

Guía de corrección

Nombre de los estudiantes:	
Grupo de prácticas :	
Nota final:	A rellenar por el profesor

***** LEER DETENIDAMENTE *****

Para la corrección de las prácticas, estos son los apartados y valoraciones que se tendrán en cuenta. El estudiante **deberá entregar este documento** cumplimentado **el día de la corrección**, indicando en la columna “E.” (referente a Estudiante) si el estudiante cumple con el requerimiento. El profesor validará si los cumple o no en la columna “P.”. **Para poder realizar la evaluación, el día de la corrección, el estudiante deberá desplegar en los laboratorios el escenario indicado en la especificación de la práctica.**

Cada estudiante debe ser capaz de explicar el código fuente y responder a las preguntas que el profesor haga a tal propósito (***Este requerimiento es indispensable para poder ser evaluado***).

Este escenario debe ser puesto en marcha delante del profesor lanzando cada uno de los procesos y componentes necesarios, coincidiendo con los pasos detallados en la memoria que cada estudiante debe entregar.

Concepto a evaluar	E.	P.
Despliegue, modularidad y escalabilidad (2 puntos)		
El sistema se despliega correctamente según se determina en la especificación de la práctica y se ha comentado en clase y sin necesidad de usar los entornos de compilación para su corrección . Ver nota al final del documento.		
Es posible desplegar tantas instancias del mismo módulo como se requiera en distintas máquinas o en la misma a criterio del profesor. Así, por ejemplo, se podrán desplegar tantos CP o Drivers como el profesor solicite. Así mismo, en cualquier momento durante la corrección, se puede requerir desplegar una instancia nueva o parar otra súbitamente simulando un “crash” de un determinado módulo.		
Funcionamiento base (hasta 3 puntos)		
Se pueden parametrizar distintos aspectos de la solución en cada uno de los módulos evitando que dichos parámetros se encuentren definidos de forma fija en el código. Así será posible parametrizar los puertos de servidores o IP o puertos a las que cada módulo debe conectarse, cualquier otro parámetro que el estudiante considere necesario para la ejecución de su solución. El código no podrá volver a compilarse en ningún momento de corrección de la práctica ante el cambio de cualquier de los parámetros de ejecución.		
La aplicación no falla durante el transcurso normal de una ejecución y cumple con la funcionalidad prevista. El módulo CENTRAL arrancará y estará		

<p>preparado para atender cuantos CP o Drivers soliciten sus distintas actividades. El archivo de servicios contendrá al menos 10 servicios para que se puedan hacer distintas pruebas durante la corrección sin interrupciones. Todo el sistema se inicia y se desarrolla sin incidencias. En este punto NO hay interacción ninguna del alumno ni profesor. Simplemente se ejecuta toda la solución con sus distintos módulos y mediante la observación de lo que se muestra en las distintas terminales de CENTRAL, Drivers y CP se puede validar perfectamente lo que ocurre en todo el sistema.</p>		
<p>Todos los módulos implementan las funcionalidades especificadas en la práctica: autenticación con aceptación o denegación de un CP o de un servicio a un Driver, envío de mensajes desde la central a los CP para suministrar o parar, etc. Los efectos sobre el ecosistema de todas las acciones se pueden observar en pantalla sin dificultad con un interfaz adecuado.</p>		
<p>Los servidores son todos concurrentes y están disponibles para recibir peticiones en todo momento.</p>		
<p>Resiliencia (hasta 3 puntos) (Solo evaluable si el funcionamiento base es correcto)</p>		
<p>Cualquier fallo en cualquier componente solo invalida el servicio proporcionado por ese componente. El resto de los componentes del sistema pueden seguir con su operativa normal salvo en lo que se vea afectado por el componente caído. Así, por ejemplo, si un taxi se desconecta en cualquier momento o deja de funcionar, el resto del sistema debe seguir funcionando correctamente. Esto aplica a cualquier componente.</p> <p>Casos específicos (no exhaustivos) a controlar en los que se paran o desaparecen abruptamente (Ctrl+C o cierre súbito de la terminal o ventana donde se ejecuta) los siguientes módulos:</p> <ul style="list-style-type: none"> - Un Monitor de un CP: La CENTRAL marcará el CP como "Desconectado". Si hubiera un suministro en marcha este finalizará salvo que el CP_E (Engine) también hubiera caído. El CP no admitirá más suministros ni siquiera manualmente. <i>Nota: En un sistema real, siempre que el sistema de pago esté funcionando, el CP podría seguir atendiendo manualmente a los Drivers.</i> - Un CP_Engine: El monitor (si no está caído) enviará una señal de avería a Central la cual procederá a marcarlo como averiado y, en su caso, enviará un mensaje al Driver al que se le estuviera prestando el servicio. El CP no puede perder lo que ya se haya suministrado y, cuando se recupere (se resuelva la avería), debe ser capaz de enviar el estado final del suministro realizado. - Un Driver se cierra mientras se le está prestando un servicio: El servicio sigue su curso. Cuando el cliente se recupera verá el resultado de su servicio. - La Central: Los CP siguen prestando su servicio hasta que lo finalicen momento en el cual se paran si la central no se ha restaurado. No será posible admitir nuevas peticiones de servicios. <i>Nota: En un sistema real, siempre que el sistema de pago esté funcionando, el CP podría seguir atendiendo manualmente a los Drivers, acumulando las recargas para luego enviarlas a la central una vez recuperada.</i> 		

