



Instituto Politécnico Nacional
UPIITA

Práctica 4: Arreglos y Matrices en Java

Unidad de Aprendizaje: Programación

Profesor: [Nombre del docente]

Alumno: Brayan Lucero Molina

Boleta: 2024640465

Grupo: 1TV3

Carrera: Ingeniería en Telemática

Fecha: 20 de octubre de 2025

Objetivo

Desarrollar programas utilizando arreglos y matrices.

Introducción

En esta práctica se trabajó con estructuras de datos tipo arreglo bidimensional o matriz. El objetivo fue aprender a almacenar, manipular y operar conjuntos de números organizados en filas y columnas.

Se desarrollaron dos programas principales:

- Una calculadora de matrices que realiza operaciones básicas como suma, resta, multiplicación, traspuesta e inversa.
- Un sistema que resuelve ecuaciones lineales de tres variables usando la inversa de una matriz 3×3 .

Estos ejercicios refuerzan la lógica de programación y el razonamiento matemático aplicado a problemas computacionales.

Desarrollo

Programa 1: CalculadoraMatrices.java — Operaciones entre matrices

Descripción: Este programa permite al usuario definir dos matrices cuadradas A y B de tamaño $n \times n$ y realizar operaciones entre ellas mediante un menú de opciones.

Pasos para construir el código:

1. Pedir al usuario el tamaño de las matrices (n).
2. Crear dos arreglos bidimensionales: $A[n][n]$ y $B[n][n]$.
3. Solicitar los valores de cada matriz.
4. Mostrar un menú con opciones: suma, resta, producto punto, multiplicación, traspuesta e inversa.
5. Realizar la operación seleccionada usando ciclos `for`.
6. Mostrar el resultado en pantalla.
7. Pausar la ejecución para que el usuario vea los resultados antes de regresar al menú.

```
1 /**
2  * @file menu.java
3  * @author Brayan Lucero Molina
4  * @brief Programa en Java que realiza operaciones con matrices
5  *         cuadradas
6  *         como suma, resta, producto punto, multiplicación,
7  *         traspuesta e inversa (para 3x3).
8  * @version 0.1
9  * @date 2025-10-18
10  *
11  * @copyright Copyright (c) 2025
12  */
13
14 import java.util.Scanner;
15
16 public class CalculadoraMatrices {
```

```

15
16 public static void main(String[] args) {
17     Scanner sc = new Scanner(System.in);
18     int n;
19
20     // Solicita el tamaño de las matrices cuadradas
21     System.out.print("Ingresa el tamaño de las matrices n x n: ");
22     n = sc.nextInt();
23
24     // Declaración de matrices A y B
25     double[][] A = new double[n][n];
26     double[][] B = new double[n][n];
27
28     // Lectura de los valores de la matriz A
29     System.out.println("Ingresa los valores de la matriz A:");
30     for (int i = 0; i < n; i++) {
31         for (int j = 0; j < n; j++) {
32             System.out.print("A[" + i + "][" + j + "]: ");
33             A[i][j] = sc.nextDouble();
34         }
35     }
36
37     // Lectura de los valores de la matriz B
38     System.out.println("Ingresa los valores de la matriz B:");
39     for (int i = 0; i < n; i++) {
40         for (int j = 0; j < n; j++) {
41             System.out.print("B[" + i + "][" + j + "]: ");
42             B[i][j] = sc.nextDouble();
43         }
44     }
45
46     int opcion;
47     // Ciclo principal del menú
48     do {
49         System.out.println("\n=====");
50         System.out.println("      CALCULADORA DE MATRICES      ");
51         System.out.println("=====");
52         System.out.println("1. Suma");
53         System.out.println("2. Resta");
54         System.out.println("3. Producto punto (1ra fila A y 1ra");
55         System.out.println("   columna B)");
56         System.out.println("4. Multiplicación de matrices");
57         System.out.println("5. Traspuesta de A");
58         System.out.println("6. Traspuesta de B");
59         System.out.println("7. Inversa de A (solo si 3x3)");
60         System.out.println("8. Inversa de B (solo si 3x3)");
61         System.out.println("0. Salir");
62         System.out.print("Elige una opción: ");

