

1 Relación de ejercicios. *Tema 2*

19. Se pide analizar los requisitos de los cuatro sistemas que se listan más abajo. Primero es necesario descomponer cada uno de ellos en subsistemas.
 - (a) Un sistema de información de la policía.
 - (b) Un sistema para una inmobiliaria.
 - (c) Un sistema para una biblioteca pública.
 - (d) Un sistema para una empresa de alquiler de vehículos.
20. Clasificar los siguientes aspectos de un diseño de sistema respecto de los tipos de cohesión que se podrían identificar si hubiera sido diseñado adecuadamente:
 - (a) Toda la información relativa a las órdenes de reserva se mantendrá dentro de una determinada clase y todo lo demás fuera de ella.
 - (b) Se creará un módulo para convertir una imagen *bitmap* al formato JPEG.
 - (c) Se creará un subsistema diferente que se ejecute cada noche para generar las estadísticas de ventas del día anterior.
 - (d) Una operación de procesamiento de datos implicará recibir entradas de varias fuentes, ordenarlas, resumir la información de cada fuente de entrada, ordenar de acuerdo con la fuente de entrada que generó la mayoría de los datos y después devolver los resultados para que sean utilizados por otros subsistemas. El código para llevar a cabo los pasos anteriores se mantendrá junto aunque se pueda llamar a diferentes programas de utilidad para realizar algunas operaciones tales como ordenar.
21. Determinar los fallos en los siguientes diseños de sistemas desde el punto de vista de la *cohesión* y qué tendría que hacerse para arreglarlos.
 - (a) Hay dos subsistemas en un sistema de matrículas de la Universidad para realizar lo siguiente, *SubsistemaA*: muestra listados con las asignaturas a un estudiante, acepta peticiones de matriculación del estudiante, verifica que el estudiante no elija asignaturas con horarios incompatibles y que éste se pueda matricular en las asignaturas elegidas, almacena todos datos y periódicamente hace una copia de seguridad de la base de datos.
Subsistema B: permite a los profesores de la Facultad calificar a los estudiantes, a los administradores encargar asignaturas a los profesores, añadir nuevas asignaturas, alterar las matrículas de los estudiantes.
Por último, también imprimirá las facturas que se envían a los estudiantes.
 - (b) En una aplicación de comercio electrónico, se creará un módulo para añadir libros al “carrito de la compra” y realizar algunas operaciones tales como calcular todo lo que deba un cliente. Un segundo módulo añadirá artículos de regalo al carro de la compra de un cliente, además de mostrar los contenidos del carro en la pantalla y enviar un correo al cliente recordándole lo que compró.

22. Una forma de conseguir *acoplamiento de control* en Java sería mediante la utilización de su mecanismo denominado *reflexión*. Dicho mecanismo permite en Java tratar directamente a una cadena (*String*) como un nombre de método, para llamar a dicho método posteriormente. Investigar el mecanismo de *reflexión* en Java y determinar las diferencias de implementación para un método que manejase cadenas de clientes utilizando: (a) *reflexión* o (b) *manteniendo el acoplamiento de control*.
23. Clasificar los siguientes aspectos de un diseño respecto de los distintos tipos de *acoplamiento* que presentan:
- (a) La clase `GrupoAsignatura` posee las variables de clase públicas: `tamanoMinimoClase` y `tamanoMaximoClase`. Los valores de estas variables son cambiados de vez en cuando por la administración de la Universidad. Existen varios métodos de las clases `Estudiante` y `Matriculacion` que acceden a dichas variables.
 - (b) Una clase interfaz de usuario importa un gran número de otras clases Java, incluyendo aquellas que dibujan gráficos, las que crean controles UI y varias otras clases de utilidad.
 - (c) Un sistema posee una clase denominada `Direccion`. Esta clase posee cuatro variables públicas que conforman diferentes partes de una dirección. Varias clases diferentes, tales como `Persona` y `Aeropuerto` manipulan instancias de la clase anterior, modificando directamente los campos de las direcciones. También, muchos métodos declaran uno de sus argumentos como del tipo `Direccion`.
Las variables de clase: `tamanoMinimoClase` y `tamanoMaximoClase` no deberían tener visibilidad pública.
24. Describir diferentes formas de reducir los casos de acoplamiento que se dan en el ejercicio anterior.
25. Determinar las clases de acoplamiento que se derivan del diseño de una aplicación *Chat* simple en Java.
26. Se nos proporciona la siguiente tabla que muestra los niveles de calidad que se alcanzan con la utilización de varias arquitecturas software. Determinar qué arquitectura entre las anteriores

Arquitectura software	Mantenibilidad	Memoria necesaria	Velocidad CPU requerida	Ancho de banda nec.	Transportabilidad
A	Alta	20MB	1 GHz	35Kbps	Unix, Windows
B	Alta	14MB	500 MHz	1Mbps	Windows sólo
C	Alta	8MB	2 GHz	2Kbps	Windows, Mac
D	Media	20MB	1 GHz	30Kbps	Unix sólo

se escogería si se tuvieran los siguientes objetivos y prioridades. Justificar la respuesta.

- (a) *Objetivos*: ejecución en Windows; trabajando con una conexión de 30Kbps o más rápida, con una máquina de 1 GHz o más rápida; necesitando no más de 25MB de memoria.
Prioridades Generales (mayor a menor): eficiencia de ancho de banda, CPU, portabilidad, optimización de memoria y mantenibilidad.
- (b) Lo mismo que el caso anterior, sólo que el ancho de banda baja al cuarto nivel de prioridad.

(c) *Objetivos:* se ejecuta en Unix, trabajando con una conexión de 1 Mbps o más rápida, trabajando con una máquina 500MHz o más rápida, necesitando no más de 40MB de memoria. *Prioridades generales(mayor a menor):* mantenibilidad, portabilidad, ancho de banda, velocidad de CPU y memoria.

27. Explicar ¿qué se entiende por Proceso de Negocio? ¿Cuáles son sus características?
28. ¿Por qué es tan importante elaborar un modelo de negocio antes de comenzar a obtener los requisitos del sistema informático que lo soportará?