



Universidad Nacional de San Martín

ALGORITMOS 1

LICENCIATURA EN CIENCIA DE DATOS

Trabajo Práctico Integrador

Versión	Fecha	Descripción
1.0	27/08/2024	T 2024 2 Cuatrimestre

INDICE

Introducción.....	3
Objetivo.....	3
Cronograma de entregas.....	3
Formato de entrega.....	3
Criterios de evaluación.....	4
Entregables del trabajo.....	6
Documentación de Análisis.....	6
Documentación de Diseño.....	6
Código fuente.....	7
Presentación.....	7
Enunciado del problema.....	8
Funcionalidad principal de la librería.....	8
Información básica.....	8
Acceso indexado.....	8
Formatos de carga/descarga.....	9
Visualización.....	9
Generación y modificación.....	9
Selección.....	9
Filtrado.....	9
Copia independiente.....	10
Concatenación.....	10
Ordenamiento.....	10
Imputación.....	10
Muestreo.....	10
Funcionalidad opcional de la librería.....	11
Agregación.....	11

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
17/9	Seguimiento	Documentación de Análisis
8/10	Seguimiento	Documentación de Diseño
12/11 y 19/11	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 dentro del grupo durante la elaboración del trabajo.	Se plantean ideas o se escucha con atención a los demás miembros del equipo.	Se participa activamente en las discusiones y en la resolución del trabajo.	Contribuye de forma proactiva en 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:

- Objetivo y Alcance
- Descripción de alto nivel del sistema
- Requerimientos funcionales más relevantes
- 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:

- Vista estática del sistema (diagrama de clases general)

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:

- Presentación de integrantes y responsabilidades dentro del trabajo
- Breve descripción del análisis del problema
- Detalle de la solución diseñada
- Problemas encontrados durante el desarrollo
- Investigación asociada para resolver los problemas
- Posibles mejoras pendientes
- Estrategias para incorporar nueva funcionalidad
- Demostración de la funcionalidad implementada con un caso de uso
- Conclusiones

Enunciado del problema

Se desea construir una librería que permita manipular y analizar datos en forma tabular (2 dimensiones) para el lenguaje Java. Deberá ofrecer estructuras de datos y operaciones que soporten la funcionalidad solicitada, como así también contemple posibles extensiones futuras, minimizando el impacto ante alguna modificación.

En principio no es necesario hacer foco en la eficiencia de las operaciones, pero sería deseable disponer también de algún mecanismo que nos permita cuantificar, al menos en tiempo, el costo de su ejecución.

Es posible apoyarse en estructuras existentes de forma nativa en el lenguaje Java para construir las estructuras de datos, pero no se debe depender de librerías externas. En caso de necesitar consumir alguna librería externa, se debe consultar previamente si es viable su aplicación.

Funcionalidad principal de la librería

Información básica

Se debe poder conocer la siguiente información sobre una instancia dada:

- Cantidad de filas
- Cantidad de columnas
- Etiquetas de filas
- Etiquetas de columnas
- Tipos de datos de columnas

De esto se desprende que cada columna tiene un tipo de dato asociado, el cual debe validarse al momento de generación o modificación. Por el momento, los tipos de datos soportados para una columna son:

- Numérico (entero, real, etc)
- Booleano
- Cadena

Nota: Debe existir un valor especial que hace referencia a valor faltante (*NA*), el cual puede asignarse a cualquier celda sin importar el tipo de dato de la columna.

Acceso indexado

Proveer acceso indexado a nivel de fila y columna. Esto permitirá que podamos acceder a:

- una fila completa si se selecciona la etiqueta de la fila
- una columna completa si se selecciona la etiqueta de la columna
- una celda si se especifican etiquetas de fila y columna simultáneamente

Una etiqueta (*label*) puede ser en formato numérico entero o cadena. Si no se especifican etiquetas en una estructura, se inicializan como secuencia de números enteros iniciada en 0.

