|  |
| --- |
| **Nombre Integrante 1: Andres Osorio**  **Nombre Integrante 2: Sebastian Cardona C**  **Nombre Integrante 3: Fabian Hernandez** |

Con respecto a la inspección del código fuente de ejemplo, responda las siguientes preguntas:

1. Ubique la plantilla *vendedores.xhtml*, revise las invocaciones que se realizan al managed bean *VendedorBean*, ubique la acción para agregar un vendedor ¿Cómo se realiza esta invocación?

Esta invocación se realiza a través del EJB de VendedorBean, instanciando el servicio local IServicioVendedoresMockLocal.

1. En la acción borrar un vendedor en la plantilla *vendedores.xhtml*, ¿Cómo se pasa el parámetro para identificar el vendedor que debe ser eliminado?

La grilla de datos obtiene los registros de los vendedores en el cual el botón eliminar, contiene el ID del vendeodr que se desea eliminar el cual viaja como parámetro a través de la petición, la cual puede ser obtenida en un objeto tipo Map (De tipo Key:Value), donde se extrae el valor del parámetro “VemddorId” para luego ser enviado al servicio en el método para la eliminación del vendedor.

1. Revise las interfaces *IServicioVendedoresMockLocal* e *IServicioVendedoresMockRemote* ¿Qué tipo de interfaces son?

La interfaz *IServicioVendedoresMockLocal* es de tipo interfas de negocio local (ejb.Local)

La interfaz *IServicioVendedoresMockRemote* es de tipo interfas de negocio remoto (ejb.Remote)

1. ~~¿Si usted cambia la anotación~~ *~~@Stateful~~* ~~a~~ *~~@Stateless~~* ~~del session bean~~ *~~ServicioVendedoresMock~~* ~~qué consecuencias habrían la aplicación? ¿Qué consecuencias genera la anotación~~ *~~@Singleton~~*~~?~~
   1. NOTA: ¿Si usted cambia la anotación *@Stateless* a *@Stateful* del session bean *ServicioVendedoresMock* qué consecuencias habrían la aplicación? ¿Qué consecuencias genera la anotación *@Singleton*?

Las consecuencias están relacionadas al ciclo de vida que tendrá la sesión, ya que al pasarla a Statefull, tendre la oportunidad de implementar las dependencias @PrePassive, @PostPassive del estado pasivo y @Remove

En el @stateful puedo tener más de una instancia de la sesión, mientras que en el @singleton solo una.

Una de los puntos en cuanto a rendimientos se refiere es que el @stateful mantiene el estado de conversaciones en memoria, dando la posibilidad de almacenar la sesión.

1. Revise las pruebas JUnit implementadas. En el proyecto Web, en el test del servicio de seguridad (*LoginBeanTest*), la inyección de la dependencia se hace por medio de la instanciación del *mock object* que la implementa. Por otro lado, en el proyecto EJB, la inyección de la dependencia en el test de *VendorServices* (*ServicioVendedoresMockTest*) se hace por medio de JNDI. ¿Cuál es la diferencia? Ejecute ambas pruebas dos veces, una con la aplicación desplegada en el servidor y otra sin dicho despliegue. ¿Qué puede concluir de dichas ejecuciones? Sea claro y concluyente.

Para descubrir un servicio desde un JUNIT se requiere de realizar una búsqueda en el directorio de la interfaces.