

NOTA Las respuestas brindadas en este documento son una guía para que el estudiante profundice en las respuestas. Bajo ningún concepto se deben considerar como respuestas completas a las preguntas planteadas.

Preguntas de Diseño Arquitectónico (70 puntos)

1. (10 puntos) Conceptos generales sobre Arquitectura de software

- a) Escriba la definición de Arquitectura de Software según Bass, Kazmann y otros. Si no recuerda la definición exacta puede hacerlo en sus propios términos.
- b) En la definición se refiere a “..... estructuras del sistema”. ¿A qué se refiere?

RESPUESTA

5 puntos por parte

a. Defina Arquitectura de Software según Bass, Kazmann y otros.

La arquitectura de un programa o sistema de computación es la estructura o estructuras del sistema, que comprenden sus componentes de software, las propiedades externas de los componentes, y la relación entre ellos” (Kazmann)

b. En la definición se refiere a “..... estructuras del sistema”. ¿A qué se refiere?
Las arquitecturas en general comprenden distintas estructuras (unidades lógicas, de implementación, de procesos). Se puede referir al modelo 4 +1 que representa varias de estas estructuras en las vistas.

2. Atributos de calidad: escenarios y tácticas

2.1 (12 puntos) Clasifique los siguientes escenarios en función del atributo de calidad al que corresponden indicando para cada uno: Fuente, Estimulo, Artefacto, Respuesta, Ambiente y Medida. En caso que alguno de los elementos no se expliciten infiéralo.

- *El grupo de software embebido debe poder reusar el componente de control del motor de cuatro cilindros sin modificación de código en los próximos motores de 6 cilindros que desarrolle la empresa*
- *En caso que una persona no autorizada obtenga acceso al sistema e intente modificar los datos de los clientes, el sistema debe registrar los eventos en el log de auditoria, notificar al administrador del sistema*

mediante un SMS y correo electrónico y bajar el servicio. Todas las acciones se deben realizar en un periodo de menos de 2 segundos.

- *El grupo de desarrollo debe poder utilizar el sistema de control de motores en procesadores de cuatro núcleos sin impacto en el código fuente del sistema.*

RESPUESTA

4 puntos por escenario

Atributo: Mantenibilidad

Fuente: El grupo de software embebido

Estimulo: reusar en los próximos motores de 6 cilindros que desarrolle la empresa

Artefacto: el componente de control del motor de cuatro cilindros

Ambiente: se infiere - en desarrollo o despliegue

Respuesta: sin modificación de código

Medida: se infiere sin costo

Atributo: Seguridad

Fuente: Persona no autorizada

Estimulo: obtiene acceso al sistema e intente modificar los datos de los clientes

Artefacto: el sistema

Ambiente: se infiere – en línea

Respuesta: registrar los eventos en el log de auditoria, notificar al administrador del sistema mediante un SMS y correo electrónico y bajar el servicio

Medida: en menos de dos segundos

Atributo: Mantenibilidad

Fuente: El grupo de desarrollo

Estimulo: utilizar en procesadores de cuatro núcleos

Artefacto: el sistema de control de motores

Ambiente: se infiere - en desarrollo

Respuesta: sin impacto en código fuente

Medida: se infiere - sin costo

2.2 (16 puntos) Dada la siguiente lista de acciones tomadas por un arquitecto durante el diseño de distintas partes de un sistema para cumplir con el conjunto de atributos de calidad que se lo presentaron. Identifique los atributos de calidad que le plantearon e identifique los nombres de las tácticas utilizadas.

- a) Estableció un mecanismo para que los datos que se consultaban frecuentemente se adquirieran tempranamente y se almacenaran en un medio de rápido acceso.
- b) Utilizó protocolos de comunicación para proteger los datos confidenciales no se vean expuestos.

- c) Utilizó como política general el principio de otorgar el menor número los privilegios a las cuentas de los usuarios.
- d) Se preocupó por definir interfaces y cuando fuera necesario agregar nuevas o utilizar adaptadores y stubs.
- e) Utilizó el patrón MVC para que se pudieran ofrecer distintas vistas sobre la información contemplando las distintas necesidades y habilidades de los usuarios.
- f) Utilizo un componente para distribuir los accesos a los servidores y evitar que los componentes de su aplicación se vean sobrecargados.
- g) Se preocupó de que los componentes de interfaz de usuario tuvierán la capacidad de anticipar lo que el usuario fuera a realizar.
- h) Diseño para los distintos componentes una interfaz que permitieran acceder a información sobre el estado del componente, su carga, etc. Utilizó programación orientada a aspectos para poder acceder a la información y registrarla en un log del sistema.

