

Algoritmos 1

Presentación Trabajo Práctico Integrador

Grupo n° 5

Integrantes:

Danny Alejandra Baron Romero

Alejandro Echeverri

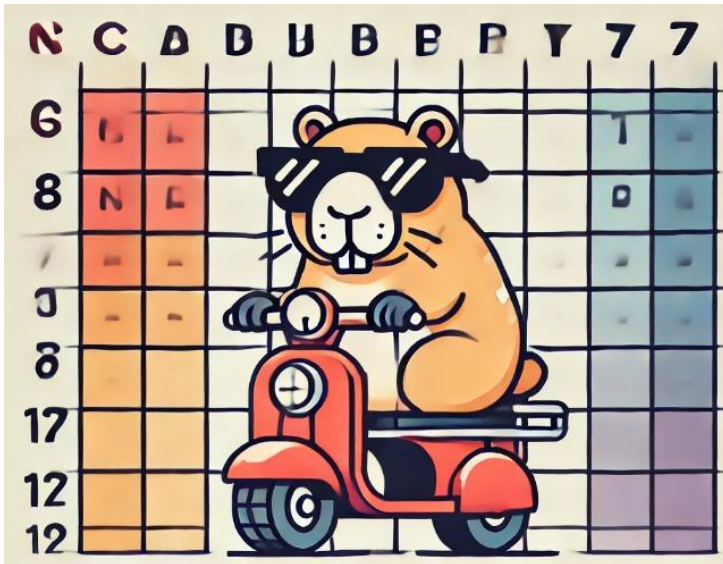
Agustin Rebechi

Cristian Vidal Sastre

Rocio Belen Rivera

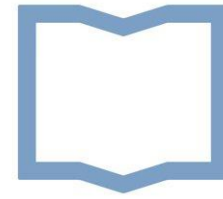
19 de Noviembre 2024

Algoritmo 1. 2024



Introducción

Nuestro equipo:



Universidad
Nacional
de San Martín



Agustín Rebechi



Alejandro
Echeverri



Danny Alejandra
Baron Romero



Cristian Vidal
Sastre



Rocío Belén
Rivera

Proyecto

Objetivo:

El objetivo principal de este proyecto es diseñar e implementar una librería en Java que permita la manipulación y el análisis de datos en forma tabular (en dos dimensiones), proporcionando estructuras de datos y operaciones que permitan gestionar dicha información de manera flexible y extensible. Se busca construir una solución robusta que no dependa de librerías externas, enfocada en la facilidad de uso, la posibilidad de futuras extensiones, y la cuantificación del costo temporal de las operaciones sin centrarse en la optimización prematura de la eficiencia.

Proyecto

Alcance:

- **Estructuras de datos:** Desarrollo de clases y métodos que permitan la representación de datos en tablas bidimensionales, soportando diferentes tipos de valores en las celdas (por ejemplo, numéricos y cadenas).
- **Operaciones sobre tablas:** Implementación de funciones para manipular, agregar, eliminar y consultar datos dentro de las tablas, así como la posibilidad de realizar análisis básicos como filtrado, ordenamiento, y agregación de datos.
- **Extensibilidad:** La librería debe estar diseñada de manera que futuras extensiones, como nuevas operaciones o estructuras adicionales, puedan implementarse sin modificar significativamente el código existente.
- **Mecanismo de medición:** Implementación de un mecanismo que permita medir el tiempo de ejecución de las operaciones realizadas sobre las tablas, ofreciendo una base para evaluar el costo computacional de las mismas.

Proyecto

Motivación:

La motivación de este proyecto radica en la necesidad de contar con una herramienta versátil y eficiente que facilite el manejo de datos tabulares en Java, algo común en diversas aplicaciones, como análisis de datos, procesamiento de información y automatización de procesos. Dado que muchas bibliotecas externas existentes pueden no adaptarse completamente a las necesidades específicas de un proyecto o conllevan restricciones en su uso, este trabajo busca desarrollar una solución nativa, extensible y ajustada a las necesidades del desarrollador. Asimismo, nos permitió aplicar los conceptos de programación orientada a objetos, diseño de software y análisis de requerimientos adquiridos durante la cursada, promoviendo una comprensión profunda del ciclo de vida de desarrollo de software a través de la implementación de un proyecto integrador.



