

Preguntas de Diseño Arquitectónico (75 puntos)

1. (15 puntos) Conceptos sobre Arquitectura de software

En el capítulo 2 del libro Software Architecture in Practice los autores exponen tres aspectos para explicar por qué la Arquitectura de Software es importante como disciplina. Uno de esos aspectos se refiere a que la arquitectura de software es una *abstracción transferible de un sistema*.

Explique sintéticamente este punto, si es necesario recurra a ejemplos que permitan entender su comprensión del ítem.

Respuesta

Transcripción de Software Architecture in Practice, capítulo 2 sección 2.4:

Transferable abstraction of a system. Software architecture constitutes a relatively small, intellectually graspable model for how a system is structured and how its elements work together, and this model is transferable across systems. In particular, it can be applied to other systems exhibiting similar quality attribute and functional requirements and can promote large-scale re-use.

El libro profundiza este aspecto resaltando su uso en Software Product Lines, explicando los beneficios de la integración de componentes off-the-shelf, la intercambiabilidad.

2. Atributos de calidad: escenarios y tácticas

Las tácticas son decisiones de diseño que influyen la respuesta sobre determinados atributos de calidad. A una colección de tácticas se le denomina estrategia arquitectónica. Como parte de la estrategia arquitectónica se ha escogido el manejo de excepciones.

- a. **(10 puntos)** Identifique y defina a que atributo de calidad corresponde esta táctica y anote otra táctica que influya el mismo atributo.

Respuesta:

Se espera que identifique, de los seis atributos de calidad vistos en el curso, a cual corresponde el manejo de Excepciones. Que comprenda la definición operacional vista en clase y documentada en el capítulo 4 del libro del curso.

El atributo de calidad es **Disponibilidad**. La disponibilidad de un sistema se puede definir como la probabilidad de que el sistema este operativo cuando se le necesite. De modo que, la disponibilidad se puede expresar como $d = \text{Tiempo Medio falla} / \text{Tiempo Medio Falla} + \text{Tiempo de Reparación}$. De modo que, el 0. 1% de probabilidad de que no esté operacional el sistema se suele expresar como un

Ingeniería	
Examen de: Arquitectura de Software	Código de materia:
Fecha: 06/08/2012	Hoja 2 de 2

99.9% de disponibilidad. Un mecanismo para recuperarse de una “falta” es la apropiada gestión de excepciones. Las excepciones constituyen un refinamiento de las tácticas agrupadas bajo la Detección de Faltas.

Otras tácticas de detección de fallas son Ping y Hearbeat.

- b. (5 puntos)** La modificabilidad como atributo de calidad tiene que ver con el costo del cambio y no con falsas predicciones sobre lo que algún día puede cambiar. Bosqueje y explique las diferentes partes de un escenario de modificabilidad e indique la utilidad del mismo.

Respuesta:

Se puede comprender un escenario como un requerimiento sobre un atributo de calidad. Las partes de un escenario son fuente, estímulo, artefacto, entorno, respuesta y medida de la respuesta. En el caso de la modificabilidad la fuente es quien hace el cambio. El estímulo describe el cambio a realizar. El artefacto la porción del sistema a modificar. El entorno cuando debe o puede hacerse (tiempo de ejecución, compilación, diseño, etc.). La respuesta indica que quien hace el cambio debe comprender como hacerlo, testarlo, desplegarlo, etc. Medida de la respuesta: Todas las posibles respuestas del escenario tienen que ver con tiempo y costo del cambio. Los escenarios son útiles para especificar de forma operativa e independiente de taxonomías un requerimiento.

- c. (5 puntos)** Explique la diferencia entre “latencia” y “throughput” en términos de performance.

Respuesta:

Latencia: Refiere al tiempo que transcurre entre que el sistema recibe un estímulo, por ejemplo 1.000 transacciones por minuto y su respuesta.

Throughput: Refiere, por ejemplo, al número de transacciones que puede procesar en un segundo

- d. (5 puntos)** Distinga entre “autorización” y “autenticación” en términos de seguridad.

Respuesta:

Autenticación de usuarios refiere a asegurar que los usuarios o computadoras remotas sean quienes dicen ser; passwords y certificados digitales son formas comunes de proveer autenticación.

