

```

1 program questao2
2   implicit none
3   real, dimension(10000, 10000) :: A
4   real, dimension(10000) :: b, x
5   integer :: n, i, niter
6
7   n = 3
8
9   print *, "Digite a ordem do sistem de equações:"
10  read (*,*) n
11
12  call readA(A, n)
13  call readB(b, n)
14  call initX(x, n)
15
16  print *, "Insira a quantidade de iterações que serão executadas"
17  read (*,*) niter
18
19  do i = 1, niter
20    call seidel(n, A, b, x)
21  end do
22
23  print *, "Resultado do sistema de equações lineares"
24  do i = 1, 3
25    print *, x(i)
26  end do
27
28
29  ! Pode-se notar que o método do jordan é mais preciso e demanda recursos
    menores para o processamento
30 end program questao2
31
32 subroutine initX(x, n)
33   implicit none
34   real, dimension(10000) :: x
35   integer :: n, i
36   do i = 1, n
37     x(i) = 0
38   end do
39 end subroutine initX
40
41 subroutine readA(A, n)
42   implicit none
43   real,dimension(10000, 10000) :: A
44   integer :: n, i, j
45   print *, "Insira os valores dos elementos do sistema na equação conforme
    solicitado"
46   do i = 1, n
47     do j = 1, n
48       print *, "Elemento (", i, j, ")"
49       read (*,*) A(i, j)
50     end do
51   end do
52 end subroutine readA
53
54 subroutine readB(b, n)
55   implicit none
56   real, dimension(10000) :: b
57   integer :: n, i

```

```

58  print *, "Insira os valores dos elementos do vetor de termos independentes
da equação"
59  do i = 1, n
60      print *, "Elemento (",i,")"
61      read (*,*) b(i)
62  end do
63  end subroutine readB
64
65  subroutine seidel(n, A, b, x)
66      implicit none
67      real, dimension(10000, 10000) :: A
68      real, dimension(10000) :: b, x
69      real :: d
70      integer :: i, j, n
71
72      do j = 1, n
73          d = b(j)
74          do i = 1, n
75              if ( j.ne.i ) d = d-a(j,i)*x(i)
76          end do
77          x(j) = d/A(j,j)
78      end do
79
80  end subroutine seidel

```