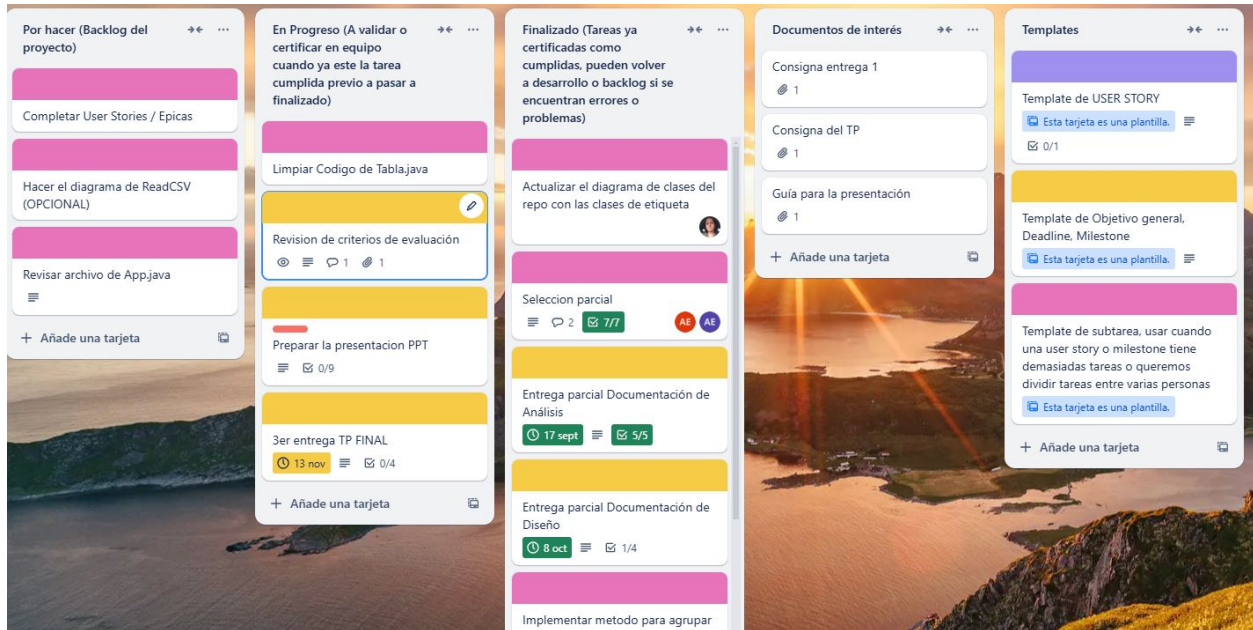
Índice



Universidad
Nacional
de San Martín

- 01.Introducción (RO)
- 02.Metodología (CRISTIAN)
- 03.Colaboración del equipo (ALE)
- 04.Descripción de la solución
- 05.Desafíos encontrados (AGUS)
- 06.Demostración en tiempo real
- 07.Análisis del proyecto
- 08.Conclusiones

Metodología



¿Por qué elegimos Kanban?

- **Visibilidad y Control:** Usamos tableros Kanban para visualizar el estado de cada tarea, desde la fase de desarrollo hasta la entrega.
- **Flujo continuo:** Facilitamos la entrega constante de valor a nuestros usuarios mediante la gestión del flujo de trabajo sin interrupciones ni sobrecargas.
- **Mejora continua:** Kanban nos permite identificar cuellos de botella y mejorar nuestro proceso de forma continu

