package parcial;

import tp03.ejercicio4.ListaGenerica;

public class ProcesadorGematria {

public static int contadorGematria(ArbolGeneral<Integer> ag, int valor) {

int resta = valor - ag.getDatoRaiz();

if (ag.esHoja() && resta == 0)

return 1;

else { VERSION 1

int cont = 0;

ListaGenerica<ArbolGeneral<Integer>> lista = ag.getHijos();

lista.comenzar();

while (!lista.fin()) {

ArbolGeneral<Integer> arbol = lista.proximo();

if (resta > 0)

cont = cont + contadorGematria(arbol, resta);

}

return cont;

}

}

}

package parcial;

import tp03.ejercicio4.ListaGenerica;

public class Gematria {

public static int contadorGematriaBis(ArbolGeneral<Integer> ag, int valor) {

Contador cont = new Contador();

contadorGematriaBis(ag, valor, contador, 0);

return cont.getValor();

}

private static void contadorGematriaBis(ArbolGeneral<Integer> ag, int valor,

Contador contador, int suma) {

if ((suma == valor) && (ag.esHoja())) {

contador.incrementar(); VERSION 2

} else {

ListaGenerica<ArbolGeneral<Integer>> lista = ag.getHijos();

lista.comenzar();

while (!lista.fin()) {

ArbolGeneral<Integer> arbol = lista.proximo();

if (suma + arbol.getDatoRaiz() <= valor)

contadorGematriaBis(arbol, valor, contador, suma + arbol.getDatoRaiz());

}

}

}