Formatos de carga/descarga

Soportar la lectura y escritura entre estructuras en memoria y el formato CSV en disco. Se debe poder establecer el carácter delimitador de columnas y si se usa encabezado (etiquetas de las columnas).

Visualización

Presentar de forma simple y sencilla en formato texto la información en forma de tabla. Se pueden definir tamaños máximos configurables para cortar la salida en caso de estructuras grandes. Por ejemplo, mostrar un máximo de columnas o filas, y mostrar un máximo de caracteres por cada celda.

Generación y modificación

Una estructura tabular puede generarse de la siguiente forma:

- a) Desde la carga de un formato compatible en disco (CSV)
- b) A través de copia profunda de otra estructura del mismo tipo
- c) Desde una estructura de dos dimensiones nativa de Java
- d) Desde una secuencia lineal nativa de Java

Una estructura tabular puede modificarse de la siguiente forma:

- a) Accediendo directo a una celda y asignando un nuevo valor
- b) Insertando una columna nueva a partir de otra columna (con misma cantidad de elementos que filas)
- c) Insertando una columna nueva a partir de una secuencia lineal nativa de Java (con misma cantidad de elementos que filas)
- d) Eliminando una columna
- e) Eliminando una fila

Selección

Permitir la selección parcial de la estructura tabular a través de una lista de etiquetas de cada índice. Esta operación no genera una nueva estructura, sino una vista reducida de la original (*slicing*).

Por ejemplo, se deberían poder elegir ciertas filas y columnas para establecer una selección o vista de la estructura original.

Casos especiales:

- `head(x)`: Devolver las primeras x filas
- `tail(x)`: Devolver las últimas x filas

Filtrado

Permitir la selección parcial de la estructura tabular a través de un filtro aplicado a valores de las celdas (*query*). Este filtro se puede componer de uno o más comparadores sobre cierta columna (`<`, `>`, `=`) que se combinan con operadores lógicos (`and`, `or`, `not`). Así es posible generar filtros del estilo “`columna1 > 3 and columna2 = Verdadero`”, filtrando así aquellas filas donde las celdas cumplen aquella condición.

Esta operación devuelve una selección o vista de la estructura original.

Copia independiente

Permitir la copia profunda de los elementos de la estructura para generar una nueva con mismos valores, pero independiente de la estructura original en memoria.

Concatenación

Permitir generar una nueva estructura tabular a partir de la concatenación de dos estructuras existentes, creando así una nueva combinando las filas de la primera y luego las filas de la segunda. Esta operación es válida si las columnas de ambas estructuras coinciden, tanto en cantidad de columnas como también orden, tipo de datos y etiquetas asociadas. Las etiquetas de las filas son generadas automáticamente.

Esta operación debe utilizar la copia de las estructuras originales.

Ordenamiento

Permitir ordenar las filas de la estructura según un criterio (ascendente o descendente) sobre una o más columnas. Si se opta por ordenar por más de una columna, se toma el orden ingresado de las columnas como precedencia para ordenar las filas.

Imputación

Incorporar la posibilidad de modificar (*rellenar*) las celdas con valores faltantes (celdas con *NA*) con cierto valor literal indicado.

Muestreo

Ofrecer una selección aleatoria de filas según un porcentaje del total de la estructura.

Funcionalidad opcional de la librería

A continuación se presentan características deseables de la librería que no son obligatorias en esta entrega pero sería interesante disponerlas.

Agregación

Incorporar la posibilidad de dividir las filas en diferentes grupos (agrupamiento o *groupby*) según una o más columnas. Luego aplicar a estos grupos una operación de sumarización estadística y devolver una nueva estructura donde:

- la etiqueta de las filas sea el nombre del grupo (si se dividió por más de una columna, se pueden separar por comas)
- las columnas sean sólo aquellas que no son parte del grupo y con tipo numérico
- las celdas tengan el valor de la operación de sumarización aplicado al grupo

Por el momento, las operaciones disponibles de sumarización son:

- suma
- máximo
- mínimo
- cuenta
- media
- varianza
- desvío estándar