Autorización: Refiere a asegurar que un usuario autenticado tenga los privilegios para modificar datos o servicios, habitualmente se ofrecen soluciones de control de acceso para usuarios o clases de usuarios.

Ingeniería	
Examen de: Arquitectura de Software	Código de materia:
Fecha: 06/08/2012	Hoja 3 de 2

3 Estilos & Patrones

Teniendo en cuenta el estilo arquitectónico Publicador/Subscriber:

- a. **(10 puntos)** Presente el problema que resuelve, identifique los elementos del estilo y su topología, y describa las responsabilidades de los mismos.

Respuesta:

Se espera que identifiquen en la respuesta los siguientes contenidos:

Problema que resuelve: permite la colaboración entre elementos que no se conocen, no saben dónde se ubican y no pueden determinar si estarán disponibles al momento de requerir la colaboración (de hecho podrían colaborar sin estar disponibles en ese momento).

Los elementos del estilo son *publicadores* que son los elementos que exponen la información o servicios que ofrecen, *suscriptores* que son los elementos que consumen la información o servicios, *mensajes* que son los elementos que modelan la información que envían los publicadores a los suscriptores, y la *infraestructura de comunicación* que es el elemento que administra la colaboración a través de colas o tópicos recibiendo los mensajes de los publicadores y asegurando que le lleguen a los suscriptores.

- b. **(10 puntos)** Identifique los atributos de calidad que son favorecidos con el estilo. Justifique su respuesta.

Respuesta:

Los atributos que se benefician son:

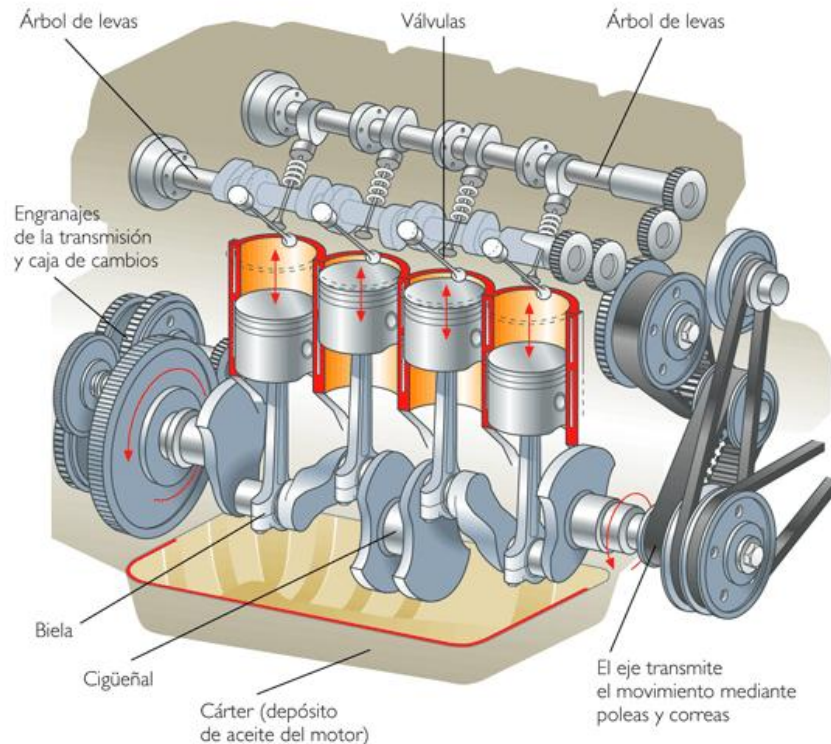
Modificabilidad: publicadores y suscriptores están totalmente desacoplados y el estilo permite agregar o quitar publicadores/suscriptores sin impactar a los demás. De la misma forma se pueden actualizar elementos sin impacto siempre que el cambio no afecte la estructura del mensaje que se comparte.

Seguridad: la gestión centralizada de los tópicos y mensajes permite que la infraestructura de comunicación implemente mecanismos de autorización, autenticación y confidencialidad

Disponibilidad: la infraestructura de comunicación implementa mecanismos de persistencia de los mensajes para asegurar el envío confiable y soporta transaccionabilidad.

