

DESARROLLO DE SOFTWARE

TERCER EXAMEN PARCIAL

Prof. Carlos Alonzo / 15 de enero de 2026

Nombre : _____

Instrucciones :

- 1. El examen es INDIVIDUAL. Evite penalizaciones y sanciones por plagios, copias o copy/paste de respuestas de terceros.**
- 2. El enunciado estará disponible desde las 8:50am del día jueves 15 de enero 2026**
- 3. El examen debe entregarse, como máximo, a las 11:10am del día jueves 15 de enero 2026**
- 4. El examen se debe responder en máximo 4 páginas de tamaño carta. NO se necesita de más espacio.**
- 5. IMPORTANTE: DEBE ENTREGAR SUS SOLUCIONES ESCRITAS A MANO, EL EXAMEN ES MANUSCRITO. ESCRIBA OSCURO PARA QUE EL PROFESOR PUEDA LEER SUS RESPUESTAS CON CLARIDAD. EVITE LAPICES QUE ESCRIBEN MUY CLARO.**
- 6. Debe armar un archivo PDF unificado con todas sus respuestas a las preguntas debidamente identificadas.**
- 7. El nombre del archivo PDF debe ser NOMBRE+APELLIDO+CEDULA.pdf. Por ejemplo, CarlosAlonzo13613542.pdf.**
- 8. Escriba sus respuestas con claridad, cuide la ortografía y su redacción.**
- 9. Deben respetar los límites de páginas de sus respuestas. En caso contrario, será penalizado.**
- 10. Debe enviar su respuesta al parcial por bandeja de entrada de Módulo 7. Si no lo entrega por esa vía, se considera que NO presentó el examen parcial.**
- 11. El subject del mensaje por Modulo 7 debe estar identificado, de igual manera, por su nombre, apellido y cédula de identidad.**
- 12. Todas sus respuestas deben ser el resultado de su propia elaboración en base a sus conocimientos, razonamientos, aportes y experiencias del proyecto.**
- 13. PROHIBIDO TERMINANTEMENTE EL USO DE HERRAMIENTAS DE IA.**
- 14. PROHIBIDO TERMINANTEMENTE EL USO DE APUNTES, LIBROS O CUALQUIER RECURSO DE INTERNET.**

EJERCICIO 1: AOP + SOLID + Mantenibilidad (5 puntos)

Durante este semestre hemos trabajado la propuesta del autor Mark Seemann de su implementación de AOP basado en SOLID. **[3 puntos]** Explique **por qué** Seemann concluye que el número de abstracciones que diseñamos es un mal indicador para medir la mantenibilidad del código. **[2 puntos]** **¿ Cuál es la postura del Profesor Carlos Alonzo con respecto a este punto ?**

Debe explicar con sus propias palabras, con argumento sólido y con claridad. Haga mención al ejemplo que Seemann indica. Fundamente su respuesta teniendo en cuenta **qué** significa la **mantenibilidad** como criterio de calidad de software. Tome en cuenta todo lo discutido en las clases durante el semestre.

Su respuesta no debe pasar de **15 líneas o media página tamaño carta.**

EJERCICIO 2 : Patrones DDD + Arquitectura Hexagonal (5 puntos)

Considere que usted está desarrollando un proyecto en Typescript usando DDD y Arquitectura Hexagonal. El proyecto contempla un objeto del modelo de dominio que representa un curso, que posee, al menos, tres atributos: nombre, que es un texto (string) que no puede contener números; fecha de creación y fecha de publicación. La fecha de publicación debe ser posterior a la fecha de creación.

Desarrolle, en Typescript, la solución para re-construir un objeto curso en base a los datos que puedan venir desde un controlador web (un endpoint) en la capa de infraestructura. Este endpoint recibe por parámetros el nombre del curso, la fecha de creación y la fecha de publicación. Para ello debe hacer uso del patrón táctico Factory y usar el patrón DTO (Data Transfer Object).

- a) Debe responder codificando **todo** en TypeScript.
- b) **[2 puntos]** Debe codificar el factory y el agregado curso.
- c) **[1 punto]** Debe garantizar invariantes
- d) Si usted codifica en su solución otras clases, esas clases NO serán evaluadas.
- e) **[2 puntos]** **¿ Con cuál otro patrón/técnica comparó el Profesor Carlos Alonzo al DTO ? Explique brevemente.**

Su respuesta no debe pasar de **30 líneas o una página tamaño carta.**

EJERCICIO 3 : Patrones de Diseño + Testing (5 puntos)

Durante el semestre hemos definido nuestra Arquitectura de Pruebas con tres niveles. El nivel inferior correspondía al **Object Mother**. Este patrón corresponde a una clase que contiene varios **Factory Method** para crear los objetos para su uso en las pruebas. Esto permite que las pruebas sean más legibles. Con el paso del tiempo, el Object Mother puede volverse demasiado desordenado, lo que afecta su mantenibilidad, debido a una infinidad de métodos muy detallados.

Usted debe proponer una mejora del Object Mother utilizando uno de los siguientes patrones de diseño creacionales : **Abstract Factory**, **Builder** o **Prototype**. Es decir, usted debe proponer una solución que mejore el patrón Object Mother. Para ello debe seleccionar uno de los tres patrones indicados. Solo uno de los tres patrones es el adecuado.

[4 puntos] Debe construir su solución propuesta en Typescript. Construya su solución utilizando el dominio del proyecto Kahoot que ha desarrollado este semestre. Justifique brevemente su solución. **[1 punto]** ¿ Cuál fue el autor que utilizó el Profesor Carlos Alonzo para diseñar la arquitectura de pruebas propuesta en clases ?

Su respuesta no debe pasar de 30 líneas o una página tamaño carta.

EJERCICIO 4 : Diagrama de Arquitectura a capas (Clean / Hexagonal) (5 puntos)

Para este último ejercicio usted deberá demostrar su “maestría” para asignar responsabilidad en distintas capas. Escoja la arquitectura que prefiera: si trabajó en Flutter con Clean Architecture puede usar las capas de presentación, dominio y datos; si trabajó en el backend puede usar Arquitectura Hexagonal con capas de dominio, aplicación e infraestructura. Lo importante es que usted cumple el objetivo del ejercicio: *demonstrar que ha aprendido a separar las responsabilidades entre objetos que colaboran entre sí en diversas capas.*

Enunciado del problema: Considere que desarrolla una aplicación web o móvil donde se debe mostrar el listado de todos los Hoteles, que se monstrarán en una vista. La información que se almacena asociada a cada hotel es: nombre, dirección, página web, país. Considere que la base de datos donde se persisten los datos esta montada en un servicio externo, en este caso Firebase, que se accede mediante un API.

[5 puntos] Usted debe entregar el diagrama de la arquitectura en capas, definiendo cuáles clases hay en cada capa y sus relaciones. Los métodos NO son necesarios, solo las clases con su nombre (con Lenguaje Ubícuo) y las relaciones entre las clases.

Su respuesta no debe pasar de **30 líneas o una página tamaño carta.**