

Materia:	Programación II - DIVISION 323		
Nivel:	2º Cuatrimestre		
Tipo de Examen:	Final		
Apellido ⁽¹⁾ :		Fecha:	
Nombre/s ⁽¹⁾ :		Docente a cargo ⁽²⁾ :	
División ⁽¹⁾ :	323	Nota ⁽²⁾ :	
DNI ⁽¹⁾ :		Firma ⁽²⁾ :	

(1) Campos a completar solo por el estudiante en caso de imprimir este enunciado en papel.

(2) Campos a completar solo por el docente en caso de imprimir este enunciado en papel.

Objetivo:

Desarrollar un sistema en Java que permita gestionar entidades relacionadas a un contexto definido por el alumno, siguiendo los principios de la Programación Orientada a Objetos y utilizando los conocimientos adquiridos durante el curso. El proyecto debe incluir las siguientes características técnicas y funcionales.

Requisitos del Proyecto

1. Contexto del Proyecto

Se podrá elegir un contexto temático para el sistema, el cual debe estar relacionado con una de las siguientes áreas:

- Gestión de productos (negocio, inventario, etc.)
- Gestión de personas (empleados, estudiantes, clientes, etc.)
- Gestión de vehículos (autos, barcos, aviones, etc.)
- Gestión de entidades creativas (naves, monstruos, aliens, personajes, etc.)

Ejemplo: Si eligen "vehículos", podrían implementar una clase abstracta Vehiculo con derivadas como Auto y Moto.

2. Clases y Herencia

- Implementar una **clase abstracta base** que defina los atributos y comportamientos comunes a todas las entidades del sistema.
- Crear al menos **tres clases derivadas** de esta clase base.
- Algunas de las clases derivadas deben implementar una **interfaz con al menos un método abstracto**.

3. CRUD (Crear, Leer, Actualizar, Eliminar)

- Implementar una **clase de gestión** que contenga una lista genérica para administrar las entidades del sistema.

- Esta clase debe implementar una **interfaz genérica** que incluya métodos para realizar las operaciones del CRUD.

4. Persistencia de Datos

- Implementar la funcionalidad para **serializar y deserializar** la lista de entidades a un archivo.
- Implementar la opción para **guardar y recuperar** los datos en formato **TXT**.
- Implementar una funcionalidad para **exportar un listado filtrado a un archivo de texto (.txt)**. Este archivo debe:
 - Incluir un encabezado descriptivo.
 - Ser legible por un ser humano (organizado para su impresión o lectura directa).

5. Diagrama UML

- Presentar un **diagrama de clases UML** que represente la estructura del proyecto, incluyendo clases, interfaces, relaciones de herencia e implementación, y métodos clave.

Instancia Oral: Defensa del Proyecto

Se deberá realizar la defensa del proyecto en una instancia oral, abordando los siguientes puntos:

1. Justificación del Diseño

- Explicar la elección del contexto y la estructura de las clases.
- Detallar cómo se implementaron búsquedas, modificaciones, persistencia e iteración.

2. Explicación Técnica

- Demostrar el uso correcto de **serialización, archivos CSV, archivos TXT**.
- Explicar el flujo de trabajo entre la lógica del sistema y la persistencia de datos.

3. Preguntas Técnicas

- Se evaluará el entendimiento de conceptos fundamentales como herencia, interfaces, polimorfismo y persistencia de datos.

Entrega

El proyecto debe entregarse completo, utilizando un repositorio de Github con:

- Código fuente debidamente comentado.
- Un archivo README con instrucciones de uso y descripción del proyecto.
- Un **diagrama UML** en formato PDF o imagen.
- Capturas de pantalla.

- Archivos generados (.dat, .csv, .json, .txt) con ejemplos prácticos de cada funcionalidad.
-

Criterios de Evaluación

- Precisión funcional: El sistema cumple los requisitos establecidos.
- Diseño: Uso adecuado de POO y persistencia.
- Documentación: Diagrama UML claro y detallado.
- Código limpio y organizado: Nombres descriptivos, comentarios y uso eficiente de los recursos de Java.
- Defensa: Capacidad para justificar decisiones técnicas y resolver preguntas.