

Escuela de Ingeniería

Examen de: Arquitectura de Software

Código de materia: 3851

Fecha: 24-2-10

Id Examen: 23300

Hoja 1 de 7

Preguntas de Diseño Arquitectónico (70 puntos)

1. Atributos de calidad, escenarios y tácticas

1.1. Escenarios

(8 puntos) Dados los siguientes ejemplos de escenarios *generales*, escritos en forma textual, correspondientes a atributos de calidad

¿Identifique qué atributos de calidad especifican estos escenarios?

Para los escenarios A y B descompóngalos en el formato de escenario general visto en el curso y en el libro Software Architecture in Practice.

- A. La plataforma sobre la cual ejecuta el sistema será cambiada a solicitud de un cliente. El cambio puede abarcar dispositivos de hardware, plataforma operativa y del middleware utilizado en la solución. El sistema debe poder modificarse en tiempo de diseño y desarrollo manteniendo la funcionalidad actual en un tiempo menor a dos semanas.
- B. Un importante conjunto de solicitudes son ingresadas desde la interfaz de usuario y afectan directamente a una entidad de datos del sistema. Estando el sistema ejecutando en condiciones normales. El sistema debe transferir los datos al componente minimizando el número de mensajes que se transmiten por la red
- C. Un interesado del sistema solicita un cambio a una estructura de datos sobre un sistema en operación. El cambio a la estructura de datos debe poder realizarse en diseño con un costo menor a 3000 dólares y en menos de 48 horas.
- D. Cuando falla un componente interno al sistema el mismo debe ser capaz de reconocer la falla y compensarla manteniendo el nivel de servicio.

1.2. Tácticas

(12 puntos) Dada la siguiente lista de tácticas de arquitectura, indique a qué atributo de calidad corresponden

Táctica	Atributo de calidad
Reiniciación	
Cache	
Ocultar información	
Limitar accesos	
Balanceo de carga	
Utilizar interpretes	
Utilización de multihilos	

Escuela de Ingeniería

Examen de: Arquitectura de Software

Código de materia: 3851

Fecha: 24-2-10

Id Examen: 23300

Hoja 2 de 7

Pooling	
Utilizar intermediarios	
Limitar exposición	
Utilizar múltiples vistas para presentar información	
Utilizar una plataforma alternativa	

1.3. (6 puntos) Explique brevemente en qué consisten las tácticas adquisición temprana de recursos y registro (o enlace) en tiempo de ejecución.

¿Qué atributo de calidad mitigan?

¿Cómo lo mitigan?

2. Estilos y patrones de arquitectura

2.1. (20 puntos) Un arquitecto realizó el diseño que se describe a continuación para un sistema de edición de información de documentos de una biblioteca digital.

El sistema consiste de una aplicación de escritorio orientada a editores de publicaciones que permite:

- Ingresar al sistema mediante usuario y contraseña.
- Abrir un documento ubicado en el sistema de archivos del PC.
- Ingresar atributos de la publicación (por ejemplo título, autor, ISBN, tipo publicación, etc) y palabras claves asociadas al documento.
- Indica el conjunto de operaciones de indexación que es necesario realizar sobre el documento (por ejemplo indexas por atributos, por contenido, etc.)

Cuando el usuario finaliza el ingreso de la información sobre la publicación la aplicación genera un mensaje con los datos ingresados y el documento asociado. Este mensaje se envía a un componente de distribución ubicado en un servidor (servidorDeDistribucion).

El componente de distribución lee los mensajes enviados por las aplicaciones de los editores y tiene la función de distribuir los mensajes a otros componentes especializados en indexación de datos (por ejemplo un componente indexa por atributos y otro indexa por contenido), los cuales residen en otros servidores. Pueden existir varias instancias de cada tipo de componentes indexador. A su vez los componentes de indexación tienen la capacidad de informarle al componente de distribución sobre qué tipo de trabajo de indexación realizan. Este mecanismo permite que en el futuro se desarrollen otras formas de indexación.