4 Documentación de Arquitectura de Software

El motor de combustión es una máquina que transforma la energía generada de la quema de combustible en movimiento. Está compuesto por varios elementos que se muestran en la siguiente imagen, correspondiente a un motor de 4 cilindros:



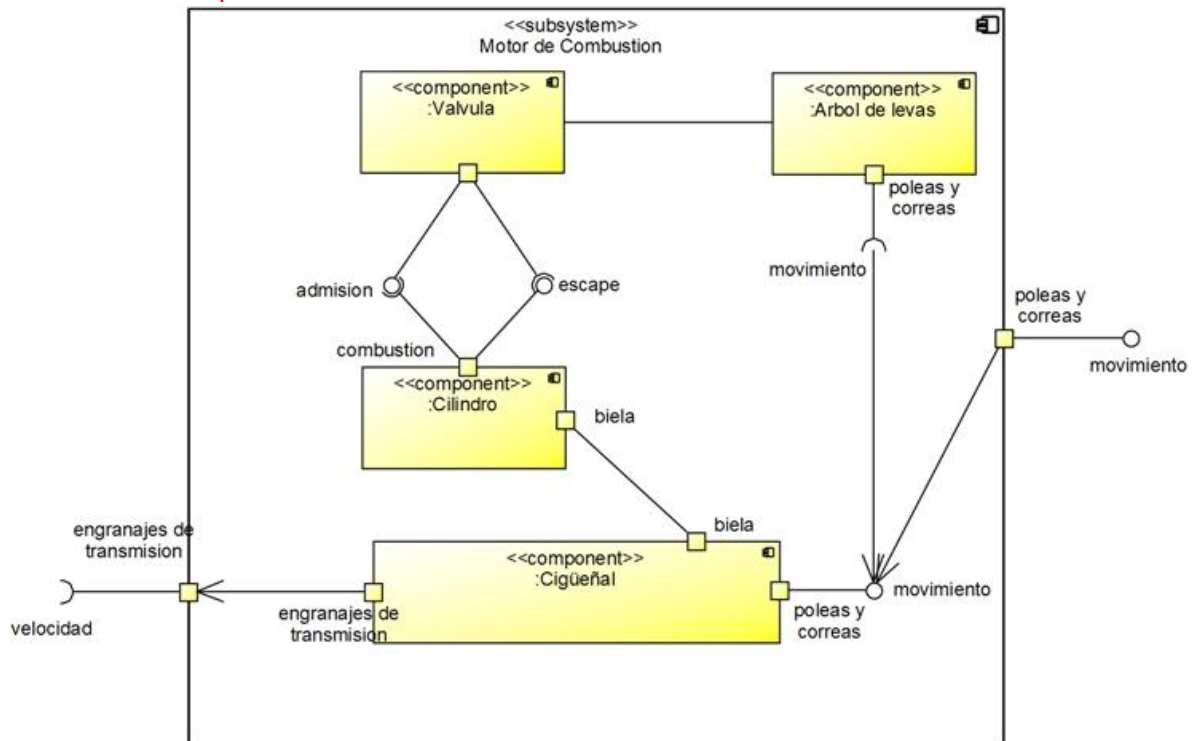
La combustión se produce en la cámara de cada **cilindro**. El cilindro comprime una mezcla de aire y combustible que entra a la cámara a través de las **válvulas**. La mezcla es luego encendida por una chispa produciendo una explosión que expande el cilindro permitiendo luego escapar los gases de la combustión. Cada cilindro está conectado a través de una **biela** a una pieza giratoria llamada **cigüeñal**. El movimiento repetitivo de compresión y expansión de los cilindros provoca que el cigüeñal gire. La velocidad con la que gira el cigüeñal está regulada por la caja de cambios que se conecta al cigüeñal a través de **engranajes de transmisión**. Mientras tanto, en la otra punta del cigüeñal se transmite el mismo movimiento a un sistema de **poleas y correas** que conecta el motor con el eje del vehículo para hacer girar las ruedas. Al mismo tiempo, también con poleas y correas se transmite el movimiento a los **árboles de levas** en la parte superior del motor. Cada árbol de levas permite con su giro abrir y cerrar las válvulas de forma sincronizada con el movimiento de los cilindros en las cámaras.

Se pide (15 puntos):

Realice un diagrama UML 2.x correspondiente a una vista de Componentes y Conectores que represente el funcionamiento de un motor de combustión de acuerdo a la descripción anterior.

