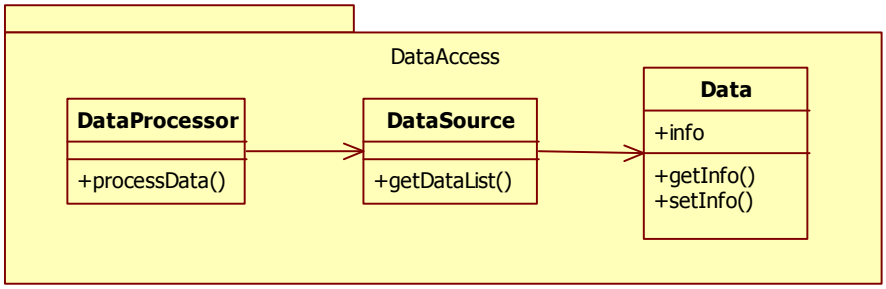
Escuela de Ingeniería		
Examen de: Arquitectura de Software		**Código de materia: .3851**
Fecha: 17-08-2009	**Id Examen: 22352**	**Hoja 1 de 6**

Preguntas de Diseño Arquitectónico (60 puntos)

1. Principios de diseño y UML (15 puntos)

El diagrama UML de la figura representa un paquete *(DataAccess)* con un conjunto de entidades (*DataProcessor, DataSource y Data*) que tienen la responsabilidad de resolver cierto acceso a datos. Adicionalmente, se adjunta una implementación del método processData().




```
public class DataProcessor {
        public void processData() {
                DataSource source = new DataSource();
                List<Data> dataList = source.getDataList();
                for(Data data : dataList) {
                        System.out.println(data);
                }
        }
}
```

a. Dibuje un nuevo modelo *(notación estricta UML)* efectuando un refactoreo sobre la entidad **DataSource** de modo tal que, **DataProcessor** pueda emplear de forma intercambiable implementaciones del método **getDataList()** para tres nuevos tipos de DataSource: **XMLDataSource**, **WebServiceDataSource** y **DBDataSource**.

b. Anote los cambios sobre el método processData() para poder emplear los métodos getDataList(); por defecto, la invocación al método processData() debe llamar al getDataList() de la clase **DBDataSource**.

c. Las entidades XMLDataSource, WebServiceDataSource y DBDataSource son reacomodadas *(movidas)* en un nuevo paquete denominada **DataAccessImpl**.

Observerse que, si el refactoreo de la parte anterior es el correcto, DataProcessor aún dependerá de la entidad DBDataSource que constituye un *detalle de implementación* del acceso a datos.
Resuelva el problema de creación de objetos mediante alguna implementación de los patrones Factory. Dibuje un nuevo modelo UML que permita mostrar que es posible aplicar el Principio de Inversión de Dependencias en el cual, las abstracciones (representadas por las entidades del paquete DataAccess) no dependen de los detalles de implementación (representados por el paquete DataAcessImpl).

2. Tácticas de atributos de calidad (15 puntos)

2.1 - Dados los siguientes escenarios:
a) Identifique a qué atributo de calidad corresponden y porqué.
b) Explique cada atributo identificado.

1. Durante la operación normal del sistema si se produce una falla en el enlace de red la Terminal de venta debe notificar al operador y continuar registrando las ventas sin tiempo de detención.

2. Un desarrollador desea incorporar un nuevo tipo de mensaje de intercambio de información en el middleware del sistema la modificación debe poder desplegarse sin necesidad de detener la operación del sistema.

3. Cuando el usuario acepta la transacción de transferencia el sistema debe responder la confirmación de la operación en un tiempo menor a un segundo con el sistema operando con una carga de 3000 usuarios concurrentes.

