

```

:ClrHome
:Disp "FLUJO DE CARGA"
:Disp "Newton-Raphson Mejorado"

:Prompt N // Número de barras
:Prompt Imax // Máx. iteraciones
:Prompt Tol // Tolerancia

// Tipos de barras
:For(A,1,N)
:Disp "Tipo Barra ", A
:Disp "1=PQ, 2=PV, 3=Slack"
:Prompt T(A) // 1 para PQ, 2 para PV, 3 para Slack
:End

// Admitancia Y-bus
:For(A,1,N)
:For(B,1,N)
:Prompt YR(A,B) // Parte real de la admitancia
:Prompt YI(A,B) // Parte imaginaria de la admitancia
:End
:End

// Potencias y tensiones iniciales
:For(A,1,N)
:If T(A)=1 or T(A)=2 // PQ y PV tienen potencia especificada
:Prompt P(A) // Potencia activa P de cada barra
:Prompt Q(A) // Potencia reactiva Q de cada barra (inicial)
:If T(A)=2
:Prompt Qmin(A) // Límite inferior de Q para barra PV
:Prompt Qmax(A) // Límite superior de Q para barra PV
:End
:End
:If T(A)=2 or T(A)=3 // PV y Slack tienen tensión especificada
:Prompt Vmag(A) // Magnitud de voltaje para PV y Slack
:End
:If T(A)=3 // Solo la barra Slack tiene ángulo especificado
:Prompt Vang(A) // Ángulo de voltaje en grados
:End
:End

// Inicialización de tensiones
:For(A,1,N)
:If T(A)=1 // Barra PQ, inicializamos tensión
:1→Vmag(A)
:0→Vang(A)
:End
:If T(A)=2 // Barra PV, ángulo inicial es 0 (será calculado)

```

```

:0→Vang(A)
:End
:If T(A)=3 // Barra Slack, ya tiene magnitud y ángulo inicializados
:Vmag(A)→Vr(A) // Parte real
:0→Vi(A) // Parte imaginaria
:End
:End

// Iteraciones Newton-Raphson
:For(I,1,Imax)
:0→Err // Inicializar error

// 1. Calcular desbalance de potencia ( $\Delta P$  y  $\Delta Q$ ) y verificar límites de Q
:For(A,1,N)
:If T(A)≠3 // Omitir barra Slack
:0→ $\Delta P(A)$ 
:0→ $\Delta Q(A)$ 

:For(B,1,N)
:If B≠A
:(YR(A,B)*Vmag(B)*cos(Vang(A)-Vang(B)) - YI(A,B)*Vmag(B)*sin(Vang(A)-Vang(B))) →
 $\Delta P(A)$ 
:(YR(A,B)*Vmag(B)*sin(Vang(A)-Vang(B)) + YI(A,B)*Vmag(B)*cos(Vang(A)-Vang(B))) →
 $\Delta Q(A)$ 
:End

// Verificar límites de potencia reactiva para barra PV
:If T(A)=2
:If Q(A) < Qmin(A)
:Qmin(A)→Q(A) // Fijar en el límite inferior
:1→T(A) // Convertir a barra PQ
:End
:If Q(A) > Qmax(A)
:Qmax(A)→Q(A) // Fijar en el límite superior
:1→T(A) // Convertir a barra PQ
:End
:End
:End

// 2. Construir la matriz Jacobiana J
:For(A,1,N)
:If T(A)≠3 // Omitir barra Slack

// Cálculo de J11 ( $dP/d\theta$ ) y J12 ( $dP/dV$ )
:For(B,1,N)
:If A≠B
:-(Vmag(A) * Vmag(B)) * (YR(A,B)*sin(Vang(A)-Vang(B)) -
YI(A,B)*cos(Vang(A)-Vang(B)))→J11(A,B)

```

```

:(Vmag(A) * (YR(A,B)*cos(Vang(A)-Vang(B)) + YI(A,B)*sin(Vang(A)-Vang(B))))→J12(A,B)
:Else
:-(Q(A) + Vmag(A)^2 * YI(A,A))→J11(A,A) // J11(A,A) (diagonal)
:Sum(B, 1, N, Vmag(B)*(YR(A,B)*cos(Vang(A)-Vang(B)) +
YI(A,B)*sin(Vang(A)-Vang(B))))→J12(A,A) // J12(A,A) (diagonal)
:End
:End

// Cálculo de J21 (dQ/dθ) y J22 (dQ/dV)
:For(B, 1, N)
:If A≠B
:-(Vmag(A) * Vmag(B)) * (YR(A,B)*cos(Vang(A)-Vang(B)) +
YI(A,B)*sin(Vang(A)-Vang(B)))→J21(A,B)
:(Vmag(A) * (YR(A,B)*sin(Vang(A)-Vang(B)) - YI(A,B)*cos(Vang(A)-Vang(B))))→J22(A,B)
:Else
:(P(A) - Vmag(A)^2 * YR(A,A))→J21(A,A) // J21(A,A) (diagonal)
:Sum(B, 1, N, Vmag(B)*(YR(A,B)*sin(Vang(A)-Vang(B)) -
YI(A,B)*cos(Vang(A)-Vang(B))))→J22(A,A) // J22(A,A) (diagonal)
:End
:End
:End

// 3. Resolver el sistema de ecuaciones ΔV y Δθ usando la matriz Jacobiana
// Implementa la resolución del sistema Jacobiano J11, J12, J21, J22 aquí

// 4. Actualizar tensiones V y ángulos θ en función de ΔV y Δθ

// 5. Verificar convergencia
:If Err<Tol
:Goto FIN
:End
:End

:Lbl FIN
:Disp "CONVERGIDO"
:Disp "Voltajes en barras"
:For(A, 1, N)
:Disp "Barra ", A, ":", Vmag(A), "∠", Vang(A)
:End

```