

**Relación de entregas para seminarios**  
**Versión final** (actualizado 21 diciembre 2018)

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

|                    |  |
|--------------------|--|
| <b>Seminario 5</b> | <b>ZMQ</b>   |
| <b>Recursos</b>    | POLIFORMAT Sad > Recursos / Seminars / <a href="#">Sem 5 ZMQ</a> /<br><ol style="list-style-type: none"> <li>1. <a href="#">Tarea_sem5.pdf</a></li> <li>2. <a href="#">ZGuide Chapter 8 - A Framework for Distributed Computing</a></li> </ol>   |
| <b>Actividades</b> | <p><i>Actividad 1:</i></p> <p>Proponer una solución, usando ZMQ (o un módulo alternativo externo), para resolver el problema del descubrimiento inicial en un sistema con la siguiente estructura:</p> <pre> graph TD     subgraph Column1         C1[CLIENT] --&gt; B1[BROKER]         B1 --&gt; W1[WORKER]         W1 --&gt; B1     end     subgraph Column2         C2[CLIENT] --&gt; B2[BROKER]         B2 --&gt; W2[WORKER]         W2 --&gt; B2     end     subgraph Column3         C3[CLIENT] --&gt; B3[BROKER]         B3 --&gt; W3[WORKER]         W3 --&gt; B3     end     W1 --&gt; W2     W2 --&gt; W3     W3 --&gt; W2     W2 --&gt; W1 </pre> <p>Tener presente que en dicho sistema:</p> <p>Los clientes son externos, y generan peticiones de modificación que llegan hasta los workers. Los workers usan un sistema de propagación con sockets PUB-SUB para que esa información llegue a todos los clientes. (Se da por asumida la implementación de este comportamiento).</p> <p>El problema del descubrimiento inicial dinámico intenta abordar el problema de iniciar la ejecución de un sistema distribuido, donde se deben poner en contacto a los distintos componentes. Se trata de que nuestro sistema (brokers y workers) se pongan en ejecución y tengan un medio de contactar unos con otros de la manera mas flexible posible. Esto significa que si durante la ejecución uno de los nodos cae y el agente del sistema distribuido necesita reiniciarse en otro nodo, también se pueda resolver sin necesidad de reiniciar todo el conjunto de agentes.</p> <p>Se trata de proponer una solución detallada a este problema, usando ZMQ o algún</p> |

|                |  |
|----------------|--|
|                | módulo equivalente. También deben analizarse los problemas que puedan surgir en la solución propuesta. |
| NIVEL OPCIONAL | <i>Actividad 2: Implementar la solución propuesta</i>  |

|                    |  |
|--------------------|--|
| Seminario 6-7      | <b>Promises</b>  |
| <b>Recursos</b>    | POLIFORMAT Sad > Recursos / Seminars / Sem06 - 07 Promesas / <ol style="list-style-type: none"> <li>1. <a href="#">sad-se07-actividad1.pdf</a></li> <li>2. <a href="#">actividad2.pdf</a></li> <li>3. <a href="#">Cliente_mongo.js</a></li> <li>4. <a href="#">dmclient.js</a></li> <li>5. <a href="#">dmserver.js</a></li> </ol>  |
| <b>Actividades</b> | <p><i>Actividad 1 (<a href="#">sad-se07-actividad1.pdf</a>):</i><br/> <i>(....)</i><br/> <i>Modifique el programa anterior, utilizando promesas, para garantizar que se respete el orden de ejecución esperado</i></p> <p><i>Actividad 2 (<a href="#">actividad2.pdf</a>)</i></p> <p>Diseñar e implementar un módulo en javascript que permita a un servidor https la funcionalidad de un <b>carrito de la compra</b> (sólo la gestión, no la interfaz). Dicho módulo permitirá realizar operaciones básicas: añadir producto a carro, quitar producto de carro.</p> <p>A continuación 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 (No como una excepción).</p> <p><i>Cambiar la implementación para realizar desde el servidor https toda la operación (añadir un producto al carrito si hay stock en la base de datos) en forma de promesa</i></p> |

|                    |   |
|--------------------|---|
| <b>Seminario8</b>  | <b>Consistencia (I)</b>   |
| <b>Recursos</b>    | POLIFORMAT Sad > Recursos / Seminars / <a href="#">Sem08 - 09 Consistencia</a> /<br><br>1. <a href="#">sad-se08-consistencia.pdf</a>  |
| <b>Actividades</b> | <p><i>Actividad 1</i><br/> Dadas las siguientes ejecuciones, justifique qué modelos de consistencia respeta cada una:</p> <p>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.<br/> 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.</p> <p><i>Actividad 2:</i><br/> Dadas las siguientes ejecuciones, justifique qué modelos de consistencia respeta cada una:</p> <p>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.<br/> 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.</p> <p><i>Actividad 3</i><br/> Dadas las siguientes ejecuciones, justifique qué modelos de consistencia respeta cada una:</p> <p>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.<br/> 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.</p> <p><i>Actividad 4</i><br/> Dadas las siguientes ejecuciones, justifique qué modelos de consistencia respeta cada una:</p> <p>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<br/> 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.</p> |

