Nombre _____

Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

8 de mayo de 2018

IIC2143 – Ingeniería de Software

Interrogación 2

Instrucciones:

- Sea preciso: no es necesario escribir extensamente pero sí ser preciso.
- En caso de ambigüedad, utilice su criterio y explicite los supuestos que considere convenientes.
- Responda y entregue cada pregunta en hojas separadas. Si no responde una pregunta debe entregar de todas formas la hoja correspondiente a la pregunta. Indique su nombre en cada hoja de respuesta.
- Esta interrogación fue diseñada para durar 120 minutos.

1. (0.6 pts)
   a. ¿Cuáles son los problemas de sobreestimar un proyecto?
      - El trabajo se expandirá hasta usar todo el tiempo disponible
      - Al comienzo del proyecto se puede malgastar el tiempo

   b. ¿Cuáles son los problemas de subestimar un proyecto?
      - Dificultad para planear y coordinar el proyecto
      - Tensión permanente con el cliente
      - Se priorizan los entregables sobre la calidad

   c. Explique qué caso genera más repercusiones en un proyecto de *software*.

      Subestimar genera más repercusiones sobre un proyecto que sobreestimar, ya que sus consecuencias no son lineales debido a la aparición de errores, *bugs* y prácticas riesgosas con impacto negativo que no es cuantificable. En cambio al sobreestimar se desperdiciarán recursos pero de manera controlada y cuantificable.

Nombre _____

2. (1.0 pts)

Usted es el *product owner* del proyecto *MisCursos*, plataforma de manejo de cursos para usarse en una universidad chilena. El alcance completo del proyecto incluye:

a. que un alumno sea capaz de construir paquetes de cursos a partir de los horarios publicados por la universidad
b. que un alumno pueda seleccionar uno de los paquetes anteriores para inscribir los cursos de su semestre

Considere que el proyecto comenzó a inicios de septiembre 2017, cuenta con un equipo de 5 desarrolladores y se planificó para que esté operando a finales de junio 2019. Hasta la fecha se han concretado 3 *sprints* de distinta duración, con los siguientes avances:

- Sprint 1: en 2 meses se realizó 10% del alcance
- Sprint 2: en 1 mes se realizó 10% del alcance
- Sprint 3: en 5 meses se realizó 20% del alcance

Considerando lo anterior:

a. Estime a partir de los datos proporcionados la productividad histórica del equipo y la de un desarrollador promedio, medidas en porcentaje de alcance por mes. En base a los datos históricos y el método de estimación PERT (e = (a + 4m + b) / 6), explique si el proyecto está atrasado o a tiempo.

    - La productividad histórica del equipo es de 5% mensual.
    - La productividad histórica de un desarrollador es de 1% mensual.
    - (0.04 + 4 * 0.05 + 0.1) / 6 = 5.667%
    - En base a esta estimación, en los 13 meses restantes se lograría hacer un 73.67% del alcance del proyecto. Considerando llevan 40% ya del alcance completo, no estaría atrasado.

b. Ahora suponga que el plazo se adelanta para enero 2019, que es cuando se publican los horarios de los cursos para el año 2019. Considerando las restricciones de tiempo, presupuesto y alcance: ¿cuál negociaría con el cliente del proyecto? Justifique su elección frente a las otras alternativas.

    - No se puede negociar tiempo porque esa es la nueva restricción del cliente.
    - Negociar presupuesto es una alternativa, pero es riesgoso y no garantiza que se logren los objetivos.
    - Negociar alcance es la alternativa menos riesgosas y que permite que se cumpla con parte de las expectativas del proyecto para la nueva fecha.

c. Luego de discutir el cambio anterior con su equipo, este le garantiza que podrán cumplir con todas las funcionalidades de *MisCursos* para enero 2019 si se elige una de las siguientes alternativas:

Nombre _____

- continuar con el desarrollo de todas las funcionalidades pero de manera muy simplificada y bajo los compromisos acordados, para luego completar o volver a hacer lo que sea necesario.
- agregar 3 desarrolladores al equipo, para así aumentar la productividad.

¿Qué opina de las alternativas descritas? ¿Qué decisión tomaría usted en esta situación?

- Continuar con el desarrollo de todas las funcionalidades impactará negativamente la calidad, lo que podría generar que el *software* no logre con lo mínimo esperado. Por otra parte, rehacer completamente un *software* es una medida costosa que se debería evitar.
- Agregar más gente a un proyecto que ya está atrasado puede hacer que este se atrase aún más. Esto se debe a la complejidad del trabajo y comunicación en equipos más numerosos, como también al tiempo y esfuerzo del proceso de inducción al proyecto de los nuevos integrantes.