RESPUESTA

2 puntos por táctica

- a) Estableció un mecanismo para que los datos que se consultaban frecuentemente se adquirieran tempranamente y se almacenaran en un medio de rápido acceso. (EFICIENCIA - MAINTAIN MULTIPLE COPIES - CACHE)
- b) Utilizó protocolos de comunicación para proteger los datos confidenciales no se vean expuestos. (SEGURIDAD - MANTAIN DATA CONFIDENTIALITY - Encriptado)
- c) Utilizó como política general el principio de otorgar el menor número los privilegios a las cuentas de los usuarios (SEGURIDAD / AUTORIZACION DE USUARIOS)
- d) Se preocupó por definir interfaces y cuando fuera necesario agregar nuevas o utilizar adaptadores y stubs. (MANTENIBILIDAD - MANTAIN EXSITING INTERFACES)
- e) Utilizó el patrón MVC para que se pudieran ofrecer distintas vistas sobre la información contemplando las distintas necesidades y habilidades de los usuarios. (USABILIDAD / SEPARAR IU DE IMPLEMENTACION)

- f) Utilizo un componente para distribuir los accesos a los servidores y evitar que los componentes de su aplicación se vean sobrecargados. (EFICIENCIA/ Balanceador de carga)
- g) Se preocupó de que los componentes de interfaz de usuario tuvierán la capacidad de anticipar lo que el usuario fuera a realizar (USABILIDAD / Mantain a model of the task)
- h) Diseño para los distintos componentes una interfaz que permitieran acceder a información sobre el estado del componente, su carga, etc. Utilizó programación orientada a aspectos para poder acceder a la información y registrarla en un log del sistema (TESTEABILIDAD/ MONITOREO INTERNO)

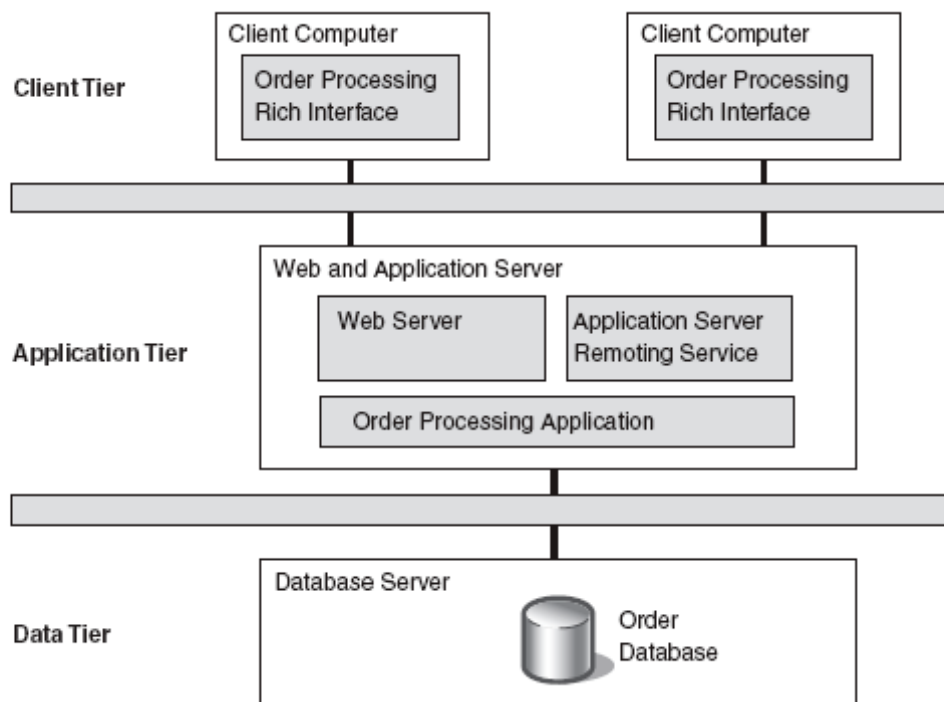
3 Estilos & Patrones (20 puntos)

Para los estilos estilo de arquitectura Capas físicas (Tiers) y Pipes and Filters