Respuesta:

Una alternativa podría ser:



Ingeniería	
Examen de: Arquitectura de Software	Código de materia:
Fecha: 06/08/2012	Hoja 6 de 2

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

La biblioteca de la universidad lo contrata como Arquitecto para construir su sistema de gestión en la web. Los requerimientos que debe implementar el sistema son:

- **R1- Gestión de Socios**
El sistema debe permitir realizar la administración (ABM) de los socios de la biblioteca.
Socio: Número de socio, Teléfono, Dirección, E-Mail, Nombre y Apellido
- **R2- Gestión de Libros**
El sistema debe permitir realizar la administración (ABM) de los libros.
Libro: Número de Libro, Autor, Tema y Título
- **R3- Consulta de Libros**
El sistema debe permitir realizar consultas de los libros de la biblioteca.
- **R4-Consulta de Socios**
El sistema debe permitir realizar consultas de los socios de la biblioteca.
- **R5-Autorización de Reserva**
El sistema debe poder realizar consultas al sistema central de la universidad que está desarrollado en .NET para obtener información del estudiante y si está habilitado (sin haberes impagos) para autorizar la reserva de un libro.
- **R6-Consulta de Reservas**
El sistema debe permitir realizar consultas de las reservas de un libro mediante una página web.
- **R7-Reservas**
El sistema debe permitir realizar reservas mediante una página web en la cual se ingresa el número de libro, el número de socio y la fecha para realizar la reserva, confirmando la reserva mediante un envío de mail. La reserva se debe realizar en forma asincrónica.

Para el almacenamiento de la información se utiliza Bases de Datos

Ingeniería	
Examen de: Arquitectura de Software	Código de materia:
Fecha: 06/08/2012	Hoja 7 de 2

Se pide:

1. **(10 puntos)**- Identifique el estilo arquitectónico en el que se basa la especificación JEE y describa los elementos del estilo según la misma.

Respuesta:

JEE se basa en el estilo de capas físicas. La especificación define una arquitectura de 4 capas: cliente, web, negocio y EIS (Enterprise Information System). La capa cliente representa tanto navegadores web, applets o aplicaciones de escritorio que se comunican remotamente con el servidor de aplicaciones. Las capas web y de negocio se alojan en el servidor de aplicaciones. La capa EIS representa la fuente de datos que puede eventualmente corresponder a otro sistema

2. **(5 puntos)** – Nombre las tecnologías JEE que utilizaría para construir esta aplicación y describa para qué propósito o requerimiento la aplicaría.

Respuesta:

En este caso la aplicación es web por lo tanto la capa de cliente solo tendrá clientes web que acceden a través de navegadores de Internet. En la capa web usaría JSF para desarrollar la presentación al usuario (pantallas y navegación de la aplicación). En la capa de negocio utilizaría EJB Session Bean para encapsular la lógica de negocio y exponerla a la presentación de la aplicación. Como la persistencia es en base de datos, utilizaría clases Java anotadas con JPA para modelar las entidades del negocio y sus relaciones y mapearlas con las tablas de la base. Para el envío de mails utilizaría JavaMail. Para consultar el estado de los estudiantes quizás utilice web services, dependiendo del mecanismo de interoperabilidad que exponga la aplicación central de la Universidad

3. **(10 puntos)**-Identifique los contenedores definidos en JEE y describa los servicios que brindan a los componentes a utilizar en este sistema.

Respuesta:

1. Transcribo desde JEE 6 Tutorial (<http://docs.oracle.com/javaee/6/tutorial/doc/bnabo.html#bnabp>)

Java EE Containers

Normally, thin-client multitiered applications are hard to write because they involve many lines of intricate code to handle transaction and state management, multithreading, resource pooling, and other complex low-level details. The component-based and platform-independent Java EE architecture makes Java EE applications easy to write because business logic is organized into reusable components. In addition, the Java EE server provides underlying services in the form of a container for every component type. Because you do not have to develop these services yourself, you are free to concentrate on solving the business problem at hand.

Container Services

Containers are the interface between a component and the low-level platform-specific functionality that supports the component. Before it can be executed, a web, enterprise bean, or application client component must be assembled into a Java EE module and deployed into its container.

The assembly process involves specifying container settings for each component in the Java EE application and for the Java EE application itself. Container settings customize the underlying support provided by the Java EE server, including such services as security, transaction management, Java Naming and Directory Interface (JNDI) API lookups, and remote connectivity. Here are some of the highlights.

