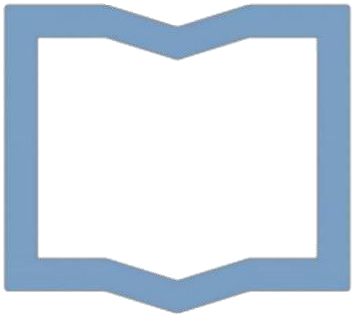

Universidad Nacional de San Martín

Escuela de Ciencia y Tecnología

Algoritmos I

(1er cuatrimestre 2025)

Profesores: Miguel Carboni, Florencia Rossi, Fabiana Taboada

Documentación de Análisis

Grupo 3

Integrantes: Pontiroli, Nicolás - Scarpin, Julián - Semec, Santino - Sosa, Agustina.

1. Objetivo:

Desarrollar una librería en Java para manipular y analizar datos tabulares (2D) con funcionalidades básicas de acceso, modificación, filtrado, ordenamiento y visualización. También se le dará importancia a la capacidad de la librería de poder extenderse para agregar nuevas funcionalidad y con ello tener la capacidad de tratar de minimizar el impacto de las mismas.

2. Alcance:

La librería está diseñada para ofrecer capacidades básicas y avanzadas de manipulación y análisis de datos en formato tabular con Java. Está diseñada para funcionar por sí misma, sin necesitar otros programas adicionales, y hace que trabajar con tablas de datos sea rápido y sencillo.

En cuanto a los tipos de datos soportados, la librería maneja valores numéricos, booleanos y cadenas de texto. Un aspecto importante es el soporte para valores faltantes, representados mediante el valor especial NA, que podrá asignarse a cualquier celda independientemente del tipo de dato de su columna correspondiente.

Para la carga y persistencia de datos, se implementará funcionalidad completa para trabajar con archivos en formato CSV. Esto incluye la capacidad de leer datos desde archivos con o sin encabezados, así como la posibilidad de exportar las estructuras de datos tabulares a este formato.

Los datos se acceden mediante etiquetas personalizables (numéricas o textuales) para filas/columnas. Sin etiquetas, se autoasignan números secuenciales (desde 0). Ofrece métodos para extraer subconjuntos como pueden ser head/tail para filas iniciales/finales, y filtros con condiciones personalizadas.

La generación de nuevas estructuras tabulares podrá realizarse directamente desde archivos CSV, mediante copia profunda de estructuras existentes, o a partir de estructuras de datos nativas de Java como arrays bidimensionales o listas de listas. Esto proporcionará flexibilidad en la creación de los conjuntos de datos iniciales.

Entre las funcionalidades avanzadas que incluirá la librería se encuentran operaciones de filtrado con operadores lógicos, capacidad de concatenar múltiples estructuras tabulares compatibles, ordenamiento de datos por una o más columnas (en orden ascendente o descendente), y herramientas para el manejo de valores faltantes mediante imputación con valores predeterminados.

Para evaluar el rendimiento, la librería incorporará mecanismos que permitan medir el tiempo de ejecución de las operaciones más críticas, como el filtrado y ordenamiento de datos. Esto facilitará la identificación de posibles cuellos de botella en el procesamiento.

Los tipos de datos complejos como fechas u objetos personalizados no estarán soportados directamente, requiriendo conversión previa a los tipos básicos. Tampoco se implementará validación de esquemas externos más allá del formato CSV básico.

En términos de capacidad, la librería está optimizada para manejar conjuntos de datos que puedan residir completamente en memoria, con un rendimiento adecuado para tablas de hasta aproximadamente un millón de filas en hardware estándar. Todos los archivos CSV procesados deberán estar codificados en UTF-8, sin soporte para otras codificaciones de caracteres.

3. Descripción de Alto Nivel del Sistema

Módulo	Responsabilidad principal	Ejemplo de clases	Ventajas	Desventajas
Principal	Modelo tabular en memoria y operaciones (filtrar, ordenar, etc.).	DataTable, Column, Row	- Lógica de negocio aislada - Fácil de testear sin I/O	- Riesgo de crecer demasiado
I/O	Leer y escribir formatos externos (por ahora CSV).	CsvReader, CsvWriter	- Si se quiere agregar formatos nuevos no toca Principal - Responsabilidad encapsulada	- Pequeño overhead de conversión.
Utils	Soporte transversal: temporizador, validaciones, constante NA.	OperationTimer, NA	- Evita duplicar helpers. - Mantiene Principal Limpio.	- Si crece sin control, se vuelve un “cajón de sastre”.

Esta división responde a los atributos de extensibilidad y mantenibilidad solicitados en la rúbrica. Cada módulo tiene una sola razón de cambio y puede evolucionar de forma independiente.

El sistema utiliza una estructura matricial optimizada para almacenamiento y acceso eficiente a datos tabulares. Implementa un esquema completo de etiquetado que permite identificar filas y columnas mediante nombres personalizados o índices numéricos. Cada columna mantiene un tipo de dato definido (numérico, texto, booleano o NA), con validación estricta durante todas las operaciones para garantizar la integridad de los datos.