Colaboración del Equipo

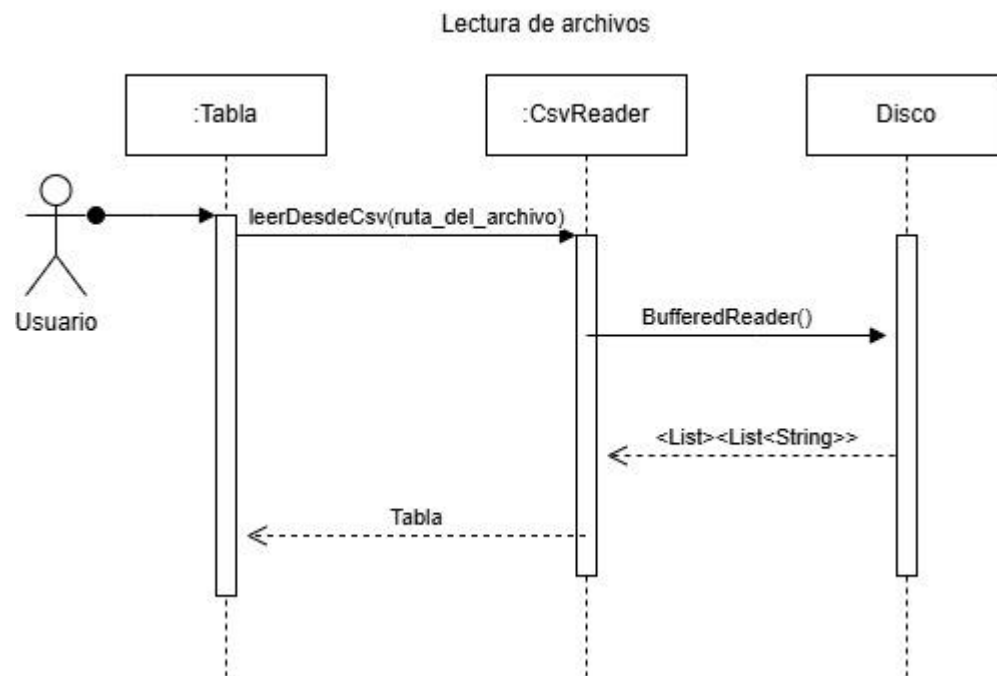
Comunicación:

- Sincrónica: A través de las clases regulares y videollamadas semanales, donde repasamos el trabajo realizado, coordinamos los próximos pasos y proponemos soluciones o mejoras.
- Asincrónica: Principalmente mediante chats de WhatsApp, utilizados para resolver dudas, compartir avances y proponer soluciones de forma ágil.

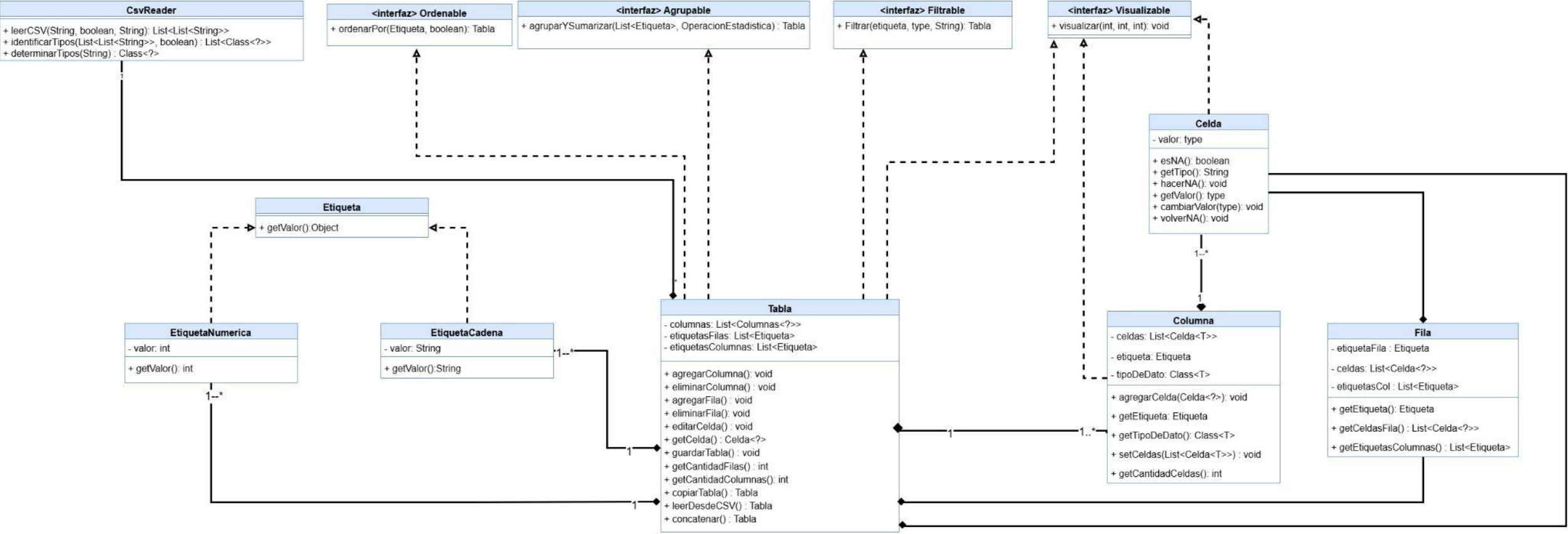
Estrategia repositorio de código ("feature branching" y "pull requests" informales) :

