESTE1

--> Entra por el oeste el coche OESTE1

Quiere entrar por el este ESTE1

Quiere entrar por el oeste OESTE2

--> Entra por el oeste el coche OESTE2

<-- Sale el coche OESTE0

Quiere entrar por el este ESTE2

<-- Sale el coche OESTE1

Quiere entrar por el oeste OESTE3

--> Entra por el oeste el coche OESTE3

<-- Sale el coche OESTE2

Quiere entrar por el este ESTE3

<-- Sale el coche OESTE3

--> Entra por el este el coche ESTE0

--> Entra por el este el coche ESTE3

--> Entra por el este el coche ESTE2

--> Entra por el este el coche ESTE1

Quiere entrar por el oeste OESTE4

<-- Sale el coche ESTE2

Quiere entrar por el este ESTE4

--> Entra por el este el coche ESTE4

<-- Sale el coche ESTE3

Quiere entrar por el oeste OESTE5

<-- Sale el coche ESTE0

Quiere entrar por el este ESTE5

--> Entra por el este el coche ESTE5

<-- Sale el coche ESTE1

Quiere entrar por el oeste OESTE6

<-- Sale el coche ESTE4

Quiere entrar por el este ESTE6

--> Entra por el este el coche ESTE6

<-- Sale el coche ESTE5

Quiere entrar por el oeste OESTE7

<-- Sale el coche ESTE6

--> Entra por el oeste el coche OESTE4

--> Entra por el oeste el coche OESTE7

--> Entra por el oeste el coche OESTE6

--> Entra por el oeste el coche OESTE5

Quiere entrar por el este ESTE7

<-- Sale el coche OESTE4

Quiere entrar por el oeste OESTE8

--> Entra por el oeste el coche OESTE8

<-- Sale el coche OESTE5

Quiere entrar por el este ESTE8

<-- Sale el coche OESTE6

*Quiere entrar por el oeste OESTE9*

*--> Entra por el oeste el coche OESTE9*

*<-- Sale el coche OESTE7*

*Quiere entrar por el este ESTE9*

*<-- Sale el coche OESTE9*

*<-- Sale el coche OESTE8*

*--> Entra por el este el coche ESTE7*

*--> Entra por el este el coche ESTE8*

*--> Entra por el este el coche ESTE9*

*<-- Sale el coche ESTE9*

*<-- Sale el coche ESTE7*

*<-- Sale el coche ESTE8*

### **Criterios de puntuación. Total 10 puntos.**

Total 10 puntos.

- **Ejercicio 1.** 5 puntos.
- **Ejercicio 2.** 5 puntos.

Se tendrá en cuenta en ambos ejercicios que:

- La ejecución de los programas produce el resultado esperado.
- No se produce interbloqueo ni inanición.
- Los hilos están bien implementados.
- Uso correcto de los métodos de sincronización vistos en los contenidos.
- Programas bien comentados, no es necesario Javadoc.
- Mensajes claros por pantalla de lo que hacen en cada momento los hilos.

### **Recursos necesarios para realizar la Tarea.**

**Ejercicio 1.**– Los contenidos y ejemplos realizados en la Unidad.

**Ejercicio 2.**– Los contenidos y ejemplos realizados en la Unidad.

### **Consejos y recomendaciones.**

Para el **Ejercicio1.**– Descarga, lee, ejecuta y comprende los proyectos que tienes de ejemplo de hilos, semáforos y CountdownLatch. Empezaría creando la estructura del programa con los hilos. Analizaría cuantos semáforos necesito, cómo lo voy a inicializar, cómo lo voy a usar. Por último, implementaría el CountdownLatch.

Para el **Ejercicio2.**– Descarga, lee, ejecuta y comprende los proyectos que tienes de ejemplo de hilos, monitores y Executor. Empezaría creando la estructura del programa con los hilos. Analizaría las condiciones de entrada y salida de la carretera según el coche que quiere acceder y si la carretera está ocupada o libre. Por último modificaría la forma de lanzar los hilos utilizando Executor para crear el pool de hilos.

### **Indicaciones de entrega.**

Lo que debes entregar:

- Los proyectos que puedan abrirse en NetBeans de las dos actividades bien comentados. No es necesario Javadoc ni generar el .jar. Te recomiendo que los crees como proyectos Maven.
- Manual con las pruebas realizadas. Únicamente es necesario los pantallazos de la salida de tu programa y una pequeña explicación de lo que ocurre o si has utilizado algún argumento de entrada a tu programa que deba yo poner también. En los pantallazos debe visualizarse tu aula virtual y la fecha del ordenador.
- Los proyectos serán probados con un jkd17.

El manual solicitado debe presentarse en formato pdf y su contenido debe ser el adecuado, portada, sin faltas de ortografía, bien organizado y claro.

Debes crear una carpeta para el ejercicio, y en ella una carpeta para cada actividad. En cada carpeta incluye el proyecto correspondiente. Puedes crear un único manual para las dos actividades. Comprimelo todo en un fichero .zip.

El envío se realizará a través de la plataforma de la forma establecida para ello, y el archivo comprimido se nombrará siguiendo las siguientes pautas:

**apellido1\_apellido2\_nombre\_SIGxx\_Tarea**

Asegúrate que el nombre no contenga la letra ñ, tildes ni caracteres especiales extraños. Así por ejemplo la alumna **Begoña Sánchez Mañas para la segunda unidad del MP de PSP**, debería nombrar esta tarea como...



**sanchez\_manas\_begona\_PSP02\_Tarea**

**Fecha de entrega:** 5 de Noviembre de 2023 a las 23:55




**Fecha límite de entrega con penalización del 30%:** 12 de Noviembre de 2022 a las 23:55. A partir de la fecha límite de entrega no se aceptará ninguna tarea.

Archivos enviados

Tamaño máximo de archivo: 5MB, número máximo de archivos: 1



[Archivos](#)



Puede arrastrar y soltar archivos aquí para añadirlos

Guardar cambios

Cancelar


Actividad previa

◀ [Tarea para PSP01 \(hasta el 08/10/23 23:55\)](#)

Ir a...

## Mantente en contacto

Dirección General de Formación Profesional e Innovación

 Resumen de retención de datos

 Descargar la app para dispositivos móviles

TODOS LOS DERECHOS RESERVADOS 

El tema fue desarrollado por  por [conecti.me](https://conecti.me)