Los componentes especializados en indexación leen los mensajes que le fueron enviados y los analizan y almacenan la información del índice y el documento en una base de datos que reside en otro servidor. La base de datos se replica cada determinado tiempo en otro servidor.

Se pide:

- Diagrama la vista de despliegue del sistema mostrando los nodos, los componentes y la forma de conexión entre los componentes.

Escuela de Ingeniería

Examen de: Arquitectura de Software

Código de materia: 3851

Fecha: 24-2-10

Id Examen: 23300

Hoja 3 de 7

- b) Identifique y describa brevemente los estilos o patrones de arquitectura utilizados.
- c) Identifique las tácticas de arquitectura utilizadas en este diseño.

2.2. (14 puntos) Explique el estilo de arquitectura **Publicador Suscriptor**.

- a) Describa los elementos del estilo (topología), las responsabilidades de sus componentes y las distintas variantes del mismo en cuanto a su funcionamiento
- b) Conoce alguna tecnología que implemente este patrón.

3. Descripción de arquitecturas

3.1. (10 puntos) Algunos de los objetivos del documento de Descripción de la arquitectura de un software son:

- a) Mostrar la forma en que se despliegan los componentes de software en la infraestructura.
- b) Mostrar qué componentes se van a desarrollar y las formas de conexión de los mismos.
- c) Mostrar a partir de qué módulos (lógicos) se construyen los componentes.

Basándose en el modelo 4+1 y en las secciones y elementos de las plantillas para Descripción de arquitectura entregadas en el curso explique qué vistas o información utilizaría para expresar cada uno de los puntos anteriores.

Escuela de Ingeniería

Examen de: Arquitectura de Software

Código de materia: 3851

Fecha: 24-2-10

Id Examen: 23300

Hoja 4 de 7

Preguntas de Tecnología (30 puntos)

NOTA: Responder en hoja aparte


1. EJB

El Comité Olímpico organizador de las Olimpíadas de Invierno actualmente en curso en Vancouver necesita un sistema de información para administrar la información de los competidores, sus equipos, los países, las disciplinas y por supuesto las competiciones y sus resultados.

Varios equipos de desarrollo trabajan en la construcción de la aplicación. Un equipo E1 desarrolla la interfaz de usuario, otro equipo E2 desarrolla la lógica de negocio y otro equipo E3 es responsable de la persistencia del modelo de dominio.

El arquitecto de la solución dispuso que la aplicación se divida en 3 módulos que se instalarán y ejecutarán en servidores independientes. El servidor SRV_1 servirá las funcionalidades de administración (crud de competidores, equipos, países, disciplinas). El servidor SRV_2 servirá las funcionalidades para las competencias (calificaciones de los jueces, resultados de cada competición). El servidor SRV_3 servirá la consulta del “medallero olímpico”.

Asimismo dispuso que las operaciones de negocio implementadas en cada módulo sean expuestas a la interfaz de usuario a través de EJBs. El equipo E2 encargado de esto construyó los siguientes beans:

Módulo	ejb-admin
Paquetización	com.vancouver2010.ejb.bean
Estructura	

Escuela de Ingeniería

Examen de: Arquitectura de Software

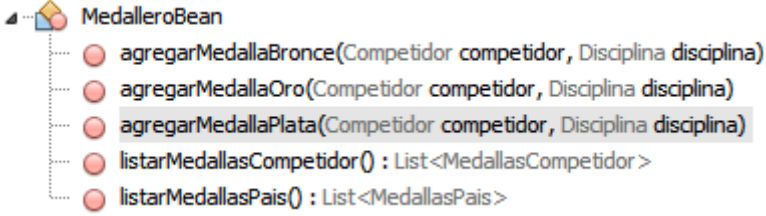
Código de materia: 3851

Fecha: 24-2-10

Id Examen: 23300

Hoja 5 de 7

Módulo	ejb-competencias
Paquetización	com.vancouver2010.ejb.bean
Estructura	 <pre> classDiagram class CompetenciasBean { +asignarPuntaje(Competidor competidor, Double nota) +iniciarCompetencia(Competencia competencia) +procesarPuntajes() +calificaciones : Map<Long, Double> +competenciaEnCurso : Competencia } </pre>