  Dado que ninguna de las alternativas presentadas es adecuada, la decisión que parece más sensata es negociar el alcance del proyecto para esa nueva fecha y así disminuir las funcionalidades a desarrollar.

Nombre _____

3. (1.0 pts) Comente qué describe cada una de las vistas del modelo 4+1 propuesto por Philippe Kruchten. Nombre un diagrama perteneciente a cada vista.

Vista lógica: Describe la estructura y funcionalidad del sistema. Los interesados son los usuarios finales.

Diagrama de clases

Vista de procesos: Trata los aspectos dinámicos del sistema, explica los procesos y cómo se comunican. Se enfoca en el comportamiento del sistema en tiempo de ejecución.

Diagrama de actividad

Vista de implementación: Ilustra el sistema desde la perspectiva de los desarrolladores, y está enfocada en la administración de los artefactos de software.

Diagrama de componentes

Vista física: Describe el sistema desde el punto de vista de un ingeniero de sistemas. Ilustra la topología de componentes de software en la capa física, así como las conexiones entre ellos.

Diagrama de despliegue

Vista de escenarios: Descripción holística que se utiliza para identificar y validar el diseño de la arquitectura.

Diagrama de casos de uso

Nombre _____

4. (0.7 pts) Relacione los nombres de los siguientes patrones *GoF* con su descripción correspondiente (utilice la letra que lista cada patrón):

a. Decorator
b. Strategy
c. Factory Method
d. Adapter
e. Observer
f. Composite
g. Command

d. Encapsula un objeto para exponer una nueva interfaz de este

__ Permite crear objetos de distintas familias sin especificar sus clases

f. Clientes interactúan con colecciones de objetos y objetos individuales uniformemente

e. Permite notificar a otros objetos cuando el estado de uno cambia

b. Encapsula comportamiento intercambiable dinámicamente

__ Simplifica la interfaz de un grupo de interfaces

g. Encapsula una petición en un objeto

a. Añade funcionalidad a un objeto dinámicamente

c. Subclases deciden qué clase en concreto se crea

__ Encapsula un objeto para controlar su acceso

__ Garantiza que una clase tenga solamente una instancia

Nombre _____

5. (1.5 pts) Considere el siguiente escenario:

"Las corredoras de propiedades ofrecen como servicio intermediar la gestión de arriendos de propiedades entre dueños y arrendatarios. Es decir, el dueño de una (o más) propiedad(es) puede contratar a una corredora para que administre el arriendo de sus propiedades. Para esto, el dueño de una propiedad otorga un poder a la corredora, con el cual esta puede buscar arrendatarios y firmar contratos de arriendo por un plazo determinado sobre la propiedad. Al iniciar el arriendo, el arrendatario entrega dinero en forma de garantía a la corredora. Al finalizar el arriendo, esta garantía es reembolsada al arrendatario siempre y cuando este no tenga deudas sobre el arriendo y la propiedad esté en las mismas condiciones que cuando fue entregada. Cada fin de mes, la corredora contacta a los arrendatarios para que paguen el mes siguiente de arriendo. Una vez recibidos los pagos, la corredora entrega el dinero correspondiente a cada dueño de una propiedad, descontando los honorarios por el servicio. Si un arrendatario pagase el arriendo luego de la fecha de vencimiento estipulada en el contrato, esto generaría una multa que deberá ser pagada junto al pago del mes siguiente.

La corredora *PreCasa* está interesada en facilitar la gestión de sus arriendos a través de un sistema de *software*. Para ello, se necesita modelar la situación descrita anteriormente, considerando como requisito mínimo que el sistema pueda generar balances sobre las cuentas de las propiedades y sus arriendos."

Realice un modelo de dominio para representar los elementos involucrados en la solución del problema descrito. Su diagrama debe ser *UML 2.0* válido, detallando las entidades junto con sus atributos relevantes y los tipos de relaciones entre entidades con su cardinalidad. Sea explicito en cómo su diseño incluye todas las consideraciones mencionadas.

El diagrama de modelo de dominio en ***UML*** debe incluir los siguientes elementos:

Entidades:
- Dueño
- Corredora
- Contrato
- Arrendatario
- Propiedad
- Arriendo

Atributos (se debe reflejar lo siguiente, no es necesario que existan los mismos nombres pero sí que se refleje esto):
- Propietario id
- Comisión en % o plata
- Precio arriendo
- Duración contrato
- Garantía

Las relaciones entre entidades deben tener un nombre y cardinalidad atingentes.

