

IIC 2143 Ingeniería de Software

Interrogación 3 - Semestre 2/2018 - Sección 02

Responda cada pregunta en hoja separada

Entregue una hoja con su nombre para cada pregunta, aunque sea en blanco

Recuerden que se encuentran bajo el código de honor

1. (20 pts) Dos de los patrones básicos muy utilizados que estudiamos en clases son Fábrica (Factory) y Fábrica Abstracta (Abstract Factory). Respecto a ellos conteste las siguientes preguntas en la forma mas corta y precisa posible:
 - a) Por qué se necesita "Factory" cuando podríamos simplemente usar constructores
 - b) Muestre un ejemplo concreto de "Factory" (diagrama de clases y breve explicación) distinto al que se vio en clases
 - c) Por qué se necesita "Abstract Factory" y cual es la diferencia con "Factory"
 - d) Muestre un ejemplo concreto de "Abstract Factory" (diagrama de clases y breve explicación) distinto al que se vio en clases

2. (30 pts) Se quiere escribir una pieza de código que sirva para calcular números primos sacando partido del conocido patrón de diseño Estrategia (Strategy). La idea es que podamos tener un algoritmo de cálculo separable del resto, de modo de poder calcular los números primos menores a 30 con a lo menos los siguientes dos algoritmos:
 - algoritmo hard_coded – los primeros números primos están guardados en forma estática en una lista
 - algoritmo standard – se verifica si el número es divisible por cada uno de los números comenzando con 2.

La idea es poder hacer algo como lo siguiente:

```
primos = Primos.new(Estrategia1.new)
primos.buscar
primos = Primos.new(Estrategia2.new)
primos.buscar
```

- a) (10 pts) Describa usando UML la solución de este problema usando el patrón Strategy.
- b) (20 pts) Escriba el código completo de la implementación en Ruby para que funcione como en el ejemplo

3. (30 pts) Recuerda la aplicación para jugar el juego "The Trivium". Queremos ofrecer esta aplicación bajo la modalidad software-as-a-service, de manera que los jugadores puedan estar geográficamente distribuidos (p.ej., uno en Santiago, otro en Arica, otro en Buenos Aires, etc.) y acceder al juego a través de la Internet empleando un browser. Considera el lado del servidor de esta aplicación.
- a) (5 pts) ¿Cuáles aspectos/funcionalidades/responsabilidades de la aplicación pertenecen al servidor y por qué? Sea tan específico como puedas con respecto al juego "The Trivium".
 - b) (5 pts) ¿Qué patrón arquitectónico es recomendable para el servidor y por qué? Enuncie —da su nombre y describalo de manera breve y precisa— un criterio o principio de diseño que guíe esta recomendación.
 - c) (5 pts) ¿Cuáles aspectos/funcionalidades/responsabilidades del servidor, identificados en a), quedan mejor ubicados en cada uno de los componentes del patrón arquitectónico identificado en b)?
 - d) (5 pts) Queremos organizar el código de la aplicación propiamente tal siguiendo el patrón arquitectónico modelo-vista-controlador (MVC). ¿Por qué queremos hacer esto? Responda de manera precisa.
 - e) (10 pts) Explique el rol que desempeña cada componente del patrón MVC, pero no en general, sino que particularmente durante la ejecución del caso de uso Jugar:
 - 1. El Jugador que tiene el turno lanza el dado
 - 2. El Juego muestra el resultado del lanzamiento del dado
 - 3. El Juego muestra las casillas a las que el Jugador puede mover su ficha
4. (20 pts) Considere una aplicación que permita a los clientes de un banco realizar todo tipo de transacciones desde consultas de saldo y estados de cuenta, pasando por compras tradicionales e inversiones en la bolsa, y hasta el pago de multas e incluso de los impuestos usando cualquier tipo de dispositivo conectable a Internet desde un computador personal, pasando por una tablet, y hasta un teléfono celular o un reloj. Por supuesto, esta debe ser una aplicación que tenga toda la funcionalidad necesaria y además ser de fácil de usar, y, dada su naturaleza, también debe ser muy confiable y tener excelente desempeño.
- Con respecto a la actividad de análisis arquitectónico durante el desarrollo de esta aplicación, responde:
- a) Identifica, y justifica que es así, dos puntos de variación.
 - b) Identifica, y justifica que es así, dos puntos de evolución.
 - c) Define dos escenarios de calidad, más allá de los obvios sobre desempeño y tiempo medio entre fallas.
 - d) Con respecto a los patrones arquitectónicos cliente-servidor y peer-to-peer, ¿cuál es el más apropiado en este caso y por qué?