

Relación de entregas para seminarios 2019

(versión final diciembre 2019)

| Seminario 2 | NodeJs – Módulo Carrito de la compra |
|-----------------------|---|
| Recursos | POLIFORMAT Sad > Recursos / Seminars / Sem02 Node Sesión 1 / <ul style="list-style-type: none"> Ejemplo de acceso a mongodb: Cliente_mongo.js Máquina vagrant con mongodb instalado |
| Actividades | <p><i>Actividad 1:</i></p> <p>implementar la funcionalidad de un carrito de la compra en modulo de NodeJs. Dicho módulo permitirá realizar operaciones básicas: añadir producto a carro, quitar producto de carro. Dichas operaciones deben ser planteadas como operaciones asincrónicas.</p> |
| NIVEL OPCIONAL | <p><i>Actividad 2 (nivel opcional):</i></p> <p>diseñar otro módulo que sea capaz de gestionar una operación de comprobación de stocks de un producto añadido a un carrito de la compra. De tal manera que al añadir al carrito de la compra se comprueba mediante una operación de consulta de un servicio con mongodb instalado, si hay o no hay stock. Si no hay produce un error que debe salir por consola:</p> |

| Seminario 3 | NodeJs – Servicio REST carrito de la compra |
|-----------------------|--|
| Recursos | POLIFORMAT Sad > Recursos / Seminars / Sem02 Node Sesión 1 / <ul style="list-style-type: none"> Ejemplo de servicio : Service_Registry Estructura del proyecto para un servicio REST (vacío) |
| Actividades | <p><i>Actividad 1:</i></p> <p>Usando Express, tal y como se utiliza en los ejemplos mostrados, convertir el módulo del carrito de la sesión anterior en un servicio REST.</p> |
| NIVEL OPCIONAL | <p><i>Actividad 2 (nivel opcional):</i></p> <p>Incorporar a la aplicación distribuida el Service_Registry mostrado como ejemplo. Por tanto el servicio para el carrito de la compra deberá registrarse en dicho servicio ofrecido por Service_Registry (y borrarse en caso de una finalización controlada del Servicio carrito)</p> <p><i>Actividad 3(nivel opcional):</i></p> <p>Incorporar la capa de datos, el servicio de acceso a mongodb, como otro componente (microservicio) más de la aplicación. Registrarlo en el servicio Service_Registry</p> |

| | |
|--|--|
| | |
|--|--|

| Seminario 4 | Express y Websockets |
|-----------------------|---|
| Recursos | POLIFORMAT Sad > Recursos / Seminars / Sem04 Express y Websockets / 1. Websockets 2. Código de aplicación chat |
| Actividades | <p><i>Actividad 1:</i></p> <p>Se proporciona un código de un chat implementado de manera básica mediante websockets. Entender y poner en funcionamiento este código</p> <p><i>Actividad 2:</i></p> <p>Realizar sobre el código las siguientes modificaciones:</p> <ul style="list-style-type: none"> • Broadcast a message to connected users when someone connects or disconnects • Add support for nicknames • Don't send the same message to the user that sent it himself. Instead, append the message directly as soon as he presses enter. |
| NIVEL OPCIONAL | <p><i>Actividad 3 (nivel opcional):</i></p> <p>Realizar sobre el código las siguientes modificaciones:</p> <ul style="list-style-type: none"> • Add "{user} is typing" functionality • Show who's online • Add private messaging |

| Seminario 5 | Promesas |
|--------------------|---|
| Recursos | POLIFORMAT Sad > Recursos / Seminars / Sem05 Promesas / 1. Ejemplo de acceso a mongodb: Cliente_mongo.js 2. Código de módulo User.js |
| Actividades | <p><i>Actividad 1:</i></p> <p>Se proporciona un código de un módulo User.js. Dentro de este módulo, cambiar la función login del módulo User para que utilice promesas en lugar de callbacks. En la medida de lo posible, cambiar también todas las funciones asincrónicas que usa esta función a promesas.</p> |

