

IIC 2143 Ingeniería de Software  
Interrogación 3 - Semestre 1 /2018 - Sección 02: Prof. J. Navón

**Tiempo: 2 horas**

**Contestar cada pregunta en hoja separada**

**Pregunta 1 - 20 pts (Patrones de Diseño)**

Un software de gestión bastante complejo incluye una componente destinada a generar reportes impresos. El problema es que el software debe manejar distintos formatos de reporte dependiendo si este está destinado a Chile, USA, Brasil, o alguno de otros 10 países donde se utiliza. Cada reporte incluye títulos de nivel 1, 2 y 3, párrafos, tablas y gráficos. Cada tipo de reporte incluye un set particular de dichos elementos.

- a) (5 pts) ¿Cual es el patrón de diseño más apropiado para este problema ? Fundamente su respuesta
- b) (10 pts) Haga un diagrama de clases UML que ilustre su solución para solo 2 tipos de reporte (Chile y USA) y 3 tipos de elementos (párrafos, tablas y gráficos). Escriba el código Ruby de las distintas clases del diagrama (solo lo mínimo necesario)
- c) (5 pts) Escriba un trozo de código Ruby que permita demostrar la forma de usar lo anterior para generar los reportes de Chile y USA mencionados

**Pregunta 2 - 20 pts (Arquitectura)**

Conteste brevemente no más de 4 líneas

- a) (4) Cuando se menciona la arquitectura de un producto de software ¿de que se está hablando realmente?
- b) (4) ¿En qué influye finalmente el diseño o la selección de una arquitectura u otra ?
- c) (4) ¿En qué consiste una arquitectura orientada a servicios (SOA) y cual es la diferencia entre las arquitecturas SOA y la de Microservicios?
- d) (4) ¿Qué se entiende por DevOps y qué tienen que ver con esto los microservicios ?
- e) (4) Explique la arquitectura de capas (layers) utilizando un ejemplo (ojalá uno conocido)

**Pregunta 3 - 20 pts (Testing)**

Considere un módulo de software que debe aceptar el nombre de un ítem de supermercado y una lista de los diferentes tamaños (máximo 5) en que se ofrece expresados en onzas (números enteros de 1 a 48). El nombre debe ser alfabético de 2 a 15 caracteres. La lista de tamaños debe ser ingresada en orden ascendente. El nombre del ítem se ingresa primero seguido de una coma y luego la lista de tamaños separados por comas. Los espacios en blanco son ignorados. Por ejemplo: Chips, 1, 2, 4, 10.

Usando técnicas de caja negra, diseñe cuidadosamente un set de pruebas mínimo que permita testear este software. (Ayuda: definir las clases de equivalencia primero)

#### **Pregunta 4 - 20 pts (SQA)**

Indique si las afirmaciones siguientes son verdaderas o falsas. Justifique brevemente

- a) (4) El análisis estático de código ha demostrado ser más eficiente que el testing en términos del costo de encontrar defectos
- b) (4) Las pruebas unitarias siempre son de caja negra
- c) (4) La estrategia de integración top down es más compleja que la estrategia opuesta por la necesidad de construir "stubs"
- d) (4) Un test A/B tiene más que ver con validación que con verificación
- e) (4) Un test de regresión es lo mismo que un test de integración

#### **Pregunta 5 - 20 pts (Arquitectura)**

En la interrogación 2 se le pidió construir un modelo de dominio para una aplicación destinada a manejar información de pruebas anteriores. Para su conveniencia se reproduce el enunciado a continuación:

El CAI ha decidido construir una aplicación web para que los alumnos puedan consultar sobre preguntas de pruebas de semestres anteriores de algún ramo específico. Un curso en un año y semestre dado puede haber tenido más de una sección, cada una con distinto profesor. En esa ocasión cada prueba puede haber un número de interrogaciones (típicamente 3) y un examen final. Cada prueba tiene un número variable de preguntas cada una de las cuales tiene un puntaje asociado. La prueba completa también tiene un puntaje máximo asociado. Así por ejemplo, esta pregunta (I2 de IIC2143 primer semestre de 2017, sección 02 (Prof. Navón) vale 20 puntos y la prueba 100. Las preguntas pueden además ser etiquetadas por los alumnos de acuerdo a características (desarrollo, UML, difícil). No hay límite en el número de etiquetas que pueden asignarse a una pregunta.

Ahora se pide bosquejar una arquitectura para este software en forma de dos diagramas

- a) (5 pts) Un diagrama de contexto (context)
- b) (15 pts) Un diagrama de contenedores (containers). Explique brevemente cada uno de ellos.

Justifique brevemente la arquitectura elegida