

**INSTITUTO TECNOLÓGICO DE COSTA RICA**

Escuela de Ingeniería en Computación

Diseño de Software – II Semestre, 2016

Prof: Maria José Artavia

Valor 55 puntos – 20%

Tiempo: 1 hora y 50 minutos

**Primer examen parcial**

Nombre: \_\_\_\_\_ Calificación: \_\_\_\_\_

**Instrucciones generales:**

- Lea cuidadosamente cada ejercicio y basándose en la teoría y práctica vista en clase responda las preguntas indicadas.
- Todas las respuestas deben estar escritas con lapicero. NO tendrá derecho a reclamo si responde con lápiz.
- Si hay sospecha de copia, el examen será anulado y se enviará carta al expediente del estudiante.
- Mantenga su teléfono celular apagado durante el examen.

**Primera parte. Selección única**

Opción única, marque con una equis (X) la respuesta correcta (puntos, 1 punto c/u)

1. La cohesión se refiere a:
  - a. Medida cualitativa del grado en el que un módulo está conectado a otros y el mundo exterior.
  - ☒ b. Medida cualitativa del grado en el que un módulo se enfoca a una sola cosa.
  - c. Técnica que consiste en no centrarse en detalles concretos del diseño, sino hacer un esquema visual a alto nivel. ✗
  - d. Dividir el problema en varios problemas más pequeños para que el costo de resolverlos sea menor. ✗
  - e. Proceso de cambiar un sistema de software de tal forma que no se altere el comportamiento externo de su código y aún así se mejora su estructura interna. ✗

2. Un arquitecto de software define **prioridades** para las diferentes partes del sistema basándose en:
  - a. Las responsabilidades de cada parte, haciéndolos independientes y ocultando sus partes internas
  - b. La detección de problemas que aparecen en muchas partes del sistema, para considerarlos en conjunto
  - c. La congruencia entre las partes, dando balance interno, compatibilidad y armonía
  - ☒ d. Información proveniente del negocio, mercado, tecnología, limitaciones existentes y retos e impedimentos
  - e. Mantener un conjunto de decisiones lo más pequeño posible x
  
3. En cuanto a la priorización que debe hacer un arquitecto de software en un sistema debe tomar en cuenta:
  - a. Cuáles son las áreas en las cuales sobresalir, cumplir con todo lo solicitado, establecer cómo compensar situaciones de conflicto entre metas diferentes.
  - b. Habilidades del equipo del proyecto, no explorar áreas nuevas en las que se pueda sobresalir, cumplir con todo lo solicitado.
  - ☒ c. Cuáles son las áreas en las cuales sobresalir, identificar los requerimientos con los que se puede cumplir, establecer cómo compensar situaciones de conflicto entre metas diferentes.
  - d. Cuáles son las áreas en las cuales sobresalir, habilidades del equipo del proyecto,
  - e. Todas las anteriores
  
4. Algunos temas transversales en la arquitectura de software son:
  - a. Desempeño, seguridad, entorno
  - b. Seguridad, usabilidad, experiencia de usuario
  - c. Usabilidad, desempeño, entorno
  - d. Experiencia de usuario, entorno, desempeño
  - ☒ e. Todas las anteriores.
  
5. La abstracción es un proceso mental natural que nos permite:
  - a. Dominar problemas complejos del mundo real.
  - b. Encapsular la información del diseño y la ejecución de un programa.
  - c. Construir una vista simplificada de cómo funcionan las cosas.
  - d. Generar un modelo abstracto del problema, que es más sencillo que el real.
  - ☒ e. Todas las anteriores.
  