- The Java EE security model lets you configure a web component or enterprise bean so that system resources are accessed only by authorized users.
- The Java EE transaction model lets you specify relationships among methods that make up a single transaction so that all methods in one transaction are treated as a single unit.
- JNDI lookup services provide a unified interface to multiple naming and directory services in the enterprise so that application components can access these services.
- The Java EE remote connectivity model manages low-level communications between clients and enterprise beans. After an enterprise bean is created, a client invokes methods on it as if it were in the same virtual machine.

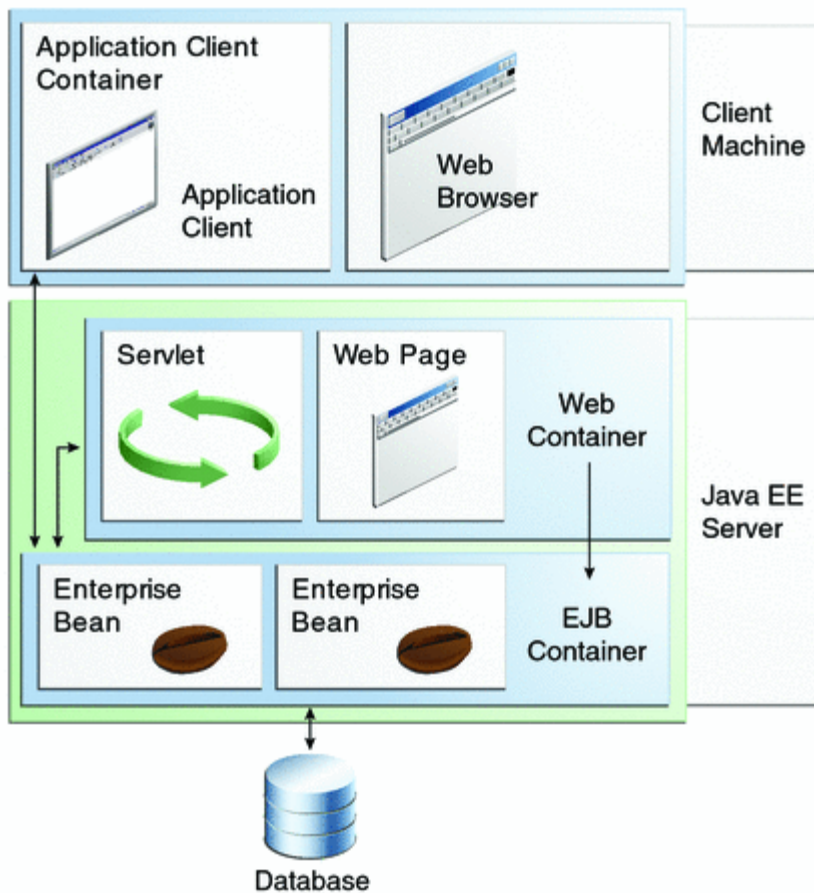
Because the Java EE architecture provides configurable services, application components within the same Java EE application can behave differently based on where they are deployed. For example, an enterprise bean can have security settings that allow it a certain level of access to database data in one production environment and another level of database access in another production environment.

The container also manages nonconfigurable services, such as enterprise bean and servlet lifecycles, database connection resource pooling, data persistence, and access to the Java EE platform APIs (see [Java EE 6 APIs](#)).

Container Types

The **deployment** process installs Java EE application components in the Java EE containers as illustrated in [Figure 1-5](#).

Figure 1-5 Java EE Server and Containers



- **Java EE server:** The runtime portion of a Java EE product. A Java EE server provides EJB and web containers.
- **Enterprise JavaBeans (EJB) container:** Manages the execution of enterprise beans for Java EE applications. Enterprise beans and their container run on the Java EE server.
- **Web container:** Manages the execution of web pages, servlets, and some EJB components for Java EE applications. Web components and their container run on the Java EE server.
- **Application client container:** Manages the execution of application client components. Application clients and their container run on the client.
- **Applet container:** Manages the execution of applets. Consists of a web browser and Java Plug-in running on the client together.

Ingeniería

Examen de: Arquitectura de Software

Código de materia:

Fecha: 06/08/2012

Hoja 10 de 2

Duración: 3 horas

Material de apoyo: No

Puntaje máximo: 100 puntos