

```

function Proyecto_Final
tic
for repeticion = 1:9
    nombreArchivo = ['Proyecto_Final_' num2str(repeticion) '.mat']
    EV1 = zeros(21,18);
    EV2 = EV1;
    ED1 = EV1;
    ED2 = EV1;
    for i = 1:21
        NA = 45+5*i;
        for j = 1:18
            lambda2 = 45 + 3*j;
            [EV1(i,j), ED1(i,j)] = VozDatos(0.56, 0.83, NA, 10, lambda2, 625, 104.17, 600,1); % Compartiendo recursos
            [EV2(i,j), ED2(i,j)] = VozDatos(0.56, 0.83, NA, 10, lambda2, 625, 104.17, 600,2); % Sin compartir recursos
            save(nombreArchivo,'EV1','EV2','ED1','ED2');
            disp([repeticion i j toc/60])
        end
    end
end

function [EV, ED] = VozDatos(alpha, beta, N, V, lambda2, mu1, mu2, tiempoSimulacion,compartir)
    reloj = 0; % Reloj de simulación
    Estado = 0; % Número de abonados activos
    NumeroPaquetesVoz = 0;
    NumeroPaquetesDatos = 0;
    ColaVoz = 0;
    ColaDatos = 0;
    tasa = Estado*beta + (N-Estado)*alpha; % Tasa de cambio de estado (en este estado)
    ProximoCambioDeEstado = random('exp',1/tasa); % Duración en el estado actual
    ProximaLlegadaDePaqueteVoz = 1/(Estado*V); % Cada abonado activo produce V paquetes/segundo
    ProximaLlegadaDePaqueteDatos = random('exp',1/lambda2); %
    ProximaSalida = [inf inf];

    NumEventos = 0;
    traza = zeros(10*ceil(tiempoSimulacion*(N*V + lambda2)),6);
    while reloj<tiempoSimulacion % Mientras dure la simulación
        NumEventos = NumEventos+1; traza(NumEventos,:) = [reloj Estado NumeroPaquetesVoz ColaVoz NumeroPaquetesDatos ColaDatos];
        [reloj, evento] = min([ProximoCambioDeEstado ProximaLlegadaDePaqueteVoz ProximaLlegadaDePaqueteDatos abs(ProximaSalida)]);
        NumEventos = NumEventos+1;
        traza(NumEventos,:) = [reloj Estado NumeroPaquetesVoz ColaVoz NumeroPaquetesDatos ColaDatos];
        switch evento % Actualiza el reloj y selecciona el próximo evento
            case 1 % Cambia el estado de actividad de los abonados
                % Procesa el cambio del estado de actividad de los abonados
            case 2 % Llega un paquete de voz
                % Procesa la llegada de un paquete de voz
            case 3 % Llega un paquete de datos
                % Procesa la llegada de un paquete de datos
            case 4 % Se desocupa un canal de voz
                % Procesa la liberación de un canal de voz
            case 5 % Se desocupa un canal de datos
                % Procesa la liberación de un canal de datos
        end
    end
    traza = traza(1:NumEventos,:);
    dt = traza(2:end,1) - traza(1:end-1,1); % tiempo entre eventos

```

```

areas = traza(1:end-1,3).*dt;           % Areas de los rectángulos
EV = sum(areas)/traza(end,1);
areas = traza(1:end-1,5).*dt;           % Areas de los rectángulos
ED = sum(areas)/traza(end,1);

```

```

function Analisis
load Proyecto_Final_1.mat
EDff = ED2/9;
EDfm = ED1/9;
EVff = EV2/9;
EVfm = EV1/9;
for i=2:9
    nombreArchivo = ['Proyecto_Final_' num2str(i) '.mat']
    load(nombreArchivo);
    EDff = EDff + ED2/9;
    EDfm = EDfm + ED1/9;
    EVff = EVff + EV2/9;
    EVfm = EVfm + EV1/9;
end
[lambda2,na]=meshgrid(48:3:99,50:5:150)
figure(1)
mesh(na,lambda2,EDff)
xlabel('Número de abonados de voz')
ylabel('flujo de paquetes de datos')
zlabel('Número de paquetes')
title('Número promedio de paquetes de datos, frontera fija')
axis([49 151 47 100 0 15])
figure(2)
mesh(na,lambda2,EDfm)
xlabel('Número de abonados de voz')
ylabel('flujo de paquetes de datos')
zlabel('Número de paquetes')
title('Número promedio de paquetes de datos, frontera móvil')
axis([49 151 47 100 0 15])
figure(3)
mesh(na,lambda2,EVff)
xlabel('Número de abonados de voz')
ylabel('flujo de paquetes de datos')
zlabel('Número de paquetes')
title('Número promedio de paquetes de voz, frontera fija')
axis([49 151 47 100 0 35])
figure(4)
mesh(na,lambda2,EVfm)
xlabel('Número de abonados de voz')
ylabel('flujo de paquetes de datos')
zlabel('Número de paquetes')
title('Número promedio de paquetes de voz, frontera móvil')
axis([49 151 47 100 0 35])

```