6. Si una entidad tiene alto acoplamiento se puede afirmar que:
  - ☒ a. Se resiente de los cambios en los elementos relacionados, es difícil de entender de manera aislada, es difícil de reutilizar.
  - ☒ b. No se ve afectada por cambios en los elementos relacionados, es fácil de entender de manera aislada, es fácil de reutilizar.
  - c. Se resiente de los cambios en los elementos relacionados, es fácil de entender de manera aislada, es fácil de reutilizar

- d. No se ve afectada por cambios en los elementos relacionados, es difícil de entender de manera aislada, es difícil de reutilizar.
  - e. Ninguna de las anteriores.
7. El acoplamiento se refiere a:
- ☒ a. Medida cualitativa del grado en el que un módulo está conectado a otros y el mundo exterior.
  - b. Medida cualitativa del grado en el que un módulo se enfoca a una sola cosa.
  - c. Técnica que consiste en no centrarse en detalles concretos del diseño, sino hacer un esquema visual a alto nivel.
  - d. Dividir el problema en varios problemas más pequeños para que el costo de resolverlos sea menor.
  - e. Proceso de cambiar un sistema de software de tal forma que no se altere el comportamiento externo de su código y aún así se mejora su estructura interna.
8. Al descomponer sistemas se deben seguir los siguientes principios:
- ☒ a. Alta cohesión, Bajo acoplamiento, ocultar la información.
  - b. ~~Baja cohesión~~, Alto acoplamiento, ocultar la información.
  - c. ~~Alta cohesión~~, ~~Alto acoplamiento~~, ocultar la información.
  - d. Alta cohesión, Bajo acoplamiento, no agrupar funcionalidades relacionadas.
  - e. ~~Baja cohesión~~, Alto acoplamiento, agrupar funcionalidades relacionadas.
9. El concepto de modularidad se refiere a:
- a. Medida cualitativa del grado en el que un módulo está conectado a otros y el mundo exterior. ✗
  - b. Medida cualitativa del grado en el que un módulo se enfoca a una sola cosa. ✗
  - c. Técnica que consiste en no centrarse en detalles concretos del diseño, sino hacer un esquema visual a alto nivel. ✗
  - ☒ d. Dividir el problema en varios problemas más pequeños para que el costo de resolverlos sea menor.
  - e. Proceso de cambiar un sistema de software de tal forma que no se altere el comportamiento externo de su código y aún así se mejora su estructura interna. ✗
10. En la arquitectura de 4+1 vistas la vista de despliegue se refiere a:
- ☒ a. Representa cómo están distribuidos los componentes entre los distintos equipos que conforman la solución incluyendo los servicios.
  - b. Mostrar los requisitos funcionales del sistema y de lo que debe hacer, las funciones y servicios definidos. ✗
  - c. Representa los flujos de trabajo del negocio y operacionales de los componentes que conforman el sistema. ✗
  - d. Mostrar cómo está dividido el sistema de software en componentes, y muestra las dependencias entre estos componentes.
  - e. Está representada por los casos de uso. ✗

**Segunda parte. Desarrollo (Valor 37 pts)**

Responda cada una de las siguientes preguntas:

1. Explique qué es el diseño arquitectural y el diseño de bajo nivel. *Valor 4 pts.*
2. Indique tres beneficios de la arquitectura de software. *Valor 3 pts*
3. Describa cuáles son las dos responsabilidades que contempla los patrones GRASP. *Valor 4 pts*
4. Explique de qué trata el patrón controlador. *Valor 2 pts*
5. Explique y ejemplifique tres olores de software. *Valor 9pts*
6. Elabore el diagrama de clases UML, con atributos, métodos, relaciones, multiplicidades, nombres de roles para el siguiente contexto: *(Valor 15 pts)*

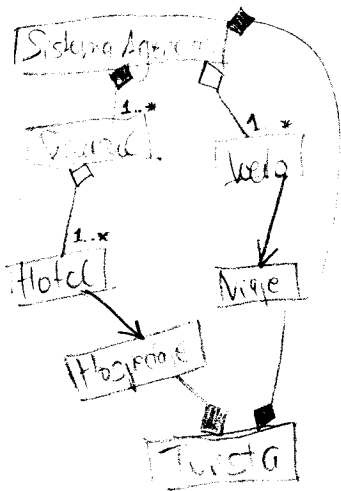
Una cadena de agencias de viajes desea disponer de una Base de Datos que contemple información relativa al hospedaje y vuelos de los turistas que la contratan.