Nombre _____

6. (1.2 pts) Considere el siguiente escenario:

"La empresa *PreCasa* quiere rediseñar su gestor de correos, para así mejorar su comunicación entre empleados de la compañía, o bien con sus clientes. Algunos de los requisitos que la empresa tiene claros son que dependiendo de a quién esté dirigido un correo (por ejemplo, si el destinatario es una cuenta interna o un cliente de la compañía) se debe agregar un texto de advertencia al final. También, si el sistema detecta que la información contenida en un correo es confidencial, los correos se deben encriptar de manera segura. Además, dependiendo del cargo del emisor del correo se debe agregar una firma al pie del correo, estandarizada por la empresa. Por último, a la compañía le gustaría contar con distintos diseños para los correos, pero no implementaría esta funcionalidad hasta futuros desarrollos."

Escoja 1 patrón de diseño *GoF* que aporte al diseño de la solución, justifique su elección y realice un diagrama de clases *UML 2.0* para representar su implementación. Este diagrama debe ser explícito y detallado de los componentes (y cómo se relacionan entre ellos). Explique cómo su diseño permitiría agregar la funcionalidad que *PreCasa* tiene intención de implementar en futuros desarrollos.

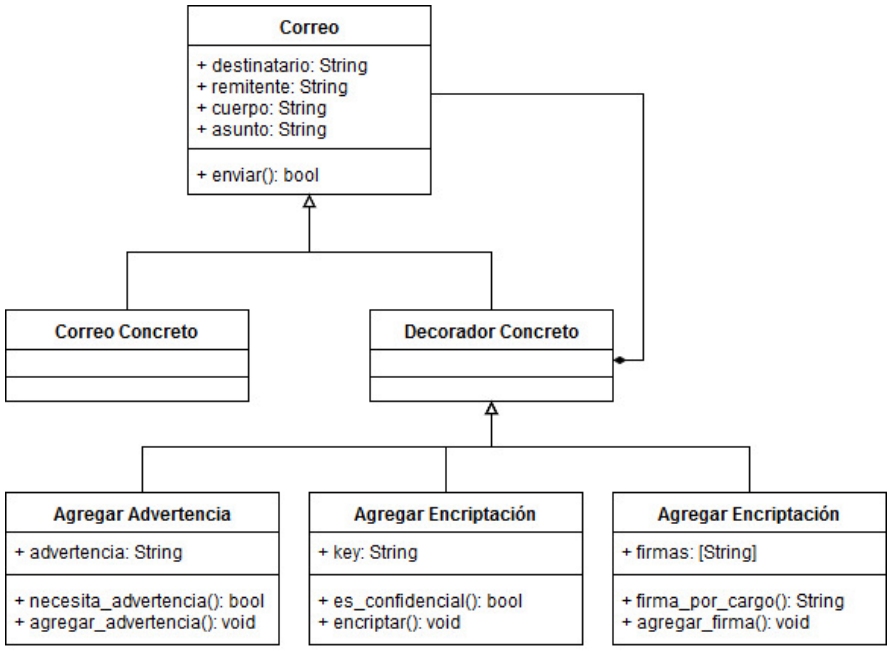
Ejemplos respuestas:

i) Elegiría el de decoradores ya que debo agregar distintas cosas al mail y me permite luego implementar, agregando nuevas subclases de decoradores, los distintos diseños. El decorador permite añadir dinámicamente funcionalidades, lo que es necesario para que detecte y cambie

ii) Ocuparía el patrón decorador, ya que tenemos un email base, al que le deben agregar cosas a veces, entonces, el sistema chequea el mail, si ve que el contenido es confidencial, le pone el decorador para encriptar, y así con las demás funcionalidades. Para los distintos diseños bastaría hacer un decorador para cada uno.

Un diagrama correcto es el siguiente:

Nombre _____


Correo
+ destinatario: String
+ remitente: String
+ cuerpo: String
+ asunto: String
+ enviar(): bool
Correo Concreto
Decorador Concreto
Agregar Advertencia
+ advertencia: String
+ necesita_advertencia(): bool
+ agregar_advertencia(): void
Agregar Encriptación
+ key: String
+ es_confidencial(): bool
+ encriptar(): void
Agregar Encriptación
+ firmas: [String]
+ firma_por_cargo(): String
+ agregar_firma(): void