*Analisis*

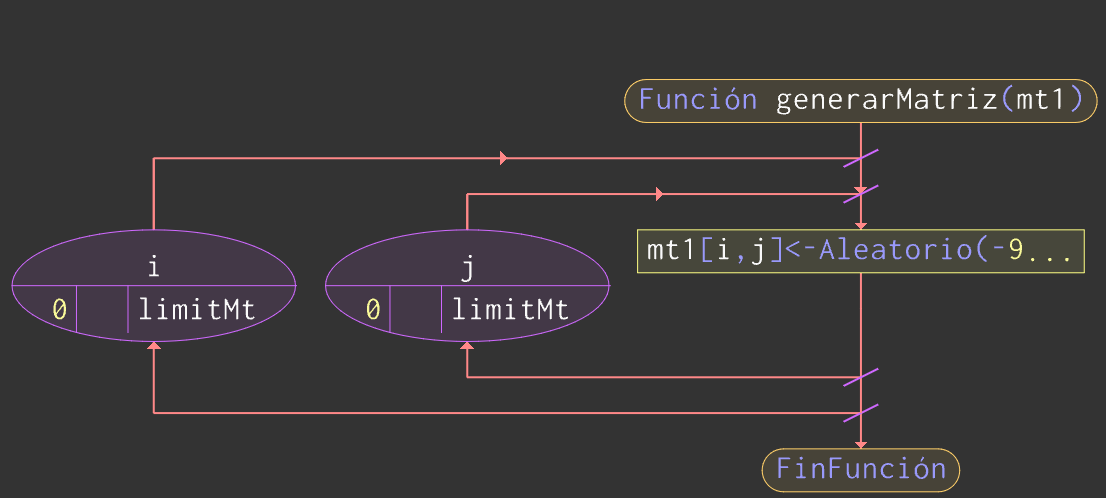
Para poder hacer los cálculos use procedimientos, en donde un procedimiento es solo la versión sobrecargada de el mismo procedimiento (sumarMatrices), y los presento a todos a través de una función que retorna valor en cadena (presentarMatrices), todos ellos son invocados desde el procedimiento principal o método main en java.