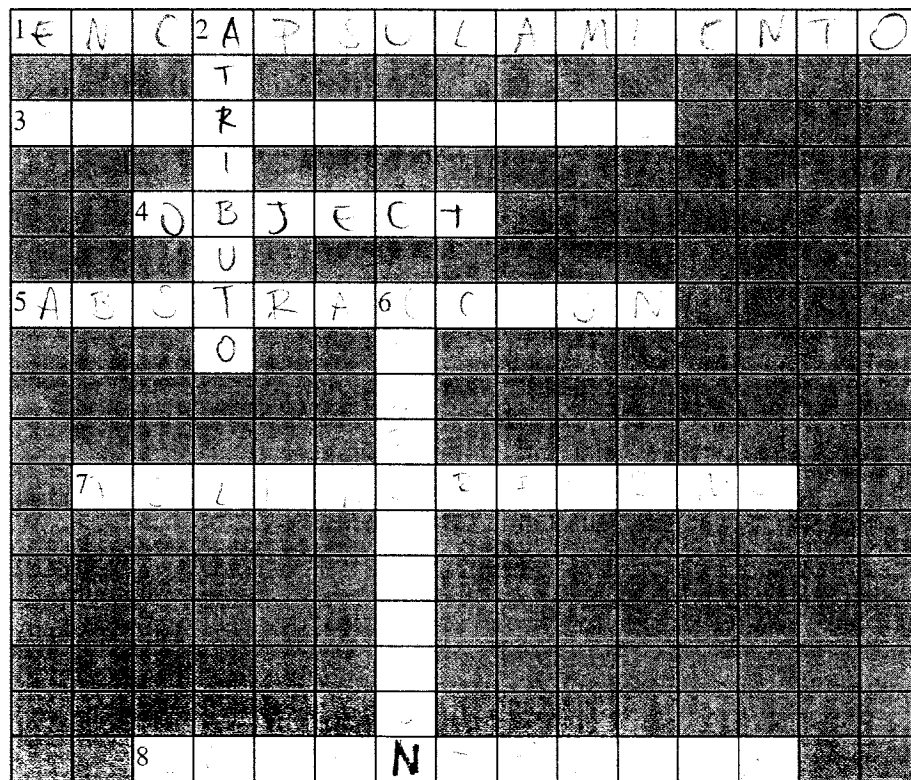
Los datos a tener en cuenta son:

- La cadena de agencias está compuesta por un conjunto de sucursales. Cada sucursal viene definida por el código de sucursal, dirección y teléfono.
- La cadena tiene contratados una serie de hoteles de forma exclusiva. Cada hotel estará definido por el código de hotel, nombre, dirección, ciudad, teléfono y número de plazas disponibles.
- De igual forma, la cadena tiene contratados una serie de vuelos regulares de forma exclusiva. Cada vuelo viene definido por el número de vuelo, fecha y hora, origen y destino, plazas totales y plazas de clase turista de las que dispone.
- La información que se desea almacenar por cada turista es el código de turista, nombre y apellidos, dirección y teléfono.

Por otra parte, hay que tener en cuenta la siguiente información:

- A la cadena de agencias le interesa conocer que sucursal ha contratado el turista.
- A la hora de viajar el turista puede elegir cualquiera de los vuelos que ofrece la cadena, y en que clase (turista o primera) desea viajar.
- De igual manera, el turista se puede hospedar en cualquiera de los hoteles que ofrece la cadena, y elegir el régimen de hospedaje (media pensión o pensión completa). Siendo significativa la fecha de llegada y de partida.



**Tercera parte. Crucigrama. (Valor 8 pts)****Vertical**

2. Información que representa el estado de un objeto.
6. Es la relación estructural más fuerte.

**Horizontal**

1. Este concepto permite cambiar el comportamiento interno de una clase y todo aquello que no es visible para el resto del sistema.
3. Declaran el comportamiento de una clase.
4. Corresponde a un elemento concreto creado a partir de una plantilla o molde.
5. Incluye elementos importantes al mismo tiempo que excluye los detalles irrelevantes de una representación.
7. Capacidad de responder de múltiples formas a un mensaje.
8. Relación particular no estructural.