```

```

62     opcion = sc.nextInt();
63
64     // Men  de operaciones
65     switch (opcion) {
66
67         case 1 -> {
68             // SUMA de matrices A + B
69             System.out.println("\nHas elegido SUMA de matrices.");
70             ;
71             double[][] suma = new double[n][n];
72             for (int i = 0; i < n; i++) {
73                 for (int j = 0; j < n; j++) {
74                     suma[i][j] = A[i][j] + B[i][j];
75                 }
76             }
77             System.out.println("Resultado de A + B:");
78             imprimirMatriz(suma);
79             pausa(sc);
80         }
81
82         case 2 -> {
83             // RESTA de matrices A - B
84             System.out.println("\nHas elegido RESTA de matrices.");
85             );
86             double[][] resta = new double[n][n];
87             for (int i = 0; i < n; i++) {
88                 for (int j = 0; j < n; j++) {
89                     resta[i][j] = A[i][j] - B[i][j];
90                 }
91             }
92             System.out.println("Resultado de A - B:");
93             imprimirMatriz(resta);
94             pausa(sc);
95         }
96
97         case 3 -> {
98             // PRODUCTO PUNTO entre primera fila de A y primera
99             // columna de B
100             System.out.println("\nHas elegido PRODUCTO PUNTO.");
101             double producto = 0;
102             for (int i = 0; i < n; i++) {
103                 producto += A[0][i] * B[i][0];
104             }
105             System.out.println("Resultado del producto punto: " +
106                 producto);
107             pausa(sc);
108         }
109     }

```

```

106     case 4 -> {
107         // MULTIPLICACION de matrices A x B
108         System.out.println("\nHas elegido MULTIPLICACION de
109             matrices.");
110         double[][] mult = new double[n][n];
111         for (int i = 0; i < n; i++) {
112             for (int j = 0; j < n; j++) {
113                 for (int k = 0; k < n; k++) {
114                     mult[i][j] += A[i][k] * B[k][j];
115                 }
116             }
117         }
118         System.out.println("Resultado de A x B:");
119         imprimirMatriz(mult);
120         pausa(sc);
121     }
122
123     case 5 -> {
124         // TRASPUESTA de la matriz A
125         System.out.println("\nHas elegido la TRASPUESTA de A.
126             ");
127         double[][] tA = new double[n][n];
128         for (int i = 0; i < n; i++) {
129             for (int j = 0; j < n; j++) {
130                 tA[i][j] = A[j][i];
131             }
132         }
133         System.out.println("Traspuesta de A:");
134         imprimirMatriz(tA);
135         pausa(sc);
136     }
137
138     case 6 -> {
139         // TRASPUESTA de la matriz B
140         System.out.println("\nHas elegido la TRASPUESTA de B.
141             ");
142         double[][] tB = new double[n][n];
143         for (int i = 0; i < n; i++) {
144             for (int j = 0; j < n; j++) {
145                 tB[i][j] = B[j][i];
146             }
147         }
148         System.out.println("Traspuesta de B:");
149         imprimirMatriz(tB);
150         pausa(sc);
151     }
152
153     case 7 -> {

```

```

151         // INVERSA de la matriz A (solo 3x3)
152         System.out.println("\nHas elegido la INVERSA de A.");
153         if (n == 3) {
154             double[][] invA = inversa3x3(A);
155             System.out.println("Inversa de A:");
156             imprimirMatriz(invA);
157         } else {
158             System.out.println("Solo funciona con matrices de
159                               3x3.");
160         }
161         pausa(sc);
162     }
163     case 8 -> {
164         // INVERSA de la matriz B (solo 3x3)
165         System.out.println("\nHas elegido la INVERSA de B.");
166         if (n == 3) {
167             double[][] invB = inversa3x3(B);
168             System.out.println("Inversa de B:");
169             imprimirMatriz(invB);
170         } else {
171             System.out.println("Solo funciona con matrices de
172                               3x3.");
173         }
174         pausa(sc);
175     }
176     case 0 -> System.out.println("\nGracias por usar la
177                               calculadora de matrices.");
178     default -> {
179         // Validaci n de opci n inv lida
180         System.out.println("\nOpci n inv lida , intenta de
181                               nuevo.");
182         pausa(sc);
183     }
184 }
185 } while (opcion != 0); // Fin del men
186 }
187
188 /**
189  * Imprime una matriz en formato tabular.
190  * @param M matriz a imprimir
191  */
192 public static void imprimirMatriz(double[][] M) {
193     for (double[] fila : M) {
194         for (int j = 0; j < M[0].length; j++) {