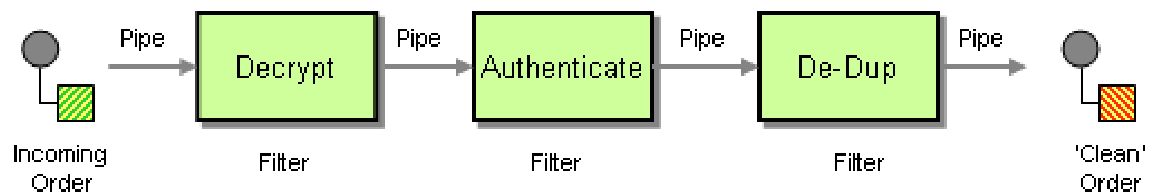
- Diagrame la topología del estilo y enumere las responsabilidades de los componentes.
- Discuta el estilo o patrón respecto a las cualidades que favorece.

RESPUESTA

10 puntos cada patron



Dentro de las principales cualidades se encuentran:
Mantenibilidad – los cambios a un componente no tienen que afectar a otros y además los cambios se focalizan.
Reuso – los componentes se pueden reusar en distintas aplicaciones
Eficiencia y Escalabilidad – los componentes se pueden desplegar en distintas capas maximizando la distribución del procesamiento.



Dentro de las principales cualidades se encuentran:
El reuso de filtros en distintas pipe lines
La mantenibilidad debido al bajo acoplamiento de los filtros
Poder paralelizar el procesamiento mejorando la eficiencia
Consultar en el material del curso por más información de este patrón

4. Documentación

4.1 (4 puntos) Defina el concepto de vista

4.2 (8 puntos) En la siguiente matriz se representan los paquetes empleados en la documentación de un módulo y sus relaciones de uso. A partir de la matriz, dibuje la vista de módulos (o vista lógica). Obs: Se puede leer la matriz de la siguiente manera, en la primera columna "client" usa "ejb", "god", "common" y "vo".

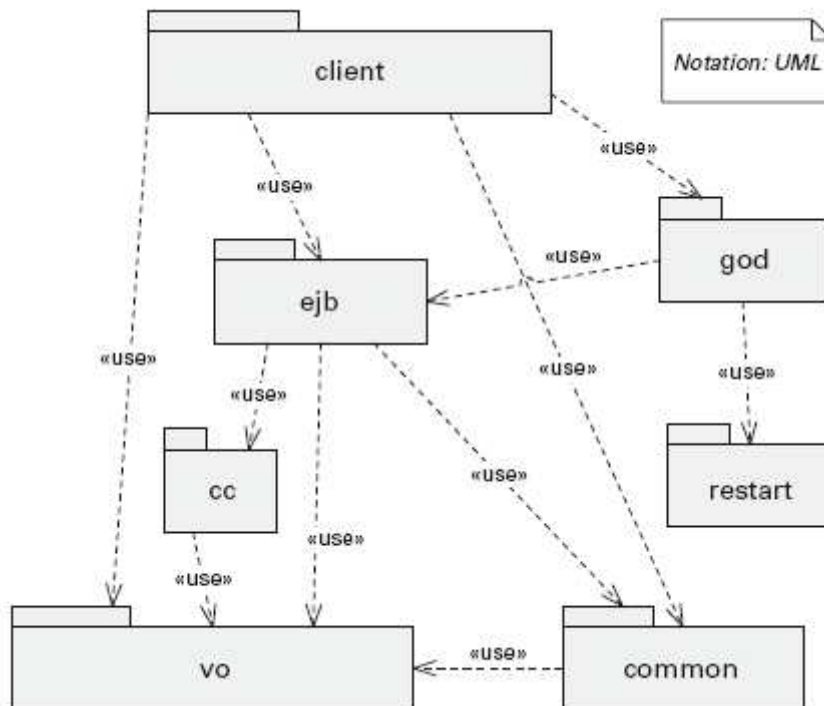
using module used module	client	ejb	cc	god	restart	common	vo
client	0	0	0	0	0	0	0
ejb	1	0	0	1	0	0	0
cc	0	1	0	0	0	0	0
god	1	0	0	0	0	0	0
restart	0	0	0	1	0	0	0
common	1	1	0	0	0	0	0
vo	1	1	1	0	0	1	0

RESPUESTA

a) Vista: Es el concepto más relevante en el contexto de la documentación de arquitecturas de software. Una vista se puede describir como la representación de un conjunto coherente de **elementos** arquitectónicos escrita para los interesados y la audiencia,. Consiste de un conjunto de elementos y sus relaciones. En gral existen 3 grupos de estructuras ...