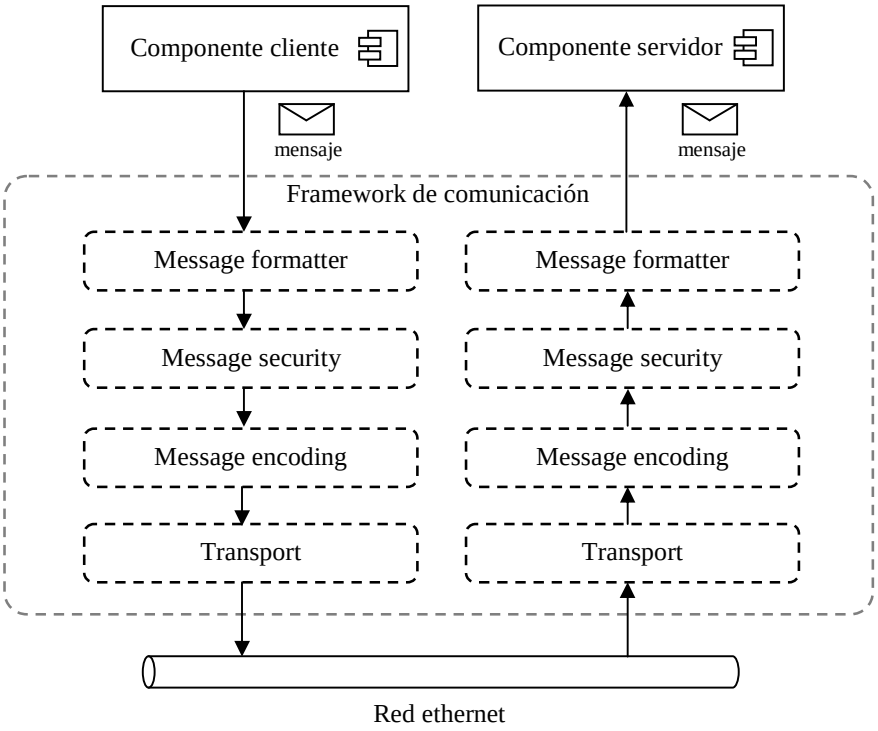
4. Si una fuente no identificada y externa a la empresa intenta acceder a la base de datos de ventas el sistema debe bloquear el acceso a la base de datos, notificar al operador de seguridad y restringir el servicio durante 15 minutos.

2.2 Un Arquitecto de software durante el diseño de un sistema tomó las siguientes decisiones.
Identifique el nombre de cada una de las tácticas utilizadas y explique a qué atributo de calidad pertenecen.

a) Se preocupó porque todos los diseñadores técnicos conocieran el principio de diseño abierto cerrado.
b) Se preocupó porque todos los diseñadores técnicos conocieran el principio de diseño libración y reuso.
c) Adquirió un software basado en redes neuronales que permite reconocer el patrón de acceso a la base de datos de tarjetas de crédito.
d) Definió un servicio transversal a todas las capas del sistema que registra todas las operaciones realizadas en las distintas capas identificando la fecha y hora, la identificación del usuario y la IP desde donde proviene el acceso.
e) Diseño un algoritmo que permite cada vez que permite calcular la suma de todos los bytes de los archivos transferidos desde los clientes para luego poder verificar que el archivo recibido no esté corrupto.
f) Utilizó un mecanismos que permite localizar recursos que se encuentran distribuidos en forma rápida.
g) en las implementaciones de los servicios web exigió que los mismos se desconecten de sus clientes luego de 3 segundos sin acceso al web servicio.
h) Definió que determinados datos deberían estar en memoria para poder accederlos rápidamente.

3. Estilos y patrones de diseño (15 puntos)

El siguiente diagrama representa brevemente la arquitectura de un reconocido framework de comunicaciones y mensajería remota entre componentes de software:



Los elementos aquí presentados son los siguientes:

- Message formatter: Transforma el mensaje de entrada representando los datos enviados o recibidos por los componentes correspondientes utilizando un formato XML estándar.

- Message security: Del lado del componente cliente, agrega información de seguridad como ser el nombre de usuario y contraseña al mensaje de entrada. Del lado del servidor, extrae dicha información adicional de seguridad para posteriormente validar las credenciales del cliente.

- Message encoding: Transforma los mensajes de entrada para que éstos puedan ser transportados por una red de comunicaciones y a su vez, transforma los datos recibidos por la red a un formato estándar. Por ejemplo, transforma mensajes XML a un formato binario y viceversa.

- Transport: Gestiona el transporte de los datos, adaptándolos a un determinado protocolo de comunicación.

a. Identifique el patrón que utiliza dicho framework y describa el objetivo del mismo.

b. Explique ventajas y desventajas de dicho patrón o estilo.

4. Descripción arquitectónica (15 puntos)

4.1 Explique y justifique para cada uno de los tipos de vistas estudiados (ver lista que sigue) cómo los accionistas de la arquitectura pueden satisfacer sus necesidades para entender la arquitectura de un determinado sistema de software.