```



```

195         System.out.print(fila[j] + "\t");
196     }
197     System.out.println();
198 }
199 }
200
201 /**
202  * Calcula la inversa de una matriz 3x3 utilizando la matriz adjunta
203  * y el determinante.
204  * @param m matriz original 3x3
205  * @return matriz inversa 3x3
206  */
207 public static double[][] inversa3x3(double[][] m) {
208     double det = determinante3x3(m);
209     if (det == 0) {
210         System.out.println("La matriz no tiene inversa");
211         return new double[3][3];
212     }
213     double[][] inv = new double[3][3];
214     inv[0][0] = (m[1][1]*m[2][2]-m[1][2]*m[2][1])/det;
215     inv[0][1] = -(m[0][1]*m[2][2]-m[0][2]*m[2][1])/det;
216     inv[0][2] = (m[0][1]*m[1][2]-m[0][2]*m[1][1])/det;
217     inv[1][0] = -(m[1][0]*m[2][2]-m[1][2]*m[2][0])/det;
218     inv[1][1] = (m[0][0]*m[2][2]-m[0][2]*m[2][0])/det;
219     inv[1][2] = -(m[0][0]*m[1][2]-m[0][2]*m[1][0])/det;
220     inv[2][0] = (m[1][0]*m[2][1]-m[1][1]*m[2][0])/det;
221     inv[2][1] = -(m[0][0]*m[2][1]-m[0][1]*m[2][0])/det;
222     inv[2][2] = (m[0][0]*m[1][1]-m[0][1]*m[1][0])/det;
223     return inv;
224 }
225
226 /**
227  * Calcula el determinante de una matriz 3x3.
228  * @param m matriz 3x3
229  * @return determinante de la matriz
230  */
231 public static double determinante3x3(double[][] m) {
232     return m[0][0]*(m[1][1]*m[2][2]-m[1][2]*m[2][1])
233         - m[0][1]*(m[1][0]*m[2][2]-m[1][2]*m[2][0])
234         + m[0][2]*(m[1][0]*m[2][1]-m[1][1]*m[2][0]);
235 }
236
237 /**
238  * Pausa la ejecuci n hasta que el usuario presione Enter.
239  * @param sc objeto Scanner para leer la entrada
240  */
241 public static void pausa(Scanner sc) {
242     System.out.println("\nPresiona Enter para continuar...");

```

```

242     sc.nextLine();
243     sc.nextLine();
244 }
245
246 }
```

Listing 1: CalculadoraMatrices.java

```

=====
      CALCULADORA DE MATRICES
=====
1. Suma
2. Resta
3. Producto punto (1ra fila A y 1ra columna B)
4. Multiplicación de matrices
5. Traspuesta de A
6. Traspuesta de B
7. Inversa de A (solo si 3x3)
8. Inversa de B (solo si 3x3)
0. Salir
Elige una opción: █
```

```

Elige una opción: 1

Has elegido SUMA de matrices.
Resultado de A + B:
10.0    10.0    10.0
10.0    10.0    10.0
10.0    10.0    10.0

Presiona Enter para continuar...
█
```

Figura 1: Izquierda: Menú principal. Derecha: Ejemplo de suma de matrices.

En la Figura 1 se observa el menú del programa, que ofrece diversas operaciones, y un ejemplo de suma de dos matrices 2×2 .

```

Elige una opción: 4

Has elegido MULTIPLICACIÓN de matrices.
Resultado de A x B:
30.0    24.0    18.0
84.0    69.0    54.0
138.0   114.0   90.0

Presiona Enter para continuar...
█
```

```

Has elegido la INVERSA de A.
La matriz no tiene inversa
Inversa de A:
0.0     0.0     0.0
0.0     0.0     0.0
0.0     0.0     0.0

Presiona Enter para continuar...
█
```

Figura 2: Izquierda: Multiplicación de matrices. Derecha: Cálculo de inversa (3×3).

