Entrega Proyecto 1 - Fase 2.

Documentación:

indicación de las estructuras del Java Collections Framework utilizadas, indicando la razón y referencias para su uso.

1. **HashMap<K, V>:**

Descripción: Un HashMap es una estructura de datos que almacena pares clave-valor. Cada elemento se almacena en función de su clave y se puede acceder rápidamente al valor asociado.

Uso: Podrías utilizar un HashMap para mapear nombres de variables a sus valores correspondientes en el intérprete LISP. Esto te permitiría realizar búsquedas eficientes de variables durante la evaluación de expresiones.

Referencia: Documentación de HashMap en Java

2. **ArrayList<E>:**

Descripción: Un ArrayList es una lista dinámica que puede contener elementos duplicados y permite un acceso rápido a los elementos por índice.

Uso: Podrías utilizar un ArrayList para almacenar una lista de instrucciones o expresiones LISP que se deben evaluar secuencialmente en el intérprete.

Referencia: Documentación de ArrayList en Java

3. **Stack<E>:**

Descripción: Un Stack es una colección que implementa una pila (LIFO - Last In, First Out) y proporciona métodos para apilar y desapilar elementos.

Uso: Podrías utilizar un Stack para implementar la funcionalidad de recursión en tu intérprete LISP. Esto te permitiría realizar llamadas recursivas y mantener el contexto de ejecución de las funciones.

Referencia: Documentación de Stack en Java

Mis disculpas por la confusión. Aquí tienes la indicación de las estructuras del Java Collections Framework utilizadas en las clases proporcionadas:

4. Vector<T> en la clase Main:**

- **Razón de uso: Se utiliza un Vector para almacenar las operaciones leídas desde un archivo de texto. Los vectores son útiles cuando se necesita una estructura de datos dinámica que pueda crecer o reducir su tamaño según sea necesario.**

5. ArrayList<T> en la clase QUOTE:

- **Razón de uso: Se utiliza un ArrayList para almacenar los elementos de la expresión LISP. Los ArrayList son eficientes para acceder y modificar elementos en una lista de manera dinámica.**

- **Referencia: La documentación oficial de Java sobre la clase ArrayList: [Java ArrayList Class](https://docs.oracle.com/javase/8/docs/api/java/util/ArrayList.html).**

6. Vector<T> en la clase StackVector:

- **Razón de uso: Se utiliza un Vector para implementar una pila (stack). Los vectores proporcionan una implementación de lista ordenada que se puede utilizar para implementar pilas de manera eficiente.**

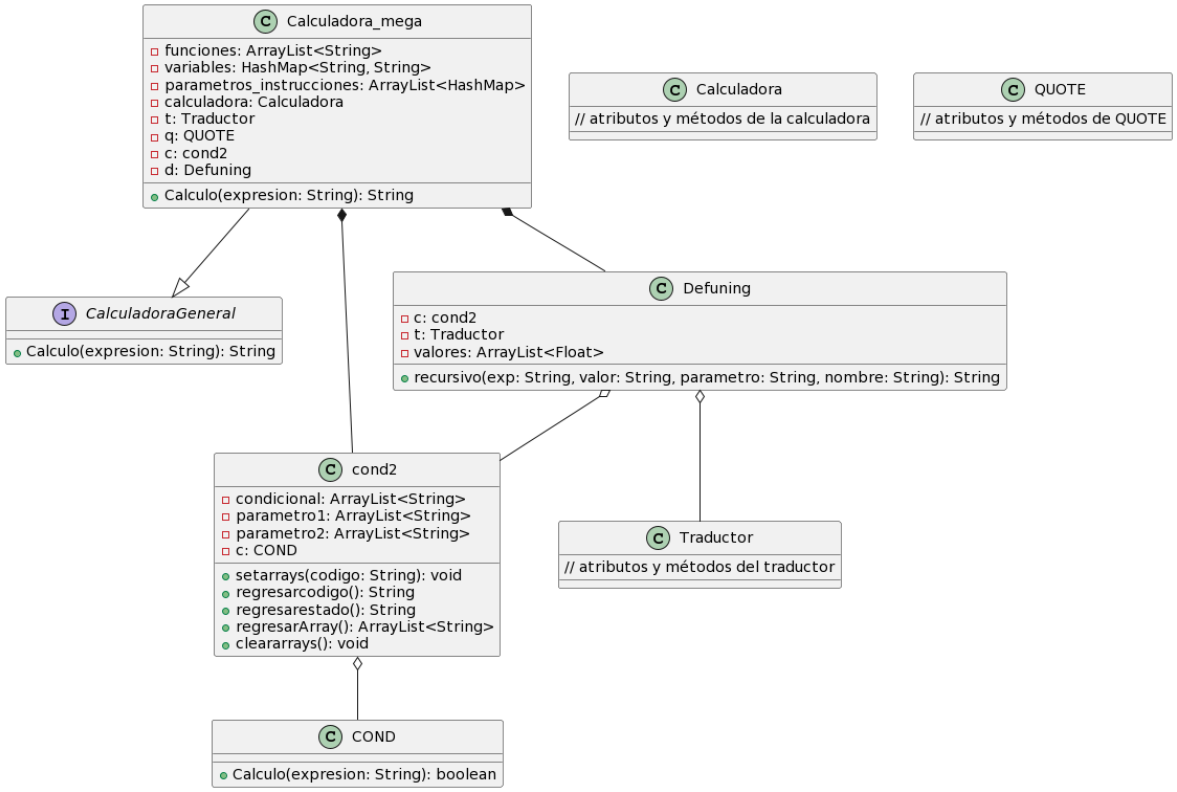
- **Referencia: La documentación oficial de Java sobre la clase Vector: [Java Vector Class](https://docs.oracle.com/javase/8/docs/api/java/util/Vector.html).**