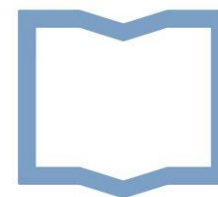
- Desarrollo continuo: Trabajan de forma iterativa en el código, desarrollando nuevas funcionalidades.
- Pruebas y revisión en equipo: Realizan pruebas internas y validan las funcionalidades con el equipo.
- Aprobación en equipo: Una vez que todos están de acuerdo en que la funcionalidad es correcta y estable, se realiza el push al repositorio central.

Descripción de la Solución



Descripción de la Solución



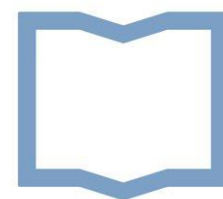


Descripción de la Solución

Diseño funcional para métodos de filtrado, agrupamiento, orden y sumarización. Estos métodos siempre devuelven un objeto tipo Tabla por lo que pueden ser encadenados.

```
Tabla pokemonPesadoAcero = tabla
    .filtrar(new EtiquetaCadena(valor:"Weight"),valorReferencia:100,operador:">")
    .filtrar(new EtiquetaCadena(valor:"Type"), valorReferencia:"Steel",operador:"=");
pokemonPesadoAcero.ordenarPor(new EtiquetaCadena(valor:"Weight"), ascendente:false);
System.out.println(x:"Los pokemons de tipo acero mas pesados con peso mayor a 100 kilos son");
pokemonPesadoAcero.visualizar(maxFilas:100, maxColumnas:20,maxLargoCadena:10);

Tabla pokemonPesadoPlanta = tabla
    .filtrar(new EtiquetaCadena(valor:"Weight"),valorReferencia:100,operador:">")
    .filtrar(new EtiquetaCadena(valor:"Type"), valorReferencia:"Rock",operador:"=")
    .ordenarPor(new EtiquetaCadena(valor:"Weight"), ascendente:false);
System.out.println(x:"Los pokemons de tipo Roca mas pesados con peso mayor a 100 kilos son");
pokemonPesadoPlanta.visualizar(maxFilas:100, maxColumnas:20,maxLargoCadena:10);
```



Desafíos encontrados

- Manejo de branches en git, sobre todo cuando trabajamos varias personas en un mismo archivo.
- Reingeniería de los diagramas de clases cuando en la etapa de diseño no pudimos prever la necesidad de componentes más avanzados o diversificados.
- Manejo de tipos de datos dinámicos en las celdas, implementación de Generics

Demostración en tiempo real

Pueden usar:

Clase pregenerada con métodos listos para probar toda la funcionalidad: App.java y pruebas específicas para ciertas clases o funcionalidades

Especificación de Clases, Interfaces y especificaciones técnicas en nuestra web generada con JavaDocs:

Documentación y Diagramas de clases integrados en nuestro repositorio.

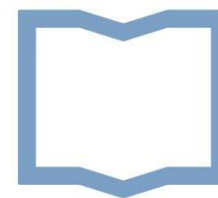
Análisis del Proyecto

Lecciones aprendidas (aquellas cosas que no salieron como lo esperábamos)

- Como todo proyecto de software el tiempo siempre es un factor limitante y hay que priorizar las funcionalidades más importantes
- Implementación de otros tipos de datos más performantes para las celdas (HashMap por ejemplo).

Casos de éxito (aquellas cosas que salieron muy bien y valen la pena mencionar.)

- Formateo personalizado al mostrar las tablas. Motivo.
- Rendimiento en lectura de archivos (pusimos como meta leer un csv de 10 mil filas en menos de 10 segundos y nuestro programa tarda menos de 1 segundo)



Conclusiones

- No solo se define una solución Técnica, sino que se definen metodologías que ayuden a mejorar la comunicación del equipo involucrado en el desarrollo del software.
- Se requiere una validación continua de los requerimientos con los StakeHolders de manera para garantizar que el proyecto cumpla los objetivos esperados.