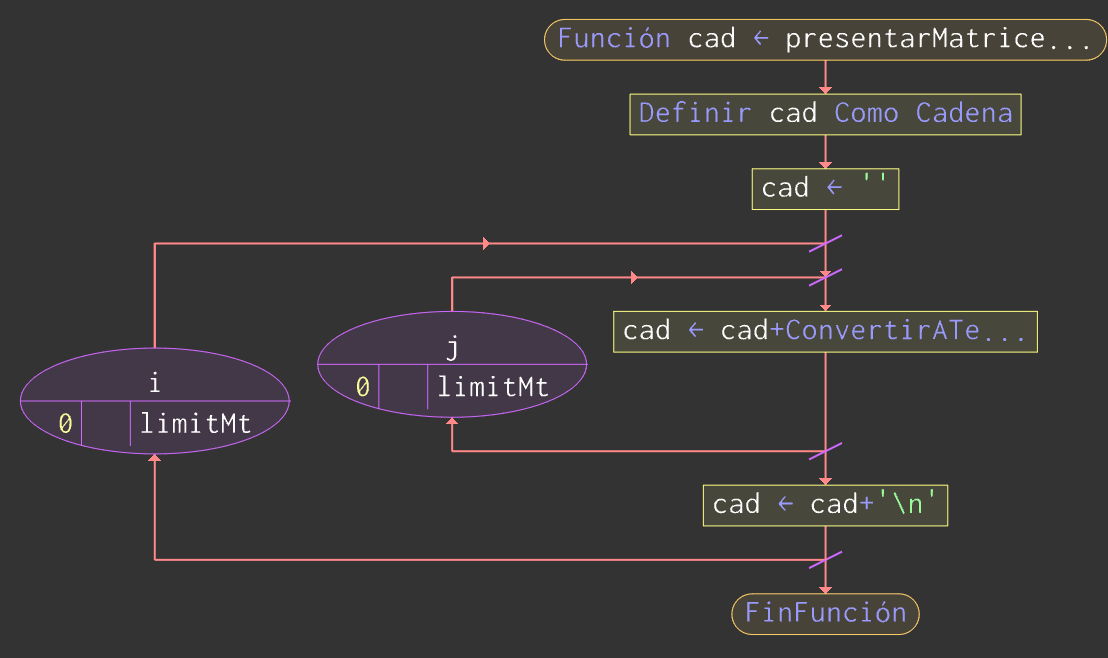
*Pseudocodigo:*

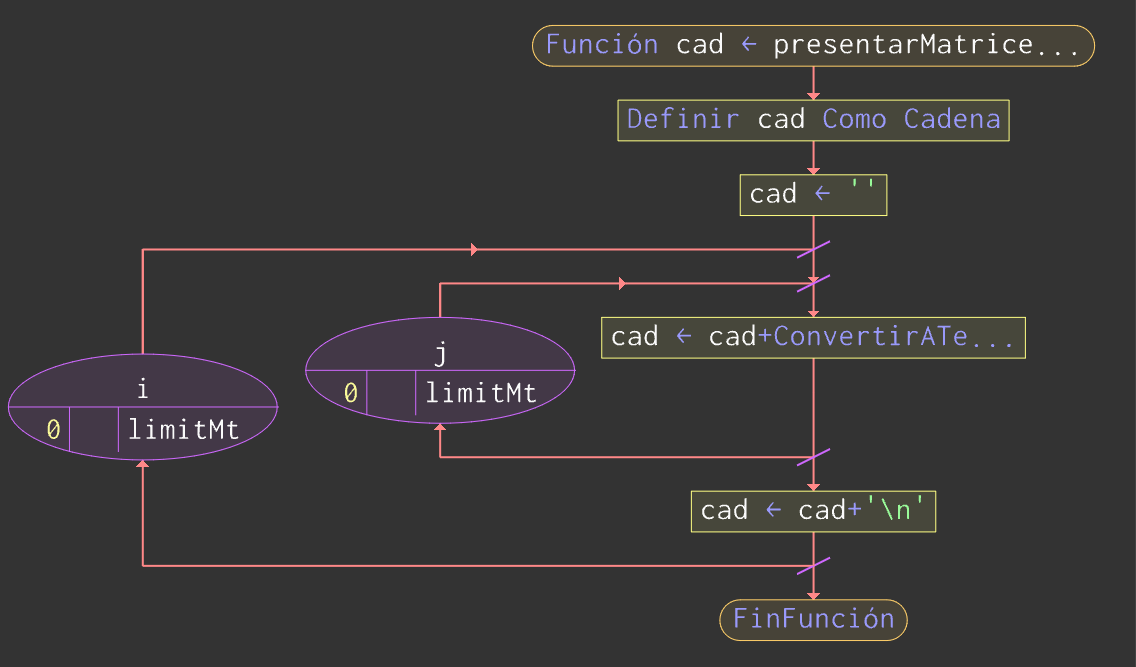
1. Algoritmo OperacionesMatrices
2. Definir limitMt Como Entero
3. Escribir "Ingrese tamaño de la matriz"
4. Leer limitMt
5. Dimension mt1[limitMt,limitMt]
6. Dimension mt2[limitMt,limitMt]
7. Dimension matrizResultado[limitMt,limitMt]
8. Dimension mtdouble[limitMt,limitMt]
9. Dimension mtdouble2[limitMt,limitMt]
10. Dimension mtResultadodouble[limitMt,limitMt]
11. generarMatriz(mt1)
12. generarMatriz(mt2)
13. Escribir "La matriz 1 generada es:"
14. Escribir presentarMatrices(mt1)
15. Escribir "La matriz 2 generada es:"
16. Escribir presentarMatrices(mt2)
17. sumarMatrices(mt1,mt2,matrizResultado)
18. Escribir "La suma de matrices es:"
19. Escribir presentarMatrices(matrizResultado)
20. restarMatrices(mt1,mt2,matrizResultado)
21. Escribir "La resta de matrices es:"
22. Escribir presentarMatrices(matrizResultado)
23. multiplicarMatrices(mt1,mt2,matrizResultado)
24. Escribir "La multiplicacion de matrices es:"
25. Escribir presentarMatrices(matrizResultado)
26. matrizInt\_a\_MatrizDouble(mt1,mtdouble, mtResultadodouble)
27. matrizInt\_a\_MatrizDouble(mt2,mtdouble2, mtResultadodouble)
28. dividirMatrices(mtdouble,mtdouble2,mtResultadodouble)
29. Escribir "La division de matrices es:"
30. Escribir presentarMatricesDouble(mtResultadodouble)
31. FinAlgoritmo
32. SubProceso generarMatriz(mt1)
33. Para i<-0 Hasta limitMt Hacer
    1. Para j<-0 Hasta limitMt Hacer
       1. mt1[i,j] <- Aleatorio(-9,10)
    2. FinPara
34. FinPara
35. FinSubProceso generarMatriz
36. Funcion presentarMatrices(mt1) : x(100)
37. Definir cad Como Cadena
38. cad <- ""
39. Para i<-0 Hasta limitMt Hacer
    1. Para j<-0 Hasta limitMt Hacer
       1. cad <- cad + ConvertirATexto(mt1[i,j]) + Tabulador
    2. FinPara