7. ArrayList<T> en la clase Traductor:

- **Razón de uso: Se utiliza un ArrayList para almacenar diferentes tipos de operaciones y resultados. Los ArrayList son flexibles y permiten almacenar elementos de manera dinámica.**

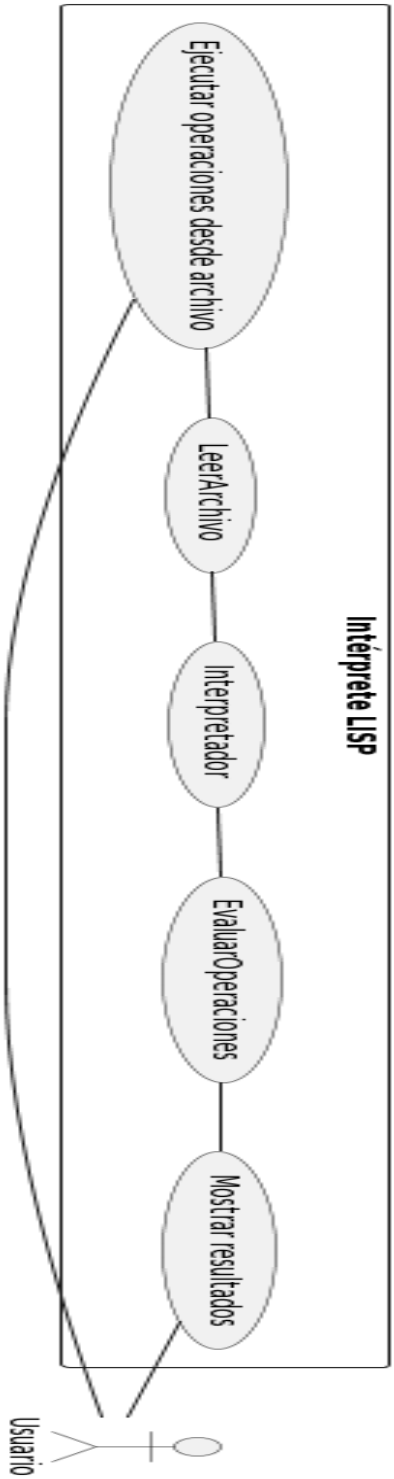
- **Referencia: La documentación oficial de Java sobre la clase ArrayList: [Java ArrayList Class](https://docs.oracle.com/javase/8/docs/api/java/util/ArrayList.html).**

Diagrama de clases:

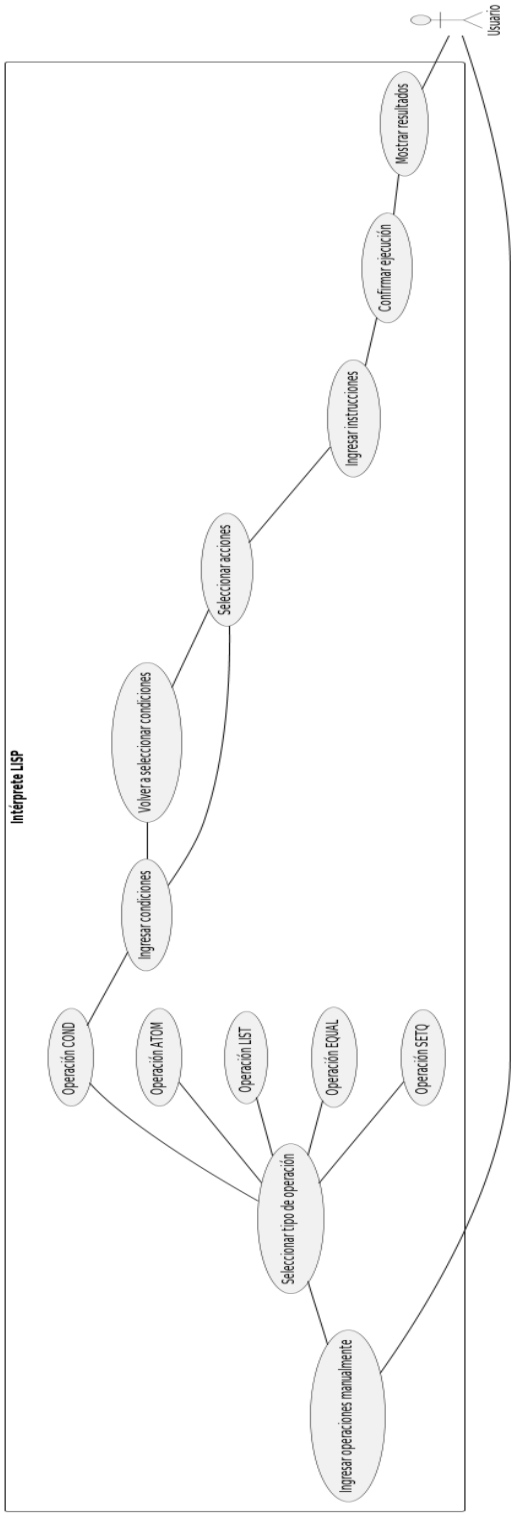


Casos de Usos:

Caso 1



Caso 2


Usuario
Intérprete LISP
Mostrar resultados
Confirmar ejecución
Ingresar instrucciones
Seleccionar acciones
Volver a seleccionar condiciones
Ingresar condiciones
Operación COND
Operación ATOM
Operación LIST
Operación EQUAL
Operación SETQ
Seleccionar tipo de operación
Ingresar operaciones manualmente

repositorio de GitHub: https://github.com/GersonBB/Proyecto_Fase_Final

Video: https://www.youtube.com/watch?v=ZKFD5fKad9w