(Ver def. de arquitectura para comprender el contexto de tales **elementos**).

b) A partir del diagrama se infiere lo siguiente:



Preguntas de Tecnología (30 puntos)
NOTA IMPORTANTE: Responde en hoja aparte

Pregunta 1 – Introducción a JEE (7 puntos)

1.1) ¿Qué es un contenedor?

Solución: Material de Curso

1.2) Describa brevemente cuatro servicios que brinda el contenedor

Solución:

JNDI, Administración de instancias, Acceso Remoto, Transacción, Seguridad, etc

Pregunta 2 - EJB (6 puntos)

2.1) Que diferencias existen entre los MDB y los Session Beans?

Solución:

Los clientes no acceden a MDB mediante interfaces.

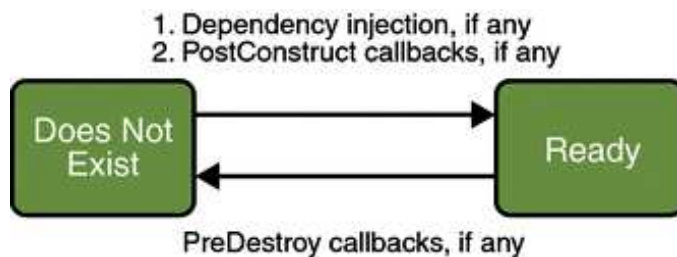
Los clientes acceden a los Session Bean mediante interfaces

La comunicación de los MDB es asincrónica

La comunicación de los Session Bean es sincrónica

2.2) Describa el ciclo de vida del Session Bean sin Estado.

Solución



Pregunta 3- JMS (8 puntos)

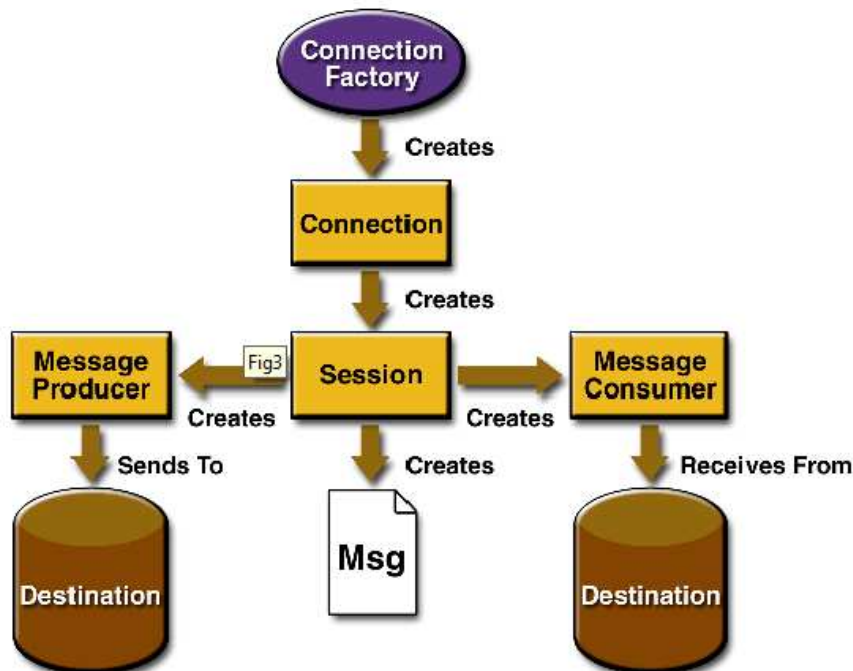
3.1) Explique los tipos de Destination que existen en JMS

Solución: Deben explicar Queue y Topic

3.2) Describa el Modelo de Programación de JMS incluyendo las responsabilidades de los principales elementos del modelo.

Solución

<http://java.sun.com/j2ee/1.4/docs/tutorial/doc/JMS4.html>

**Connection Factories**

A *connection factory* is the object a client uses to create a connection to a provider. A connection factory encapsulates a set of connection configuration parameters that has been defined by an administrator.