En la Figura 2 se muestra cómo el programa realiza una multiplicación matricial y el proceso para obtener la inversa de una matriz 3×3 .

Comentarios sobre el código:

- Se implementaron funciones `imprimirMatriz`, `inversa3x3` y `determinante3x3`.
- El menú se mantiene activo con un ciclo `do-while`.
- Se agregó una pausa después de cada operación para facilitar la lectura de resultados.

Programa 2: SistemaEcuaciones.java — Solución de un sistema 3x3

Descripción: Resuelve un sistema de tres ecuaciones lineales con tres incógnitas usando el método matricial $A^{-1}B$.

$$\begin{cases} A_{11}x_1 + A_{12}x_2 + A_{13}x_3 = B_1 \\ A_{21}x_1 + A_{22}x_2 + A_{23}x_3 = B_2 \\ A_{31}x_1 + A_{32}x_2 + A_{33}x_3 = B_3 \end{cases}$$

Pasos para construir el código:

1. Pedir los valores de la matriz $A(3 \times 3)$ y el vector $B(3 \times 1)$.
2. Calcular el determinante de A para verificar si tiene inversa.
3. Obtener la matriz inversa de A .
4. Multiplicar A^{-1} por B para encontrar el vector X .
5. Mostrar el resultado con las tres incógnitas.

```
1  /**
2   * @file menu.java
3   * @author Brayan Lucero Molina
4   * @brief Programa en Java que resuelve sistemas de ecuaciones
5   *         lineales
6   *         utilizando el metodo de eliminacion de Gauss y el metodo de
7   *         Cramer.
8   * @version 0.1
9   * @date 2025-10-19
10  */
11
12  import java.util.Scanner;
13
14  public class SistemaEcuaciones {
15
16  public static void main(String[] args) {
17      Scanner sc = new Scanner(System.in);
18      int n;
19
20      // Pido al usuario el numero de ecuaciones (y variables)
21      System.out.print("Ingresa el numero de ecuaciones (y variables):
22      ");
23      n = sc.nextInt();
24
25      // Defino la matriz de coeficientes A y el vector de terminos
26      independientes b
```

```

23 double[][] A = new double[n][n];
24 double[] b = new double[n];
25
26 // Solicito los datos del sistema
27 System.out.println("\nIngresa los coeficientes del sistema:");
28 for (int i = 0; i < n; i++) {
29     System.out.println("Ecuacion " + (i + 1) + ":");
30     for (int j = 0; j < n; j++) {
31         System.out.print("a[" + i + "][" + j + "]: ");
32         A[i][j] = sc.nextDouble();
33     }
34     System.out.print("Termino independiente b[" + i + "]: ");
35     b[i] = sc.nextDouble();
36 }
37
38 int opcion;
39 // Ciclo principal del menu
40 do {
41     System.out.println("\n=====");
42     System.out.println("    SISTEMA DE ECUACIONES    ");
43     System.out.println("=====");
44     System.out.println("1. Mostrar sistema ingresado");
45     System.out.println("2. Resolver por metodo de Gauss");
46     System.out.println("3. Resolver por metodo de Cramer (solo si");
47         3x3");
48     System.out.println("0. Salir");
49     System.out.print("Elige una opcion: ");
50     opcion = sc.nextInt();
51
52     // Uso de switch para seleccionar la operacion
53     switch (opcion) {
54         case 1 -> {
55             // Muestra el sistema que el usuario introdujo
56             System.out.println("\nSistema de ecuaciones:");
57             mostrarSistema(A, b);
58             pausa(sc);
59         }
60
61         case 2 -> {
62             // Resuelve el sistema usando eliminacion de Gauss
63             System.out.println("\nResolviendo por metodo de Gauss");
64                 :");
65             double[] xGauss = gauss(A, b);
66             System.out.println("Solucion (Gauss):");
67             imprimirVector(xGauss);
68             pausa(sc);
69         }
70     }
71 }

