Reynaldo Hasaan Juarez Aranda Eliminación de Gauss | D 20 | M 10 | A 2025

```
Inicio
//Leer dimensión del sistema
int N
Imprimir ("Leer dimensión:")
Scanf("%d", &N)
double A[N][N+1] //Matriz aumentada
double x[N]
//Leer coeficientes A y b
Para i=0 hasta N-1
   Para j=0 hasta N
      Scanf(A[i][j])
  Fin
  Fin
//Pivoteo parcial
Para i=0 hasta N-2
max=i
Para K=i+1 hasta N-1
   Si Fabs(A[K][i])>Fabs(A[max][i])
Entonces max=K
Fin
   Intercambiar fila i con fila max
Fin
//Eliminación hacia adelante y normalización
Para i=0 hasta N-2
 Pivote=A[i][i]
Dividir fila i entre pivote
Para K=i+1 hasta N-1
 Factor=A[K][i]
 Restar factor * fila i de fila K
Fin
Fin
//Última fila
Pivote final=A[N-1][N-1]
Dividir fila N-1 entre pivote final
//Sustitución hacia atrás
Para i=N-1 hasta 0
 x[i]=A[i][N]
 Para j=i+1 hasta N-1
 x[i] -= A[i][j] * x[j]
 Fin
 Fin
//Mostrar solución
Para i=0 hasta N-1
Imprimir ("x %d=", i+1, x[i])
Fin
//Condición del sistema
minPivote=Fabs(A[0][0])
maxPivote=Fabs(A[0][0])
Para i=1 hasta N-1
p=Fabs(A[i][i])
Si p<minPivote entonces minPivote=p
Si p>maxPivote entonces maxPivote=p
Fin
Condición=maxPivote/minPivote
Si condición > 1e6
   Imprimir("sistema mal condicionado")
Sino ("sistema bien condicionado")
Fin
```

Metodo Gauss - Jordan

```
Inicio
 int N
Imprimir("Leer dimension:")
scanf("%d",&N)
double A[N][N+1],x[N]
//Leer matriz aumentada
 Para i=0 hasta N-1
   Para j=0 hasta N
     scanf(A[i][j])
  Fin
  Fin
//Gauss Jordan con pivoteo parcial
 Para i=0 hasta N-1
//Pivoteo parcial
max=i
Para k=i+1 hasta N-1
Si fabs(A[k][i])>fabs(A[max][i])
entonces max=k
Fin
Intercambiar fila i con fila max
//Normalizar fila pivote
pivote=A[i][i]
Dividir fila i entre pivote
//Eliminar columna i en todas las filas (menos la ya hecha)
Para k=0 hasta N-1
 Si k != i Entonces
 factor = A[k][i]
 Restar factor * Fila i de fila k
Fin
Fin
Fin
//Extraer solución
 Para i=0 hasta N-1
 x[i]=A[i][N]
  Fin
//Mostrar matriz final y solución
   Para i=0 hasta N-1
    Para j=0 hasta N
     Imprimir (A[i][j])
    Fin
  Para i=0 hasta N-1
  Imprimir (x[%d]=",i+1 x[i])
  Fin
//Evaluar condición del sistema
  minPivote=fabs(A[0][0])
  maxPivote=fabs(A[0][0])
   Para i=1 hasta N-1
    p=fabs[A[i][i]]
  Si p<minpivote entonces minpivote=p
  Si p>maxpivote entonces maxpivote=p
  Fin
  Para condicion=maxpivote/minPivote
  Si condicion>1 e6
   Imprimir("sistema mal condicionado")
  Sino
   Imprimir("sistema bien condicionado")
   Fin
```