

SDI 24810 - Sistemas de Comercio Electrónico

Semestre agosto – diciembre de 2020

Examen Final

Viernes 18 de diciembre, 16:00 hrs

Duración: 2:45 hrs

CU: 000163314

Nombre: José Francisco Altamirano Zevallos

Al finalizar su examen o cuando el profesor lo solicite (al final de la duración) **genere un pdf** y súbalo a CANVAS.

Responda las preguntas:

Preg 1).- ¿Cuál es el objetivo de estresar las aplicaciones multiusuario?

El objetivo es probar la concurrencia del sistema; es decir, ver que tantos usuarios pueden utilizar el sistema al mismo tiempo sin que se presenten fallas en el servidor o que este se caiga.

Preg 2).-Mencione dos características que debe contemplar el proceso de estresamiento de aplicaciones multiusuario y comente brevemente el porqué de tales características.

Debe de tomar en cuenta la cantidad de usuarios con los que se va a estresar y la cantidad de transacciones que cada usuario va a realizar. Esto se debe a que, de esta manera, se puede llevar un buen control sobre el estresamiento y sus estadísticas, viendo cosas como la media del ciclo, su desviación estándar, y sus tiempos máximos y mínimos. Observando como se ven afectadas estas características según lo que se va cambiando.

Preg 3).- ¿Cuál es la diferencia entre las pruebas para los requerimientos funcionales y para los requerimientos no funcionales?

Las pruebas de los requerimientos funcionales están orientadas hacia la lógica de la aplicación; es decir, deben de enfocarse en cosas como que se evite vender cosas que no están disponibles. Las pruebas de requerimientos no funcionales están orientadas hacia el funcionamiento de la aplicación; es decir, deben de enfocarse en cosas como la cantidad de transacciones que se deben de soportar, tiempos de respuesta, etc.

Preg 4).-

i) ¿Cómo se llevan a cabo las Pruebas Funcionales?

Para llevar a cabo una prueba funcional, es necesario contemplar todos los casos a los que se puede ver expuesto un método y ver que su funcionamiento sea correcto en cada uno de los casos; un ejemplo de como hacerlo es utilizar una aplicación de Java que llame al método de un webservice y pruebe a ejecutarlo con los diferentes parámetros a los que se podría ver expuesto.

ii) ¿Cómo podemos garantizar que sean repetibles y para qué?

Para garantizar que una prueba sea repetible es necesario definir un steady state; una vez que se haya definido, se realiza la prueba sobre este estado y se analiza y anota el resultado; finalmente, al regresar al steady state y realizar otra vez la prueba, esta debe de comportarse de la misma manera. Es necesario definir esto para asegurar la integridad del sistema y su confiabilidad.

Preg 5).-¿Qué es la inyección de dependencias? Mencione tres lugares donde las empleó en su proyecto.

La inyección de dependencias es cuando se pasa un objeto de una clase a otra; esto sirve para que una clase pueda crear un objeto según lo definido por la otra y evite tener que hacerlo dentro de si misma, esto le da modularidad al proyecto. En el proyecto lo usamos al llamar a los web services mediante External WSDL desde el BPEL, al llamar al BPEL desde el CA (Composite Application) y al llamar el web service del CA desde el POJO (Pure Old Java Object).

Preg 6).- Describa brevemente qué es una Arquitectura SOA.

La arquitectura SOA es aquella en la que los servicios son proveídos por una aplicación a través de un protocolo de comunicación. En el caso del proyecto, el servicio de comprar un libro esta proveído por nuestra aplicación diseñada en BPEL y hosteada en CA, siendo que su protocolo de comunicación es SOAP.

Preg 7).- i) ¿Qué es un servidor de aplicación?

Un servidor de aplicación es un servidor en el cual están hospedadas múltiples aplicaciones, siendo que para acceder a los servicios ofrecidos por estas aplicaciones es necesario que el cliente realice una conexión con el servidor, pueden conectarse múltiples clientes a un solo servidor.

ii) ¿Porqué se controlan / administran desde un página web?