```

```

69
70     case 3 -> {
71         // Resuelve por metodo de Cramer (solo si es 3x3)
72         if (n == 3) {
73             System.out.println("\nResolviendo por metodo de
74                 Cramer:");
75             double[] xCramer = cramer(A, b);
76             System.out.println("Solucion (Cramer):");
77             imprimirVector(xCramer);
78         } else {
79             System.out.println("El metodo de Cramer solo
80                 funciona con sistemas 3x3.");
81         }
82         pausa(sc);
83     }
84
85     case 0 -> System.out.println("\nGracias por usar el
86         sistema de ecuaciones.");
87
88     default -> {
89         // Opcion no valida
90         System.out.println("\nOpcion invalida, intenta de
91             nuevo.");
92         pausa(sc);
93     }
94 }
95
96 } while (opcion != 0); // Repite hasta que el usuario elija salir
97
98 /**
99  * Muestra el sistema de ecuaciones de forma legible.
100  */
101 public static void mostrarSistema(double[][] A, double[] b) {
102     for (int i = 0; i < A.length; i++) {
103         for (int j = 0; j < A[0].length; j++) {
104             System.out.print(A[i][j] + "x" + (j + 1));
105             if (j < A[0].length - 1) System.out.print(" + ");
106         }
107         System.out.println(" = " + b[i]);
108     }
109 }
110
111 /**
112  * Metodo de Eliminacion de Gauss.
113  * Convierte la matriz en forma triangular superior y luego aplica
114     sustitucion regresiva.
115  */

```

```

112 public static double[] gauss(double[][] A, double[] b) {
113     int n = A.length;
114
115     // Clono las matrices para no modificar las originales
116     double[][] M = new double[n][n];
117     double[] B = new double[n];
118     for (int i = 0; i < n; i++) {
119         System.arraycopy(A[i], 0, M[i], 0, n);
120         B[i] = b[i];
121     }
122
123     // Etapa de eliminacion hacia adelante
124     for (int k = 0; k < n; k++) {
125         // Busco el pivote mas grande en la columna
126         double max = Math.abs(M[k][k]);
127         int filaMax = k;
128         for (int i = k + 1; i < n; i++) {
129             if (Math.abs(M[i][k]) > max) {
130                 max = Math.abs(M[i][k]);
131                 filaMax = i;
132             }
133         }
134
135         // Intercambio de filas si es necesario (pivoteo)
136         double[] temp = M[k];
137         M[k] = M[filaMax];
138         M[filaMax] = temp;
139         double t = B[k];
140         B[k] = B[filaMax];
141         B[filaMax] = t;
142
143         // Eliminacion de las variables inferiores
144         for (int i = k + 1; i < n; i++) {
145             double factor = M[i][k] / M[k][k];
146             for (int j = k; j < n; j++) {
147                 M[i][j] -= factor * M[k][j];
148             }
149             B[i] -= factor * B[k];
150         }
151     }
152
153     // Sustitucion regresiva
154     double[] x = new double[n];
155     for (int i = n - 1; i >= 0; i--) {
156         double suma = B[i];
157         for (int j = i + 1; j < n; j++) {
158             suma -= M[i][j] * x[j];
159         }

```

```

160         x[i] = suma / M[i][i];
161     }
162     return x;
163 }
164
165 /**
166  * Metodo de Cramer (solo para sistemas 3x3).
167  * Usa determinantes para calcular cada variable.
168  */
169 public static double[] cramer(double[][] A, double[] b) {
170     double detA = determinante3x3(A);
171     if (detA == 0) {
172         System.out.println("El sistema no tiene solucion unica.");
173         return new double[3];
174     }
175
176     // Creo matrices reemplazando las columnas por el vector b
177     double[][] Ax = reemplazarColumna(A, b, 0);
178     double[][] Ay = reemplazarColumna(A, b, 1);
179     double[][] Az = reemplazarColumna(A, b, 2);
180
181     // Calculo las variables
182     double[] x = new double[3];
183     x[0] = determinante3x3(Ax) / detA;
184     x[1] = determinante3x3(Ay) / detA;
185     x[2] = determinante3x3(Az) / detA;
186
187     return x;
188 }
189
190 /**
191  * Reemplaza una columna de la matriz A por el vector b.
192  */
193 public static double[][] reemplazarColumna(double[][] A, double[] b,
194     int col) {
195     double[][] M = new double[3][3];
196     for (int i = 0; i < 3; i++) {
197         System.arraycopy(A[i], 0, M[i], 0, 3);
198         M[i][col] = b[i];
199     }
200     return M;
201 }
202
203 /**
204  * Calcula el determinante de una matriz 3x3.
205  */
206 public static double determinante3x3(double[][] m) {
207     return m[0][0]*(m[1][1]*m[2][2] - m[1][2]*m[2][1])

