

```
A=[1 1 -1;0 1 3; -1 0 2]
```

```
A = 3x3
     1     1    -1
     0     1     3
    -1     0     2
```

```
b= [9;3;2]
```

```
b = 3x1
     9
     3
     2
```

```
[L,U,P]=LUP(A)
```

```
L = 3x3
     1     0     0
     0     1     0
    -1     1     1
U = 3x3
     1     1    -1
     0     1     3
     0     0    -2
P = 3x3
     1     0     0
     0     1     0
     0     0     1
```

```
x=sustituciones(L,U,P,b)
```

```
x = 3x1
    -10
     15
     -4
```

```
Ainv=inv_LUP(A)
```

```
Ainv = 3x3
    -1.0000    1.0000   -2.0000
     1.5000   -0.5000    1.5000
    -0.5000    0.5000   -0.5000
```

```
function [L,U,P] = LUP(A)

[n,m] = size(A);
if n~=m, error('Matriz A debe ser cuadrada'), end
L = eye(n);
P = eye(n);
U = A;

for i = 1:n-1
    [~,r] = max(abs(U(i:m,i)));
    col = i+(r-1);
    if U(col,i) == 0, error('Matriz es singular'), end
    if i~= col
        U([i, col],:) = U([col, i],:); %Igual a Gauss
    end
    %Pivoting
    P(i,col) = P(i,i);
    P(i,i) = P(col,col);
    %Row swapping
    U([i, col],:) = U([col, i],:);
    %Zeroing out below
    for k = i+1:n
        U(k,i) = 0;
    end
    %Row scaling
    U(i,i) = 1;
    for k = i+1:n
        U(k,i) = 0;
    end
end
```

```

        P([i, col], :) = P([col, i], :); %Intercambiamos cada vez que se cambien los renglones
        L([i, col], 1:i-1) = L([col, i], 1:i-1); %Cambiamos el valor de la matriz equivalente
    end

    for h = i+1 : n
        mult = U(h,i)/U(i,i); %Igual a Gauss
        L(h,i) = mult; %Asignamos el valor a L
        subrenglon = i:n; %Igual a Gauss
        U(h,subrenglon) = U(h,subrenglon) - mult*U(i,subrenglon);

    end
end

if U(n,n) == 0, error('Matriz es singular'), end

end

function x=sustituciones(L,U,P,b)
    %Checar que las dimensiones concuerden...
    [m,~]=size(P);
    if m~=size(b), error('Las dimensiones del vector b no concuerdan con las de la matriz'), end

    %Ax=b
    %PAx=LUx; L(Ux)=P*b; Ly=P*b
    %Primer paso L*y=P*b=Pb
    Pb= P*b;
    y=L\Pb;
    %Segundo paso U*x=y
    x=U\y;
end

function A_inv=inv_LUP(A)
    [n,m] = size(A);
    if n~=m
        error('Matriz no es cuadrada')
    end
    [L,U,P] = LUP(A);
    %PA=LU
    %P=L*U*A_inv
    %P*L_inv=U*A_inv
    A_inv = L\P;
    %Res=U*A_inv
    %Res*U_inv=A_inv
    A_inv= U\A_inv;
end

```