    3. cad <- cad + "\n"
40. Devolver cad
41. FinPara
42. FinFuncion presentarMatrices
43. Funcion presentarMatricesDouble(mtdouble) :x(100)
44. Definir cad Como Cadena
45. cad <- ""
46. Para i<-0 Hasta limitMt Hacer
    1. Para j<-0 Hasta limitMt Hacer
       1. cad <- cad + ConvertirATexto(mt1) + Tabulador
    2. FinPara
    3. cad <- cad + "\n"
47. Devolver cad
48. FinPara
49. FinFuncion presentarMatricesDouble
50. SubProceso sumarMatrices(mt1, mt2, matrizResultado)
51. Para i<-0 Hasta limitMt Hacer
    1. Para j<-0 Hasta limitMt Hacer
       1. matrizResultado[i,j] <- mt1[i,j] + mt2[i,j]
    2. FinPara
52. FinPara
53. FinSubProceso sumarMatrices
54. SubProceso restarMatrices(mt1, mt2, matrizResultado)
55. Para i<-0 Hasta limitMt Hacer
    1. Para j<-0 Hasta limitMt Hacer
       1. matrizResultado[i,j] <- mt1[i,j] - mt2[i,j]
    2. FinPara
56. FinPara
57. FinSubProceso restarMatrices
58. SubProceso multiplicarMatrices(mt1,mt2, matrizResultado)
59. Para i<-0 Hasta limitMt Hacer
    1. Para j<-0 Hasta limitMt Hacer
       * 1. matrizResultado[i,j] <- mt1[i,j] \* mt2[i,j]
    2. FinPara
60. FinPara
61. FinSubProceso multiplicarMatrices
62. SubProceso matrizInt\_a\_MatrizDouble(mt1, mt2, matrizResultado)
63. Para i<-0 Hasta limitMt Hacer
    1. Para j<-0 Hasta limitMt Hacer
       1. matrizResultado[i,j] <- ConvertirAReal (mt1[i,j])
    2. FinPara
64. FinPara
65. FinSubProceso matrizInt\_a\_MatrizDouble
66. SubProceso dividirMatrices(mtdouble, mtdouble2, mtResultadodouble)
67. Para i<-0 Hasta limitMt Hacer
    1. Para j<-0 Hasta limitMt Hacer
       1. Para k<-0 Hasta limitMt Hacer
          * 1. mtResultadodouble[i,j] <- mtResultadodouble[i,j] + mtdouble[i,k] / mtdouble2[k,j]
       2. FinSi
       3. FinPara
    2. FinPara
68. FinPara
69. FinSubProceso dividirMatrices