El hecho de que se controle un servidor desde una página web hace sea mucho más fácil de entender y utilizar al contar con una interfaz gráfica, lo que le da una mayor accesibilidad; además, al encontrarse en un sitio web, permite que cuente con una mayor seguridad al no ser tan vulnerable a una falla local, lo que permite que lo puedas acceder en todo momento desde cualquier lugar.

iii) ¿Se pueden controlar / administrar por otro medio?

Si, se puede administrar utilizando algún programa local o una terminal; sin embargo, no cuenta con las ventajas que presenta hacerlo mediante una página web.

Preg 8).- La aplicación que se desarrolló es una aplicación “de juguete”. Mencione qué parte no es para nada realista.

Hay múltiples partes que no son realistas en esta aplicación, la primera de ellas es que los servidores de Glassfish y OpenESB están hospedados de manera local y no tienen un domain verdadero en internet, por lo que es un servidor “local”; otras cosas que son poco realistas se encuentran en el funcionamiento de la aplicación, como, por ejemplo, que el sistema de mensajería invente una compañía de entrega y una fecha basándose en número al azar.

Preg 9).- Siguiendo con la pregunta anterior,

i) ¿Qué requiere para ser realista?

Sería importante buscar la manera en la que podamos cambiar el alcance del servidor de local a global, para que de esta manera funcione como un verdadero servidor de aplicaciones; además, sería importante ajustar el funcionamiento de la aplicación para que utilice datos verdaderos y no inventados.

ii) ¿Cómo lo lograría?

Para hacer esto, sería necesario rentar un servidor de los que existen en línea, para posteriormente pasar a instalar y configurar las cosas para su correcto funcionamiento, como, por ejemplo, los componentes, librerías, etc. Además, para que la aplicación utilice datos verdaderos, debería de implementarse en un negocio real, y modificarse según las necesidades de dicho negocio.

Preg 10).- ¿Cuál es el papel del ESB en la arquitectura utilizada en la aplicación desarrollada en el proyecto Final?

El ESB en nuestro proyecto funcionará como nuestro servidor de aplicaciones, es en este donde se guardará como service assembly la composite application que realizamos en el OpenESB Studio.

Preg 11).- ¿Cuál es el papel del BPEL?

En el BPEL es donde se definirá la lógica de negocio que seguirá la aplicación; es decir, que es lo que va a recibir, cuales son los pasos que va a seguir el programa para su ejecución y finalmente que es lo que va a entregar. En el caso de nuestro proyecto, especificamos mediante WSDL los inputs y outputs de la aplicación y mediante External WSDL documents, los métodos junto con sus inputs y outputs que utilizará la aplicación.

Preg 12).- ¿Para qué sirve el proyecto “Composite Application”?

El composite application es la manera en la que el proyecto realizado en el BPEL es construido como aplicación de manera que pueda ser subido como un Service Assembly al servidor de aplicaciones; además, también sirve para realizar pruebas mediante mensajes SOAP del BPEL antes de pasarlo al servidor de aplicaciones.

Preg 13).- ¿Porqué utilizamos “POJOs” para probar nuestra aplicación?

Utilizamos POJOS para probar nuestra aplicación debido a que estos pueden recibir inyecciones de dependencias, en nuestro caso llaman al Service Assembly que se encuentra en el servidor de aplicaciones mediante un WebService. De esta manera tenemos acceso a nuestra aplicación.

Preg 14).- ¿Qué papel tiene el “Logger” en el desarrollo / depuración de nuestras aplicaciones?

El logger es un componente de los servidores que nos ayuda a saber, mediante el texto impreso en el output del servidor, los procesos que están realizándose en el servidor. Este nos ayuda a saber si hay algún problema, y en caso de haberlo, cual problema es y donde se encuentra localizado.

Preg 15).-

- i) ¿Qué es la ejecución dinámica de código?

La ejecución dinámica de código es cuando se nos permite cambiar el código mientras este se esta ejecutando, contrario a cuando es necesario detener la ejecución, cambiarlo y luego ejecutarlo de nuevo.

- ii) ¿Para qué la utilizamos?

