Para la memoria, vamos a crear un objeto Memoria que tenga como atributos arreglos, uno para cada combinación de tipo y “categoría” (tipo es el tipo de dato, categoría es si es variable, constante o temporal). Es decir, la clase Memoria tendría como atributos:

int[] varInt;

int[] cteInt;

int[] tempInt;

float[] varFloat;

float[] cteFloat;

float[] tempFloat;

…

El constructor de dicha clase tomará N parámetros enteros (siendo N la cantidad de atributos), con lo cual se inicializará el tamaño de cada uno de los arreglos. Este tamaño debe ser proporcionado como parte del código intermedio, además de los cuádruplos. Es decir, para cada función, debemos listar en el código intermedio cuántas variables enteras hubo, cuántas constantes enteras, temporales enteras, etc.

En la máquina virtual, se tendrá una referencia a un objeto de la clase Memoria llamada memGlobal, que será, como su memoria lo indica, la memoria global. Además habrá un stack de memorias locales, donde la que esté en el tope de la pila es la memoria local actual.

Habrá una función getValue que tome como parámetro un entero que represente una dirección, y obtenga el valor de la casilla del arreglo correspondiente según su valor, que está preestablecido.