



Universidad Nacional de San Martín

ALGORITMOS 1

LICENCIATURA EN CIENCIA DE DATOS

Trabajo Práctico Integrador

Versión	Fecha	Descripción
1.0	30/09/2025	T 2025 2do Cuatrimestre

INDICE

Introducción

- Objetivo
- Cronograma de entregas
- Formato de entrega
- Criterios de evaluación
- Entregables del trabajo
 - Documentación de Análisis
 - Documentación de Diseño
 - Código fuente
 - Presentación

Enunciado del problema

Introducción

El trabajo integrador de la materia reúne conceptos de los temas vistos durante la cursada. Se parte de un problema a resolver para realizar un análisis inicial, proponiendo un diseño parcial del mismo a partir de herramientas propuestas en la materia y finalmente implementando la solución en un lenguaje orientado a objetos. El desarrollo del trabajo se establece en etapas de entregas parciales para facilitar el seguimiento y corrección a lo largo del cuatrimestre.

El ciclo de vida del desarrollo de software se compone de varias fases, las cuales pueden variar según la metodología y modelo de procesos adoptado. En este trabajo se hace foco en la fase de implementación, apoyada previamente con documentación mínima de análisis y diseño. Se construirá un sistema desde sus inicios, produciendo así documentación que permita especificar suficientemente no sólo sus requerimientos sino también la solución propuesta a partir de diagramas, asimilando las decisiones y reflexiones que se desprenden del proceso.

Objetivo

El trabajo práctico integrador busca afianzar los conceptos vistos durante la materia a través del análisis, diseño e implementación en el lenguaje Java de un problema determinado. Cada una de las fases del ciclo de vida de desarrollo mencionadas requiere ciertos entregables que serán evaluados acorde a los criterios definidos en este documento.

Cronograma de entregas

Las fechas de entrega parcial se plantean para realizar el seguimiento activo del trabajo, por lo cual no son obligatorias, pero impactan en el aspecto de organización del grupo. La entrega final comprende a todos los entregables incluidos en este trabajo práctico integrador, el cual debe presentarse y defenderse en clase.

Fecha	Tipo de entrega	Entregables
9/9 y 12/9	Seguimiento I	Documentación de Análisis
7/10 y 10/10	Seguimiento II	Documentación de Diseño
11/11 y 14/11	Final	Todos

Formato de entrega

Se debe presentar el trabajo completo en formato digital. No existe una estructura predefinida requerida, es posible apoyarse en diferentes estándares o herramientas que vimos durante la materia. Para cada entregable, se plantean algunas sugerencias que pueden seguir.

Es requisito que se incluyan los contenidos mínimos solicitados para cada *entregable*.

Criterios de evaluación

Utilizaremos una evaluación basada en rúbricas donde se describen los distintos aspectos a considerar. La evaluación final del trabajo se descompone en dos partes:

- a) Evaluación grupal sobre la elaboración de los entregables.

Aspectos \ Nivel	1 - Regular	2 - Bueno	3 - Muy Bueno	4 - Excelente
Análisis del problema	No se interpreta correctamente el problema o no se presenta documentación vinculada al proceso de análisis.	Se presenta documentación suficiente de análisis acorde a lo solicitado en los entregables.	La documentación de análisis está completa y argumentada, relacionando temas vistos en clase.	El desarrollo de la documentación evidencia lectura de la bibliografía y emplea investigación adicional para mejorar el análisis.
Diseño de la solución basado en POO	No se presenta documentación del diseño acorde al análisis del problema, o está incompleta.	La documentación de diseño representa una solución básica en POO al problema dado.	La documentación de diseño está completa, incorpora conceptos de diseño, y favorece la extensibilidad y mantenibilidad.	Se hace uso de patrones de diseño, incluye argumentación y reflexión sobre principios, y contempla diversos atributos de calidad.
Implementación de la solución	El código implementado no satisface el problema completo, no compila, o arroja errores inesperados.	El código implementado resuelve el problema y no presenta errores en tiempo de compilación y ejecución.	El código implementado incorpora buenas prácticas vistas en clase y está alineado al diseño propuesto.	El desarrollo evidencia conocimiento profundo sobre el lenguaje, contempla el atributo de eficiencia, o incluye pruebas unitarias.
Organización y Presentación	Se entrega fuera de término o la documentación no tiene formato consistente.	Los entregables tienen un formato consistente y se prepara presentación para la defensa.	Se utilizan herramientas para facilitar la colaboración, versionado y organización de la documentación y código.	Se utilizan herramientas para gestionar el ciclo de vida del trabajo.

- b) Evaluación individual sobre la participación durante las prácticas y en la defensa del trabajo en la fecha de presentación.

Aspectos \ Nivel	1 - Regular	2 - Bueno	3 - Muy Bueno	4 - Excelente
Participación en el grupo	No se evidencia participación	Se plantean ideas o se escucha con	Se participa activamente en las	Contribuye de forma proactiva en

	dentro del grupo durante la elaboración del trabajo.	atención a los demás miembros del equipo.	discusiones y en la resolución del trabajo.	la resolución del trabajo y fomenta la colaboración en el grupo.
Defensa del trabajo	No asiste a la fecha de defensa del trabajo o no participa durante la presentación.	Expone con claridad su rol y aporte realizado al trabajo final.	Responde adecuadamente con seguridad las preguntas realizadas durante la defensa.	Relaciona conceptos vistos durante la materia, reflexionando y argumentando decisiones de diseño.

Entregables del trabajo

En esta sección se describe todo lo que se requiere entregar para evaluar el trabajo grupal, comprendiendo documentación solicitada, código de implementación y material referido a la presentación. En cada caso se describen contenidos mínimos y en ciertos casos algunos sugeridos.