```

```

207         - m[0][1]*(m[1][0]*m[2][2] - m[1][2]*m[2][0])
208         + m[0][2]*(m[1][0]*m[2][1] - m[1][1]*m[2][0]);
209     }
210
211     /**
212     * Imprime el vector solucion en pantalla.
213     */
214     public static void imprimirVector(double[] x) {
215         for (int i = 0; i < x.length; i++) {
216             System.out.println("x" + (i + 1) + " = " + x[i]);
217         }
218     }
219
220     /**
221     * Espera que el usuario presione Enter antes de continuar.
222     */
223     public static void pausa(Scanner sc) {
224         System.out.println("\nPresiona Enter para continuar...");
225         sc.nextLine();
226         sc.nextLine();
227     }
228
229 }

```

Listing 2: SistemaEcuaciones.java

```

Ingresa el numero de ecuaciones (y variables): 2

Ingresa los coeficientes del sistema:
Ecuacion 1:
a[0][0]: 2
a[0][1]: 3
Termino independiente b[0]: 8
Ecuacion 2:
a[1][0]: 1
a[1][1]: -1
Termino independiente b[1]: 1

```

```

1. Mostrar sistema ingresado
2. Resolver por metodo de Gauss
3. Resolver por metodo de Cramer (solo si 3x3)
0. Salir
Elige una opcion: 2

Resolviendo por metodo de Gauss:
Solucion (Gauss):
x1 = 2.2
x2 = 1.2

```

Figura 3: Izquierda: Entrada de coeficientes. Derecha: Resultado de las incógnitas.

En la Figura 3 se visualiza la captura de datos para los coeficientes y la salida de los valores calculados para x_1, x_2, x_3 .

Comentarios sobre el código:

- Se reutilizó la función de inversa 3x3 del primer programa.
- El resultado del vector X se imprimió en formato columna.
- Se validó que la matriz A no tenga determinante cero antes de intentar invertirla.

Ejecución

Programa 1

Se realizaron pruebas con matrices de distintos tamaños para verificar cada operación:

- Matrices 2x2 para probar suma.
- Matrices 3x3 para multiplicación.

```
Elige una opción: 1

Has elegido SUMA de matrices.
Resultado de A + B:
6.0      8.0
10.0     12.0
```

```
Elige una opción: 4

Has elegido MULTIPLICACIÓN de matrices.
Resultado de A x B:
30.0    24.0    18.0
84.0    69.0    54.0
138.0   114.0   90.0
```

Figura 4: Ejecución de operaciones de suma e inversa.

La Figura 4 muestra los resultados obtenidos al realizar la suma de dos matrices y la multiplicación de una matriz 3x3

Programa 2

Se probó el sistema:

$$\begin{cases} 4x - 2y + 6z = 8 \\ 2x + 8y + 2z = 4 \\ 6x + 10y + 3z = 0 \end{cases}$$

En la Figura 5 el programa resuelve correctamente el sistema y muestra los valores de las incógnitas calculadas por gauss y cramer

```
Resolviendo por metodo de Gauss:  
Solucion (Gauss):  
x1 = -1.8048780487804879  
x2 = 0.2926829268292684  
x3 = 2.6341463414634148
```

```
Resolviendo por metodo de Cramer:  
Solucion (Cramer):  
x1 = -1.8048780487804879  
x2 = 0.2926829268292683  
x3 = 2.6341463414634148
```

Figura 5: Ejecución del sistema de ecuaciones 3x3.

Conclusiones

Esta práctica permitió aplicar los conocimientos sobre arreglos bidimensionales y operaciones entre matrices. Se comprendió cómo implementar métodos matemáticos como la obtención de la inversa y la multiplicación de matrices dentro de un programa en Java.