| | |
|--|---|
| | <p><i>Actividad 2:</i></p> <p>En el carrito de la compra anteriormente desarrollado, realizar un módulo de acceso a la base de datos implementada con mongodb. Este módulo realizará la comprobación de stock al añadir productos al carrito, Usar promesas para realizar las operaciones asincrónicas en lugar de los callbacks, cuando sea posible.</p> |
|--|---|

| Seminario 6 | ZeroMQ |
|-----------------------|--|
| Recursos | POLIFORMAT Sad > Recursos / Seminars / Sem06 ZeroMQ / 1. Explicación del mecanismo de circuit breaker |
| Actividades | <p><i>Actividad 1:</i></p> <p>Implementar el carrito de la compra como componente comunicado por Zeromq. Cada carrito realiza las mismas operaciones pero las peticiones llegan por medio de mensajes usando el middleware ZeroMQ. Desarrollar un broker (con los sockets Zeromq adecuados). Este broker recibirá como mensaje las peticiones de introducir o eliminar un producto. Si es la primera operación que se pide (es decir, si la petición se origina desde un componente con un identificador que es nuevo para el broker) se le asignará un carrito. El broker debe tener la capacidad de registrar los carritos y decida a cuál de ellos (que no tenga ya una asignación) asignar una petición (balanceado de carga). También debe llevar la cuenta de qué componente (cliente) usa que carrito, hasta que el componente que origina la petición decide cerrar el uso del carrito (operación close o finish). En ese momento se elimina al cliente que usa el carrito y se lleva al carrito al estado vacío y listo para usar.</p> <p>¿Podría usarse el broker implementado con ZeroMQ y los carritos con REST y conectarse de alguna manera.? ¿Proporciona esta solución alguna ventaja?</p> |
| NIVEL OPCIONAL | <p><i>Actividad 2:</i></p> <p>Proporcionar un mecanismo (usando Zeromq) que permita determinar si el carrito continua funcionando con normalidad. En caso contrario usar el mecanismo del circuit breaker (proxy de estado fallido) para evitar que se produzca un error en cascada que colapse el sistema.</p> |

| Seminario8 | Consistencia (I) |
|--------------------|---|
| Recursos | POLIFORMAT Sad > Recursos / Seminars / Sem08 Consistencia / 1. sad-se08-consistencia.pdf |
| Actividades | <i>Actividad 1</i> |

Dadas las siguientes ejecuciones, justifique qué modelos de consistencia respeta cada una:

1. W1(x)1, R2(x)1, W3(x)2, R4(x)1, R3(x)1, W4(x)3, R2(x)3, R1(x)2, R3(x)3, R2(x)2, R1(x)3, R4(x)2.
2. W1(x)1, W2(y)3, W1(y)2, W3(x)2, R2(x)2, R4(x)2, R3(y)2, R1(y)3, R2(x)1, R4(y)2, R3(y)3, R4(x)1, R3(x)1, R4(y)3.

Actividad 2:

Dadas las siguientes ejecuciones, justifique qué modelos de consistencia respeta cada una:

1. W1(x)3, W2(x)1, W1(y)2, R3(x)3, R2(x)3, R3(x)1, R2(y)2, R3(y)2.
2. W1(x)3, R2(x)3, W1(y)2, W2(x)1, R3(x)1, R1(x)1, R3(x)3, R2(y)2, R3(y)2.

Actividad 3

Dadas las siguientes ejecuciones, justifique qué modelos de consistencia respeta cada una:

1. W1(x)1, R2(x)1, R3(x)1, W2(x)2, R1(x)2, R4(x)2, W4(x)3, R1(x)3, R2(x)3, R3(x)3.
2. W1(x)1, R2(x)1, R3(x)1, W2(x)2, R4(x)2, R4(x)1, R1(x)2, W2(x)3, R3(x)2, R4(x)3, R1(x)3, R3(x)3.

Actividad 4

Dadas las siguientes ejecuciones, justifique qué modelos de consistencia respeta cada una:

1. W1(x)1, R2(x)1, W2(y)3, W3(x)4, R3(y)3, R4(x)4, R4(y)3, R1(y)3, R4(x)1, R3(x)1
2. W1(x)1, R2(x)1, R3(x)1, W2(x)2, R1(x)2, R4(x)1, W4(x)3, R1(x)3, R2(x)3, R3(x)2, R3(x)3.