Para la manipulación básica de datos, el sistema ofrece operaciones para crear, leer, actualizar o eliminar permitiendo el acceso preciso a celdas individuales mediante coordenadas de fila/columna, así como la inserción y eliminación controlada de filas o columnas completas. Estas operaciones básicas están optimizadas para mantener un equilibrio entre rendimiento y seguridad en la manipulación de datos.

En el ámbito de las operaciones avanzadas, la librería incorpora un potente motor de filtrado que permite operadores lógicos (AND, OR) y de comparación (>, <, =, !=). Complementando esta capacidad, implementa algoritmos eficientes para la concatenación de tablas compatibles y métodos de ordenamiento multicriterio, tanto ascendente como descendente, diseñados para manejar grandes volúmenes de datos.

El subsistema de persistencia proporciona capacidades completas para la importación y exportación de datos en formato CSV, con flexibilidad para configurar parámetros como caracteres delimitadores y manejo de encabezados. Este módulo garantiza la correcta serialización y deserialización de los datos, manteniendo la coherencia de tipos y la estructura tabular.

Para la visualización de resultados, la librería genera representaciones tabulares claras en formato texto, con mecanismos configurables para limitar la salida cuando se trabaja con conjuntos de datos extensos. Este sistema de visualización incluye ajustes automáticos de formato según el tipo de dato y herramientas para paginación controlada de resultados.

4. Requerimientos Funcionales (RFs):

Macro-requerimiento 1: Gestión de Estructuras Tabulares			
ID	Requerimiento	Descripción	Referencia de módulo
RF-1.1	Creación desde múltiples fuentes	Generar estructuras desde CSV, copias profundas, o arrays nativos de Java. Permitir seleccionar las columnas que formen parte del df.	Principal + I/O
RF-1.2	Modificación de celdas	Asignar valores a celdas específicas, validando el tipo de dato de la columna. Asignar nuevos nombres a columnas. Reemplazar datos.	Principal
RF-1.3	Inserción/eliminación	Añadir o quitar filas/columnas con etiquetas personalizadas.	Principal
RF-1.4	Concatenación de tablas compatibles	Unir dos tablas compatibles (mismas columnas, tipo y orden) para formar una nueva con todas las filas de ambas.	Principal

Macro-requerimiento 2: Operaciones de Consulta			
ID	Requerimiento	Descripción	Referencia de módulo
RF-2.1	Filtrado lógico	Filtrar filas basado en condiciones lógicas (<,>,=, |, &)	Principal
RF-2.2	Ordenamiento	Ordenar por una o más columnas (ascendente/descendente).	Principal
RF-2.3	Selección parcial	Obtener vistas (head, tail, slicing por etiquetas).	Principal
RF-2.4	Manejo de valores atípicos	Soluciones para asignar valores literales (mean, median, etc) o eliminarlos	Utils
RF-2.5	Muestreo aleatorio de filas	Selección aleatoria de muestras	Utils
RF-2.6	Conteo de valores	Contar datos, en toda la tabla o en columnas. También valores faltantes	Principal

Macro-requerimiento 3: Persistencia y Visualización			
ID	Requerimiento	Descripción	Referencia de módulo
RF-3.1	Carga/descarga CSV	Leer y escribir archivos CSV con delimitadores personalizables.	I/O
RF-3.2	Visualización en consola	Mostrar datos en formato tabla, limitando filas/columnas visibles.	Principal

5. Requerimientos No Funcionales (RNFs):

ID	Propiedad	Métrica Clara	Referencia de módulo
RNF-01	Extensibilidad	Nuevos tipos de datos (ej: fechas) podrán añadirse sin modificar el núcleo.	I/O
RNF-02	Rendimiento	Filtrado de filas debe completarse en < 500 ms.	Utils
RNF-03	Mensajes de error descriptivos	Creación de Excepciones personalizadas con información de lo que falló	Utils
RNF-04	Portabilidad	Compatibilidad con Java 8+ (sin dependencias externas).	Utils
RNF-05	Documentación	Documentar el 100% de los métodos públicos con ejemplos de uso.	Doc

6. Trabajo a futuro:

Funcionalidad opcional – Agregación (Group-By):

Para la tercera etapa del proyecto se proyecta incorporar un módulo de agregación que permita obtener resúmenes estadísticos por grupos, funcionalidad habitual en bibliotecas de análisis de datos.

Objetivo:

Implementar un procedimiento groupBy() que:

1. Agrupe las filas según el valor de una o más columnas seleccionadas.

2. Calcule sobre cada grupo funciones de resumen aplicables a las columnas numéricas restantes.

3. Devuelva una nueva estructura tabular cuya etiqueta de fila identifique al grupo y cuyos valores correspondan a las funciones solicitadas.