Esto es útil para poder realizar cambios directamente sobre la aplicación sin afectar a su uptime; asimismo, lo utilizamos para corregir errores según los vamos detectando y analizar cuál es la reacción de la aplicación al corregirlos.

Preg 16).-

- i) ¿Qué significa que una porción de código sea “static” y cuál es su implicación en términos de servicios concurrentes?

El código estático es aquel que pertenece a todas las instancias existentes de una clase, por lo que si es modificado, afecta a todas las instancias que utilicen este mismo código; en el caso de un servicio concurrente que utilice código estático, un cambio sobre este afectará a todos los clientes.

- ii) ¿Qué significa que un servicio sea “trade safe”?

Para que un servicio sea trade safe, es necesario que toda transacción respete la integridad de la base de datos; es decir, el servicio debe de ser capaz de asignar recursos a cada usuario hasta terminar de usarlos, de esta manera se puede evitar que se de el caso en el que dos usuarios quieran al mismo tiempo 10 unidades de un producto del cual solo quedan 10 unidades y el servicio permita a los dos realizar la transacción, resultando esto en que se termina con una existencia de -10 unidades.

Preg 17).-

- i) ¿Cómo identificamos que existen “cuellos de botella” con el proceso de estresamiento?

Los cuellos de botella son identificables en el proceso de estresamiento cuando se realizan menos transacciones que las que se especifica que se hacen, también sucede que, si es muy severo el cuello de botella, haya menos usuarios de los que se supone que se están ejecutando; esto se puede notar al solicitar las estadísticas finales en el proceso de estresamiento. Otra manera de notarlo es analizando los recursos con los que cuenta el servidor; por ejemplo, en el caso del proyecto, al ver que tanto la RAM y el CPU estaban en su punto máximo, se puede notar que hay un cuello de botella.

- ii) ¿Cómo identificamos si es un problema de los servicios (piense en la Base de Datos) o en algún lugar de la infraestructura tecnológica utilizada?

Esto se puede analizar al consultar la configuración tanto de la base de datos, como de la infraestructura utilizada y ver si son compatibles; además, cambiando los parámetros del estresamiento, también se puede probar con diferentes configuraciones para ver como responde la aplicación.

Preg 18).- ¿Cuántas tecnologías para máquinas virtuales conoce y cuáles son?

Conozco que las tecnologías de máquinas virtuales se dividen en las máquinas virtuales de sistema vs las de proceso.

Siendo que las de sistema son aquellas que emulan una computadora con su sistema operativo y son utilizables como una computadora virtual; y las de proceso son las que emulan el sistema operativo, pero solamente para ejecutar un proceso.

Para poder usar una máquina virtual de sistema existen aplicaciones como VMWare y Oracle VM VirtualBox, mientras que para las de proceso, se usa cualquier aplicación capaz de crear un Managed Runtime Environment.

Preg 19).- Mencione dos protocolos de interoperabilidad con que puede trabajar con java y en qué difieren.

Un protocolo de interoperabilidad es SOAP (Simple Object Access Protocol) y otro es REST (Representational State Transfer), siendo que ambos tienen como objetivo permitir el intercambio de mensajes; la diferencia que tienen es que SOAP utiliza muchos recursos para mandar mensajes y permite su uso de manera síncrona y asíncrona, mientras que REST utiliza menos recursos para mandar mensajes (por lo que se suele utilizar para mensajes masivos), pero solo permite mensajes de tipo síncrono.

Preg 20).- En términos de interoperabilidad, ¿cuándo utilizar un protocolo síncrono o uno asíncrono?

Es importante utilizar un protocolo síncrono cuando la aplicación requiere que ambos haya un intercambio de información en tiempo real; un ejemplo de esto es una aplicación de chat. Se utiliza un protocolo asíncrono cuando no se tiene el requerimiento previamente mencionado, ya que utiliza muchos más recursos utilizar un protocolo síncrono, por lo tanto, se usa cuando no es necesario que el intercambio de información sea realizado en tiempo real; un ejemplo de esto es una aplicación de correos.