- Módulos en Views &Beyond o vista de diseño modelo 4+1
- Componentes y Conectores en Views &Beyond o vista de Implementación en el modelo 4+1
- Asignación en Views &Beyond o vista de Despliegue en el modelo 4+1

4.2 Identifique qué vistas representan más adecuadamente los intereses de cada accionista.

Preguntas de Tecnología (40 puntos)

NOTA: Responder en hoja aparte

1. Plataforma JEE (10 puntos)

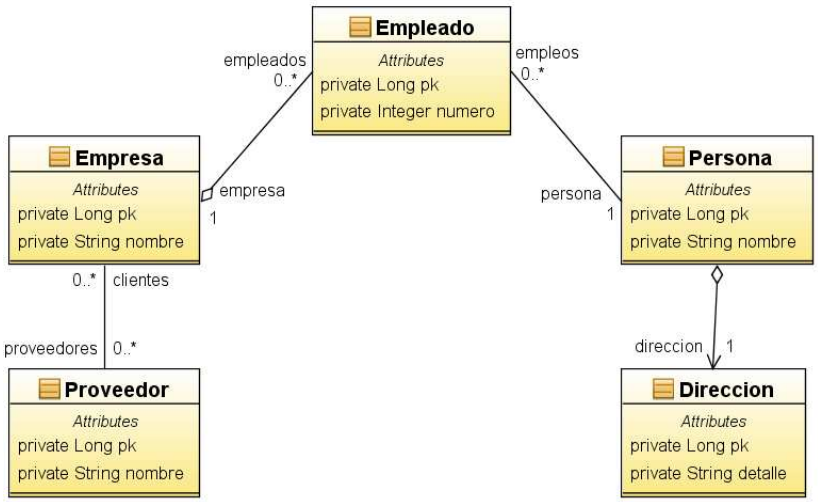
Indique el estilo de arquitectura que propone la especificación JEE para la construcción de aplicaciones empresariales y describa sus partes (No detallar tecnologías).

2. Componentes EJB (10 puntos)

Un compañero de trabajo le pide que le describa los pasos necesarios para instanciar un EJB Message Driven Bean desde una aplicación cliente Java haciendo un lookup remoto al JNDI de una aplicación JEE.
Explique la respuesta que daría a su compañero.

3. Persistencia en JEE (20 puntos)

A continuación se presenta un diagrama de clases con las entidades persistentes del dominio de una aplicación construida con tecnologías JEE.



Para obtener el modelo de tablas correspondiente a este diseño es necesario satisfacer los siguientes lineamientos definidos por el DBA responsable de la empresa:

- Todos los objetos de base de datos y sus atributos deben expresarse en mayúsculas, utilizando el carácter "_" como separador de las palabras significativas.
- Los nombres de las tablas deben comenzar con el prefijo "TBL_"
- Los nombres de los campos en cada tabla deben comenzar con un prefijo único de 3 letras que identifican la tabla a la que pertenecen seguido del carácter "_" (por ejemplo, para la tabla Personas el prefijo de los campos puede ser "PER_")
- En particular, los campos que almacenan claves foráneas deben agregar al prefijo los caracteres "FK_" (Por ejemplo, el campo que guarda la dirección en la Persona puede ser "PER_FK_DIRECCION")
- Las restricciones que se definan para las claves foráneas (*foreign key constraints*) deben nombrarse concatenando los nombres de las tablas de la relación más el sufijo "FK", separados por el carácter "_". (Por ejemplo, la restricción para la clave foránea dirección en la tabla de Persona podría ser "PERSONAS_DIRECCIONES_FK")

Nota:

El atributo definido para identificar unívocamente cada objeto para cada clase es "pk", que es utilizado como identificador JPA. No se utiliza ningún generador para el identificador, es asignado por el usuario.

Se pide:

3.1 Escriba **en lenguaje JAVA** las clases que implementan el diseño del diagrama incluyendo las anotaciones JPA necesarias para persistir los objetos de acuerdo a los lineamientos establecidos (omitir getters y setters, solo describir atributos).

3.2 Proponga un diagrama que muestre las tablas que se crearán en la base de datos de acuerdo a los mapeos realizados y que sea coherente con los requerimientos del diagrama de clases inicial.

Duración: 3 horas
Sin Material