Destinations

A *destination* is the object a client uses to specify the target of messages it produces and the source of messages it consumes. In the PTP messaging domain, destinations are called queues. In the pub/sub messaging domain, destinations are called topics.

Connections

A *connection* encapsulates a virtual connection with a JMS provider. A connection could represent an open TCP/IP socket between a client and a provider service daemon. You use a connection to create one or more sessions.

Sessions

A *session* is a single-threaded context for producing and consuming messages. You use sessions to create message producers, message consumers, and messages. Sessions serialize the execution of message listeners; for details, see [Message Listeners](#).

A session provides a transactional context with which to group a set of sends and receives into an atomic unit of work. For details, see [Using JMS API Local Transactions](#).

Message Producers

A *message producer* is an object that is created by a session and used for sending messages to a destination. It implements the `MessageProducer` interface.

Message Consumers

A *message consumer* is an object that is created by a session and used for receiving messages sent to a destination. It implements the `MessageConsumer` interface.

A message consumer allows a JMS client to register interest in a destination with a JMS provider. The JMS provider manages the delivery of messages from a destination to the registered consumers of the destination.

Message Listeners

A *message listener* is an object that acts as an asynchronous event handler for messages. This object implements the `MessageListener` interface, which contains one method, `onMessage`. In the `onMessage` method, you define the actions to be taken when a message arrives.

Message Selectors

If your messaging application needs to filter the messages it receives, you can use a JMS API message selector, which allows a message consumer to specify the messages it is interested in. Message selectors assign the work of filtering messages to the JMS provider rather than to the application. For an example of an application that uses a message selector, see [A J2EE Application That Uses the JMS API with a Session Bean](#).

Messages

The ultimate purpose of a JMS application is to produce and to consume messages that can then be used by other software applications. JMS messages have a basic format that is simple but highly flexible, allowing you to create messages that match formats used by non-JMS applications on heterogeneous platforms.

A JMS message has three parts: a header, properties, and a body. Only the header is required. The following sections describe these parts:

- [Message Headers](#)
- [Message Properties](#) (optional)
- [Message Bodies](#) (optional)

For complete documentation of message headers, properties, and bodies, see the documentation of the `Message` interface in the API documentation.

Pregunta 4 – JSF (9 puntos)

Nota: Hay una sola opción correcta y las preguntas contestadas de forma incorrecta restan puntos

4.1) ¿Cuál de las siguientes declaraciones de reglas de navegación de JSF es correcta?

- A) <navigation-rule>
 <from-view-id>/login.jsp</from-view-id>
 <navigation-case>
 <from-outcome>login</from-outcome>
 <to-view-id>/menu.jsp<to-view-id>
 </navigation-case>
</navigation-rule>
- B) <navigation-rule>
 <to-view-id>/menu.jsp<to-view-id>
 <navigation-case>
 <from-outcome>login</from-outcome>
 <from-view-id>/login.jsp</from-view-id>
 </navigation-case>
</navigation-rule>
- C) <navigation-rule>
 <to-view-id>/menu.jsp<to-view-id>
 <navigation-case>
 <from-outcome>/login.jsp</from-outcome>
 <from-view-id>login</from-view-id>
 </navigation-case>
</navigation-rule>
- D) <navigation-case>
 <to-view-id>/menu.jsp<to-view-id>
 <navigation-rule>
 <from-outcome>/login.jsp </from-outcome>
 <from-view-id>login</from-view-id>
 </navigation-rule>
</navigation-case>

Respuestas

- 1-Opción B
- 2-Opción C
- 3-Opción A
- 4-Opción D
- 5-Ninguna de las anteriores

4.2) ¿Cuáles son los diferentes tipos de Scope en JSF?

- A) none,request, application
B) none,response,session,application

- C) request,application,session
- D) none,request,session,application

Respuestas

- 1-Opción B
- 2-Opción C
- 3-Opción A
- 4-Opción D
- 5-Ninguna de las anteriores

4.3) En qué archivo/s se declara los Managed Beans?

- A) En el web.xml
- B) En el jsf-config.xml
- C) En el config-faces.xml
- D) En el jboss-web.xml

Respuestas

- 1-Opción A y C
- 2-Opción C
- 3-Opción A
- 4-Opción D
- 5-Opción A y B
- 6-Opción B
- 5-Ninguna de las anteriores

Solución: 1,4,5

Duración:	3 horas
Con material:	No
Puntaje máximo:	100 puntos