Actividad5

Proporcione un ejemplo de ejecución donde intervengan al menos cuatro procesos y cumplan con los requisitos siguientes. En caso de que no se puedan cumplir todas las condiciones exigidas en un mismo apartado, justifique por qué.

1. Se cumpla la consistencia caché pero no se cumpla la consistencia FIFO
2. Se cumpla la consistencia FIFO pero no se cumpla la consistencia caché

Actividad 6

Proporcione un ejemplo de ejecución donde intervengan al menos cuatro procesos y cumplan con los requisitos siguientes. En caso de que no se puedan cumplir todas las condiciones exigidas en un mismo apartado, justifique por qué.

1. Se cumpla la consistencia procesador pero no se cumpla la consistencia causal

| | |
|--|---|
| | 2. Se cumpla la consistencia causal pero no se cumpla la consistencia procesador |
| | <p><i>Actividad 7</i></p> <p>Proporcione un ejemplo de ejecución donde intervengan al menos cuatro procesos y cumplan con los requisitos siguientes. En caso de que no se puedan cumplir todas las condiciones exigidas en un mismo apartado, justifique por qué.</p> <p>1. Se cumpla la consistencia secuencial pero no se cumpla la consistencia caché</p> <p>2. Se cumpla la consistencia caché pero no se cumpla la consistencia secuencial</p> |

| | |
|-----------------------|---|
| Seminario 9 | Seguridad en sistemas distribuidos |
| Recursos | POLIFORMAT Sad > Recursos / Seminars / Sem 09 Seguridad en sistemas distribuidos / 1. Presentación (tareas constan al final del mismo) |
| NIVEL OPCIONAL | <p><i>Actividad 1:</i></p> <p>(se puede realizar cualquier de las siguientes o varias)</p> <ul style="list-style-type: none"> • Plantearse como desarrolladores del ejercicio de prácticas todas las cuestiones examinadas durante el seminario • Considerar las comunicaciones con ZMQ, ¿son seguras? Se puede suplantar la identidad de algunos de los componentes, pueden interceptarse. • ¿Qué acciones se pueden tomar para mejorar la seguridad del desarrollo? ¿Hasta qué punto son necesarias? • Escoger un incidente de seguridad recientemente en un servicio en la nube y analizar sus causas. |

| | |
|-----------------------|---|
| Seminario 10 | Blockchain |
| Recursos | POLIFORMAT Sad > Recursos / Seminars / Sem10 Tecnología blockchain / 1. Blockchain |
| NIVEL OPCIONAL | <p><i>Actividad 1:</i></p> <p>Realizar un componente que permita almacenar información que se necesita en un foro de mensajes (con la misma interfaz del trabajo de laboratorio). Dicho componente deberá usar la tecnología blockchain a través de un componente instalado en nuestra aplicación distribuida capaz de conectarse a una</p> |

| | |
|--|---|
| | <p>red blockchain y publicar un mensaje de un usuario. También debe ser capaz de recuperar información a petición del cliente. Para simplificarlo sólo serán necesarios mensajes de un usuario que van destinados a un tema o a otro usuario (contenido del mensaje y destino).</p> <p>Para ello, apoyarse en la herramienta Multichain y su API JSON, y algún módulo node (buscar en www.npmjs.com) que facilite su utilización.</p> <p>El resultado debe ser tal que puedan funcionar varios componentes conectados a distintos nodos de la red blockchain, y proporcionar soporte para los datos.</p> |
|--|---|

| | |
|-----------------------|--|
| Seminario 11 | Despliegue de servicios |
| Recursos | POLIFORMAT Sad > Recursos / Seminars / Sem11 Despliegue de servicios / |
| NIVEL OPCIONAL | <p><i>Actividad 1:</i></p> <p>Desarrollar la estructura necesaria (carpetas y archivos) para el despliegue de los ejercicios realizados en el laboratorio, usando docker y docker-compose.</p> |