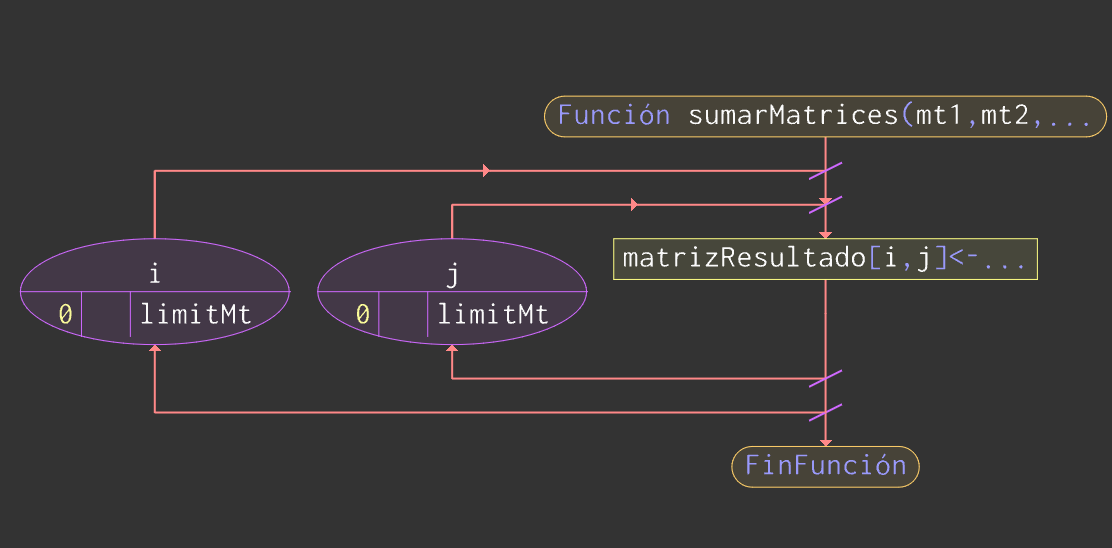
*Flujograma:*

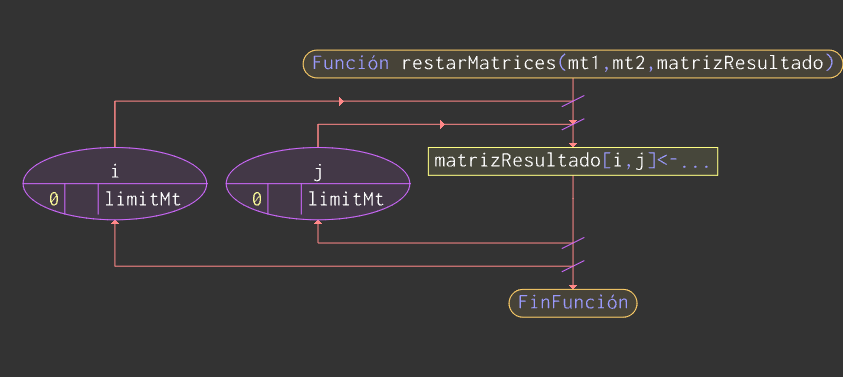


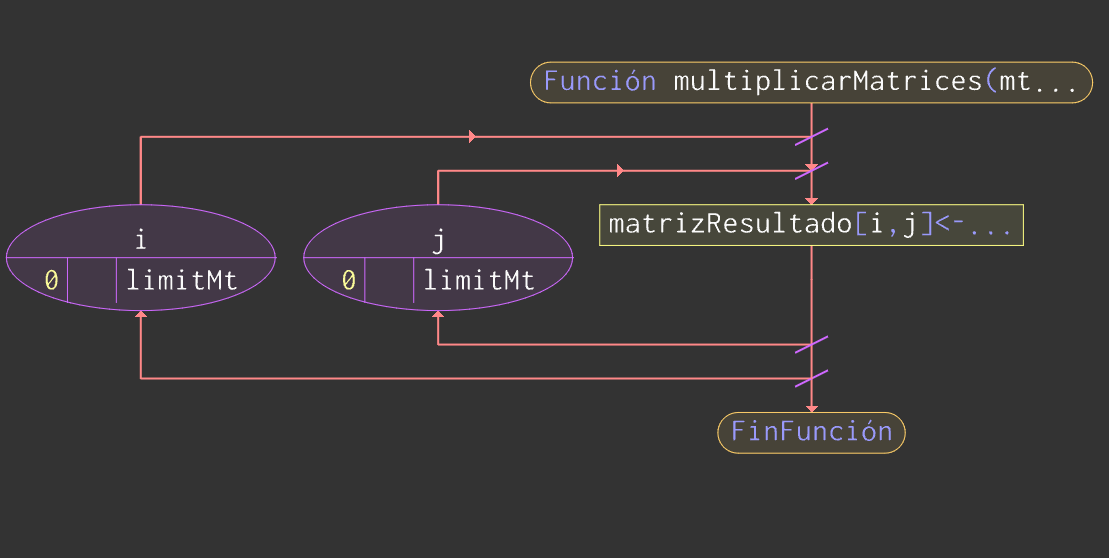


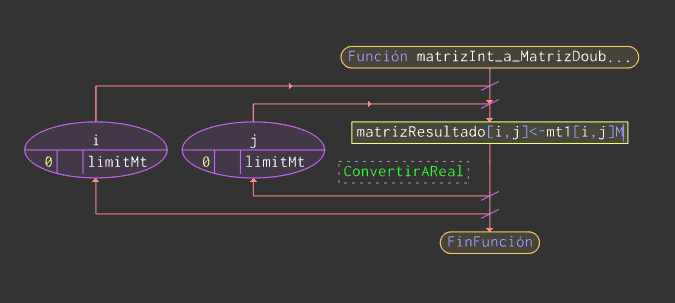


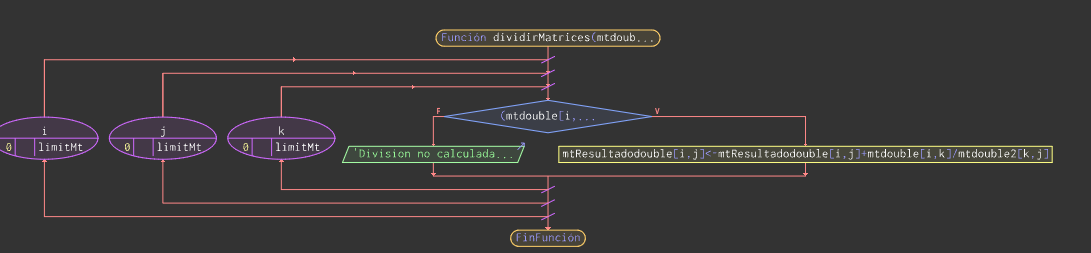












*Pruebas de Escritorio:*

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Nota: Cuando un numero de [i] tiene su valor de al lado vacio es porque avanza su posicion en [j] ( hice esto para evitar que sea largo el ejercicio como las anteriores pruebas de escritorio que he hecho) | | | | | | | | | |
| mt1 | | mt2 | | mtdouble | | mtdouble2 | | matrizResultado ( al multiplicar) | |
| [0] |  | [0] |  | [0] |  | [0] |  | [0] |  |
| [0] | 7 | [0] | 1 | [0] | 7 | [0] | 1 | [0] | 7 |
| [1] | -3 | [1] | 0 | [1] | -3 | [1] | 0 | [1] | 0 |
| [2] | 7 | [2] | -4 | [2] | 7 | [2] | -4 | [2] | -28 |
| [1] |  | [1] |  | [1] |  | [1] |  | [1] |  |
| [0] | 8 | [0] | -2 | [0] | 8 | [0] | -2 | [0] | -16 |
| [1] | -1 | [1] | 7 | [1] | -1 | [1] | 7 | [1] | -7 |
| [2] | -8 | [2] | -5 | [2] | -8 | [2] | -5 | [2] | 40 |
| [2] |  | [2] |  | [2] |  | [2] |  | [2] |  |
| [0] | -2 | [0] | -5 | [0] | -2 | [0] | -5 | [0] | 10 |
| [1] | 3 | [1] | 7 | [1] | 3 | [1] | 7 | [1] | 21 |
| [2] | -7 | [2] | 3 | [2] | -7 | [2] | 3 | [2] | -21 |