|  |  |
|--|--|
|  | <p><i>Actividad 5</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> <ol style="list-style-type: none"> <li>1. Se cumpla la consistencia caché pero no se cumpla la consistencia FIFO</li> <li>2. Se cumpla la consistencia FIFO pero no se cumpla la consistencia caché</li> </ol><br><p><i>Actividad 6</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> <ol style="list-style-type: none"> <li>1. Se cumpla la consistencia procesador pero no se cumpla la consistencia causal</li> <li>2. Se cumpla la consistencia causal pero no se cumpla la consistencia procesador</li> </ol> |
|  | <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> <ol style="list-style-type: none"> <li>1. Se cumpla la consistencia secuencial pero no se cumpla la consistencia caché</li> <li>2. Se cumpla la consistencia caché pero no se cumpla la consistencia secuencial</li> </ol>   |

|                    |   |
|--------------------|---|
| <b>Seminario9</b>  | <b>Consistencia (II)</b>  |
| <b>Recursos</b>    | POLIFORMAT Sad > Recursos / Seminars / <a href="#">Sem08 - 09 Consistencia</a> /<br><br>1. <a href="#">sad-se09-consistencia.pdf</a>  |
| <b>Actividades</b> | <p><i>Actividad 1:</i></p> <p>Suponemos 3 procesos servidores "dmserver", llamados S1, S2 y S3. Modelamos el foro de mensajes con una única variable 'X' para todos los datos del foro. De forma que cualquier modificación de los datos supone una escritura, modelada como W(x) y cualquier lectura de los datos, supone R(x). Cada vez que un servidor actualice sus datos debido a que un servidor "forum" conectado a él le comunique cambios, ejecutaremos S?W(x) a. Siendo '?', 1, 2 ó 3. Cada vez que un servidor comunique sus datos a su servidor "forum" correspondiente, consideraremos que se ejecuta S?R(x) a. Razonar qué modelos de consistencia se soportan en nuestro sistema, suponiendo que hemos implementado el paso de mensajes entre servidores mediante el modelo Pub/Sub.</p> <p><i>Actividad 2:</i></p> <p>Suponemos 3 procesos servidores "dmserver", llamados S1, S2 y S3 y modelamos 3 navegadores web, llamados N1, N2 y N3. Cada navegador está conectado a un servidor web "forum" que a su vez está conectado a su correspondiente servidor "dmserver". De esta forma, modelamos el sistema haciendo que N1 esté conectado a S1, N2 conectado a S2 y N3 conectado a S3. SAD: Seminario 9 Página 2. Modelamos el foro de mensajes con una única variable 'X' para todos los datos del foro. De forma que cualquier modificación de los datos supone una escritura, modelada como W(x) y cualquier lectura como R(X), pero en este caso nos interesa únicamente los accesos de tipo N?W(x) y N?R(x), donde ? es 1, 2, 3. Se pide razonar qué modelos de consistencia se soportan, teniendo en cuenta que únicamente nos interesa describir el comportamiento de los navegadores. Es decir, queremos observar la consistencia de los datos únicamente entre los 3 nodos cliente finales.</p> <p><i>Actividad 3:</i></p> <p>Consideramos el mismo modelo que en la actividad 1, pero utilizando 3 variables, X, Y y Z. La variable X para representar la lista de usuarios del foro, la variable Y para representar los mensajes privados del foro y la variable Z para los mensajes del foro público. Razonar qué modelos de consistencia se soportan.</p> <p><i>Actividad 4</i></p> <p>Ampliando el análisis efectuado en la actividad 3, razonar qué ventajas podría suponer considerar una variable X para la lista de usuarios, y una variable diferente para cada uno de los temas del foro y para cada una de las conversaciones privadas del foro. Argumentar qué cambios podríamos hacer en la implementación para implementar eficientemente los diferentes modelos de consistencia, y si estas modificaciones tendrían sentido práctico.</p> |

|                           |   |
|---------------------------|---|
| <b>Seminario<br/>10</b>   | <b>Blockchain</b>   |
| <b>Recursos</b>           | POLIFORMAT Sad > Recursos / Seminars / <a href="#">Sem10 Tecnología blockchain</a> /<br>1. <a href="#">Blockchain</a>   |
| <b>NIVEL<br/>OPCIONAL</b> | <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).<br/> Dicho componente deberá usar la tecnología blockchain a través de un componente instalado en nuestra aplicación distribuida capaz de conectarse a una 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).<br/> Para ello, apoyarse en la herramienta Multichain y su API JSON, y algún módulo node (buscar en <a href="http://www.npmjs.com">www.npmjs.com</a>) que facilite su utilización.<br/> 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> |