Documentación de Análisis

Se solicita disponer de una especificación parcial, ya que no es el foco de evaluación en esta materia. Deberá estar **corregida y consensuada** entre el grupo y docente.

Contenidos mínimos requeridos:

- p1. Objetivo y Alcance
- p2. Descripción de alto nivel del sistema
- p3. Requerimientos funcionales más relevantes
- p4. Requerimientos no funcionales

Documentación de Diseño

El sistema debe ser diseñado bajo el **paradigma orientado a objetos**, contemplando los principios y conceptos vistos en la materia. Se deben tener presente los siguientes atributos de calidad:

- **Extensibilidad:** Es necesario que el modelo pueda extenderse fácilmente para incorporar nueva funcionalidad según sea necesario.
- **Mantenibilidad:** Hacer uso de los principios de POO para construir un producto que sea de fácil mantenimiento, por ejemplo evitando la duplicación de código y desacoplando los subsistemas cuando sea posible.
- **Gestión de errores:** El sistema debe capturar todos los errores posibles en tiempo de ejecución y clasificarlos según corresponda con una jerarquía de excepciones.

En primer lugar, este documento debe incluir una **vista general estática** a partir de un diagrama de clases que permitirá identificar todas las entidades del sistema con sus relaciones correspondientes, aunque no es necesario incorporar todos los atributos y métodos de las mismas.

En segundo lugar, se diseñarán **vistas adicionales** que deberán surgir de al menos un módulo, componente o proceso crítico para el sistema, decisión que tendrá que estar justificada. A excepción de una vista estática general, el análisis se suele realizar a bajo nivel para describir la estructura y comportamiento de lo considerado complejo de entender dentro del sistema.

Consideraciones importantes al momento de diseñar vistas:

- Las vistas deben ser consistentes entre ellas
- Una vista se compone de diagrama y descripción asociada
- La descripción de una vista puede incluir:
 - Detalles sobre los elementos y conexiones relevantes
 - Estilos arquitectónicos o patrones utilizados
 - Justificación de decisiones de diseño involucradas (si aplica)

Contenidos mínimos requeridos de la Documentación de Análisis: de al menos 2 Casos de Uso Importantes y de alta interacción entre clases.

- Vista estática del sistema
 - p5.Diagrama Conceptual
 - p6. Diagrama de clases del Modelo
 - p7.Diagrama de la Vista y Controlador
- p8.Diagrama de Secuencia de al menos 2 Casos de Uso Importantes y de alta interacción entre clases.
- P9. Diagrama de Navegación de GUI

Contenidos opcionales:

- Diagrama de clases de un módulo o subsistema relevante
- Diagrama de secuencia y su diagrama de clases asociado
- Diagrama de estados y su diagrama de clases asociado

Código fuente

Entrega de la implementación en código fuente en lenguaje Java, compatible con versión 8 o superior. Es deseable la entrega de la documentación de uso asociada, es posible utilizar herramientas de generación automática como Javadoc. También es muy valorada la incorporación de pruebas unitarias, pudiendo utilizar herramientas como JUnit.

Contenidos mínimos requeridos:

- El código compila sin errores
- La ejecución no arroja excepciones inesperadas

Presentación

Si bien es en sí no un entregable, la defensa del trabajo se realizará durante la presentación del mismo, en la cual pueden generar contenido multimedia de apoyo. En esta instancia se debe demostrar el funcionamiento del producto construido con algún caso de uso.

Pueden incluirse dentro de la entrega todo contenido generado para la presentación, como así también documentación o artefactos relacionados para mostrar el funcionamiento.

Contenidos recomendados:

- A. Presentación de integrantes y responsabilidades dentro del trabajo
- B. Breve descripción del análisis del problema
- C. Detalle de la solución diseñada: de al menos 1 caso de uso (de los 9 puntos definidos)
- D. Problemas encontrados durante el desarrollo
- E. Investigación asociada para resolver los problemas
- F. Posibles mejoras pendientes
- G. Estrategias para incorporar nueva funcionalidad
- H. Demostración de la funcionalidad implementada con un caso de uso
- I. Conclusiones

Enunciado del problema

Se desea construir un videojuego en Processing, utilizando lenguaje Java, inspirado en el clásico videojuego de la década de 1980: "1942". El proyecto debe implementar los principios de la Programación Orientada a Objetos (POO) y contener tanto mecánicas jugables como un módulo de análisis de datos.

Requerimientos principales

1. Juego principal

- Debe incluir al menos un jugador controlable y enemigos u obstáculos.
- Implementar niveles o fases crecientes en dificultad.
- Manejo de colisiones, vidas, puntaje y condiciones de victoria/derrota.
- Uso de herencia, interfaces y polimorfismo en el diseño de clases.

2. Módulo de Ciencia de Datos

- Recolección de estadísticas de la partida: tiempo de juego, puntaje, precisión en disparos, enemigos derrotados, vidas perdidas, etc.
- Almacenamiento de datos en un archivo (CSV o JSON).
- Visualización dentro del juego o en un módulo separado (tablas, gráficos simples en Processing).
- Análisis básico de métricas: promedio, máximo, mínimo, porcentaje de victorias.

3. POO y calidad de código

- Diagrama de clases del sistema.
- Principios de diseño: extensibilidad, mantenibilidad, manejo de excepciones.
- Implementación en Processing (Java), con código que compile y ejecute correctamente.

4. Extensiones

- Ajuste dinámico de dificultad según el rendimiento del jugador.
- Comparación de desempeño entre partidas (ranking histórico).
- Incorporar métricas avanzadas (varianza, desvío estándar, tendencias).