<p>Nota: El Monitor es el encargado de decirle a Central el estado del CP. Para mayor claridad la tabla de estados del Monitor y Engine sería:</p> <p>Monitor_OK and Engine_OK => Activado (Verde)</p> <p>Monitor_OK and Engine_KO (Caído) => Averiado (Rojo)</p> <p>Monitor_KO (caído) and Engine_OK => Desconectado</p> <p>Monitor_KO (caído) and Engine_KO (Caído) => Desconectado</p> <p>El estado desconectado se produce cuando Central NO recibe ningún mensaje de vida desde Monitor.</p>		
<p>El sistema se recupera de forma correcta cuando se restituye el servicio de cualquier componente caído debiendo reiniciarse el mínimo número de módulos el sistema. La inclusión de nuevas instancias de cada módulo (nuevos Drivers, nuevos CP) no produce fallos.</p>		
<p>Se capturan los errores derivados de cualquier caída y se muestran al usuario de forma controlada en todos los interfaces donde aplique. Ej.: Si la CENTRAL está fuera de servicio o cae durante una partida, los Drivers o el CP visualizará un error del tipo "Imposible conectar con la CENTRAL".</p>		
General (hasta 2 puntos)		
<p>Entrega de la memoria (Informe de desarrollo, detalle de despliegue, resultados, formato y corrección).</p>		
<p>Diseño, confort, interfaces, acabado profesional.</p>		
<p>Otros aspectos técnicos reseñables por el estudiante. Ejemplos:</p> <ul style="list-style-type: none"> - Establecimiento de mecanismos adicionales de resiliencia como dos centrales o dos brokers) - Se ha implementado correctamente el protocolo de intercambio de mensajes en los sockets basado en tramas bien formadas <p><stx>D<etx><lrc>.</p>		

Nota: Es obligatorio desplegar todo el escenario, no se corregirán componentes independientes. LA PRÁCTICA NO ES EVALUABLE si no se logra desplegar el escenario mínimo en, al menos, 3 computadoras distintas (físicas o virtuales), con todos los componentes conectados e interactuando adecuadamente.

Las opciones de despliegue y su correspondiente valoración será el siguiente:

1- En máquinas del laboratorio o accediendo a una instalación remota de algún componente (ej. Kafka y/o Central) y despliegue del resto en el laboratorio. Opción mejor valorada.

2- En una red con portátiles propios: En caso de que la opción anterior no funcione se acepta la creación de una red con portátiles pero siempre con el criterio de despliegue distribuido. Sería menos valorado (a criterio del profesor puede restar algún punto) pero no impide su presentación ni buena evaluación.

3- En un portátil con máquinas virtuales (o dockerizando) en diferentes redes dentro de un solo equipo.

4- En un portátil en modo local host: Será evaluable, pero con una penalización importante (a criterio del profesor).

En todos los casos se puede y será valorado el uso dockers realizando diferentes contenedores con las distintas aplicaciones separadas para distribuirlas en las máquinas.

Importante: Un despliegue incorrecto puede penalizar el resto de los apartados de la corrección.