|  |  |  |
| --- | --- | --- |
|  | Módulo: Aplicación de Técnicas de Ingeniería de Software  Tema: Segundo Examen Práctico  Grupo: DS321-A  Fecha:18 de octubre de 2025 | **Nota:** |
| Estudiante: Diego Mauricio Mejia Pérez |

**Indicaciones:** solucionar el examen según el caso planteado con Java. Además, deberá crear un repositorio en GitHub donde tendrá el ejercicio resuelto, deberá subir el enlace del repositorio. También deberá grabar un video (será evaluado) explicando la solución y subir el enlace del video junto que el enlace del repositorio al respectivo buzón en la plataforma.

**Gestor de Reservas de Aulas ITCA.**

La escuela ITCA desea desarrollar un sistema en consola que permita gestionar las reservas de aulas para clases, prácticas y eventos internos. Cada reserva debe cumplir reglas específicas según su tipo, y el sistema debe garantizar que no existan conflictos de horarios, además de permitir operaciones de registro, búsqueda, modificación, cancelación y generación de reportes.

**Requisitos técnicos:**

El programa debe implementarse utilizando el paradigma de Programación Orientada a Objetos. Debe incluir las siguientes clases, estructuras y características:

* Clases: Aula, Reserva (abstracta), ReservaClase, ReservaEvento y ReservaPractica.
* Categorias: TipoAula {TEORICA, LABORATORIO, AUDITORIO }, TipoEvento { CONFERENCIA, TALLER, REUNION }.
* Interfaz: Validable.
* Uso de encapsulamiento, herencia y polimorfismo.
* Implementación de excepciones personalizadas.
* Uso de estructuras de control (if, switch, bucles, try/catch).
* Uso de colecciones ArrayList y filtros mediante Streams.
* Interacción mediante consola con Scanner.
* Validación de fechas y horas con LocalDate y LocalTime.

Requisitos funcionales:

1. Gestión de aulas: registrar, listar y modificar.
2. Registro de reservas: alta de reservas de tipo clase, evento y práctica con validaciones completas.
3. Búsqueda, modificación y cancelación de reservas, con control de estados (activa, cancelada, histórica).
4. Reportes: top 3 aulas con más horas reservadas, ocupación por tipo de aula y distribución por tipo de reserva.
5. . Persistencia en archivos CSV o TXT.
6. Validaciones robustas de entrada con mensajes descriptivos.
7. Exportación de reportes a archivo de texto.
8. Ordenamiento configurable de resultados.
9. Búsqueda por texto en campo responsable.

**Rúbrica de Evaluación**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Criterio** | **Excelente (5)** | **Bueno (4)** | **Regular (3)** | **Deficiente (1-2)** |
| Modelado POO (20 pts) | Diseño completo con herencia, polimorfismo, encapsulamiento y uso adecuado de interfaces y enums. | Diseño correcto con leves omisiones en encapsulamiento o uso de herencia. | Uso básico de clases con algunos errores conceptuales en POO. | No aplica principios POO o el diseño es incorrecto. |
| Estructuras de control (15 pts) | Implementa correctamente estructuras de decisión, bucles y manejo de excepciones. | Usa estructuras de control correctamente con mínimos errores de lógica. | Estructuras parcialmente correctas o con errores frecuentes de validación. | Uso incorrecto o inexistente de estructuras de control. |
| Reglas de negocio (20 pts) | Cumple todas las validaciones de horarios, solapamientos y condiciones de tipo de aula. | Cumple la mayoría de las validaciones con leves inconsistencias. | Varias validaciones ausentes o incorrectas. | No aplica las reglas de negocio. |
| Gestión CRUD (15 pts) | Implementa correctamente registro, modificación, cancelación y listados. | CRUD funcional con leves errores en alguna operación. | Solo algunas operaciones CRUD funcionan correctamente. | CRUD incompleto o con errores críticos. |
| Claridad en la explicación (10 pts) | Explica claramente la estructura del código, el propósito de las clases, métodos y reglas aplicadas. | Explica la mayoría de los componentes de forma entendible, con leves omisiones. | Explicación poco clara o incompleta en varios aspectos. | No explica adecuadamente el funcionamiento o hay confusión total. |
| Dominio del tema (10 pts) | Demuestra dominio completo de POO, estructuras de control y reglas implementadas. | Demuestra dominio general con pequeñas imprecisiones. | Presenta varias confusiones o imprecisiones sobre los conceptos básicos. | No demuestra comprensión del código ni de los conceptos. |
| Presentación y comunicación (10 pts) | Expone con buena dicción, orden lógico y demuestra el código funcionando. | Presenta de forma comprensible con leves problemas de fluidez. | Presentación poco estructurada o con fallas notables de comunicación. | Presentación desorganizada, difícil de seguir o incompleta. |