Módulo	ejb-medallero
Paquetización	com.vancouver2010.ejb.bean
Estructura	 <pre> classDiagram class MedalleroBean { +agregarMedallaBronce(Competidor competidor, Disciplina disciplina) +agregarMedallaOro(Competidor competidor, Disciplina disciplina) +agregarMedallaPlata(Competidor competidor, Disciplina disciplina) +listarMedallasCompetidor() : List<MedallasCompetidor> +listarMedallasPais() : List<MedallasPais> } </pre>

Consideraciones:

- Para minimizar los accesos a la base datos y favorecer un mejor tiempo de respuesta a los jueces, se dispuso que las notas que carga cada juez se procesen y almacenen juntas de una sola vez al finalizar cada competencia. Al cargar cada nueva nota el juez podrá ver las notas cargadas anteriormente para la competencia. No se permite utilizar almacenamientos temporales de ningún tipo durante la carga.
- El módulo *competencias* solamente requiere consumir las funcionalidades de consulta del módulo *administracion*.
- Cada módulo debe exponer a los demás únicamente las operaciones requeridas.
- Luego de realizar un testeo de estrés, el arquitecto nota que el módulo *medallero* tiene serios problemas de disponibilidad (se cae aleatoriamente). Como medida de “prevención” dispone que las operaciones para agregar medallas se invoquen de forma asincrónica desde el módulo *competencias*.

Se pide:

1. Determine las interfaces que usará cada bean para exponer sus funcionalidades y especifique las anotaciones requeridas para convertir las clases bean en los EJB del tipo que considere necesario en cada caso. Indique las consideraciones de Diseño que considere importantes para la implementación de las interfaces. **(10 puntos)**
2. Proponga cómo implementaría la comunicación asincrónica entre los módulos *competencias* y *medallero*. **(10 puntos)**

Escuela de Ingeniería

Examen de: Arquitectura de Software

Código de materia: 3851

Fecha: 24-2-10

Id Examen: 23300

Hoja 6 de 7

2. JPA

El equipo de desarrollo E3 ha tenido serios problemas para lograr un modelo físico del modelo de dominio. La estructura de la base de datos la obtienen a partir de los mapeos realizados con JPA sobre las clases Java del dominio. A continuación le presentan a ud. las clases con sus mapeos:

```
@Entity
@Table(name="COMPETENCIAS")
public class Competencia {

    @PrimaryKey(name="CLAVE")
    private Long id;

    @Column(name="NOMBRE")
    private String nombre;

    @Column(name="DISCIPLINA")
    private Disciplina disciplina;

    @Relation
    private List<Competidor> competidores;

}
```

```
@JPA
@Table(name="COMPETIDORES")
public class Competidor {

    @Column(name="CLAVE")
    private Long id;

    @Column(name="NOMBRE")
    private String nombre;

    @Column(name="FECHA_NACIMIENTO")
    private Date fechaNacimiento;

    @ManyToOne
    private Pais paisNacimiento;

}
```

```
@Class
@Table(name="DISCIPLINAS", primaryKey="id")
public class Disciplina {

    @Column(name="CLAVE")
```

Escuela de Ingeniería

Examen de: Arquitectura de Software

Código de materia: 3851

Fecha: 24-2-10

Id Examen: 23300

Hoja 7 de 7

```
private Long id;

@Column(name="NOMBRE")
private String nombre;

}
```

```
@Entity
@Table(name="PAISES")
public class Pais {

    @Id(name="CLAVE")
    private Long id;

    @Column(name="NOMBRE")
    private String nombre;

}
```

Se pide:

1. Identifique y corrija los errores que encuentre en los mapeos de estas clases, de forma de asegurar que se pueda obtener un modelo físico de la base de datos a partir de los mapeos corregidos. **(10 puntos)**

Duración: 3 horas
Sin Material