

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

**IMPLEMENTACIÓN Y EVALUACIÓN DE UNA RED INALÁMBRICA
DE SENSORES A TRAVÉS DE ALGORITMOS DE
TRANSMISIÓN BASADOS EN GOSSIPING POR RF**

ANEXO 1

ANEXOS REFERENCIADOS EN LA TESIS

ANEXO A:

TABLA A-1. Bandas ISM definidas por el ITU [i]

Rango de Frecuencia [Hz]	Frecuencia Central [Hz]	Disponibilidad
6.765–6.795 MHz	6.780 MHz	Sujeto a normas locales.
13.553–13.567 MHz	13.560 MHz	
26.957–27.283 MHz	27.120 MHz	
40.66–40.70 MHz	40.68 MHz	
433.05–434.70 MHz	433.92 MHz	Solamente Región 1
902–928 MHz	915 MHz	Solamente Región 2.
2.400–2.500 GHz	2.450 GHz	
5.725–5.875 GHz	5.800 GHz	
24–24.25 GHz	24.125 GHz	
61–61.5 GHz	61.25 GHz	Sujeto a normas locales.
122–123 GHz	122.5 GHz	Sujeto a normas locales.
244–246 GHz	245 GHz	Sujeto a normas locales.

[i] CORDERO, Santiago y GUZMAN, Edison. “Diseño y construcción para el monitoreo inalámbrico de tarjetas de adquisición de datos aplicado a electrocardiografía”. 2010

ANEXO B:

Se tiene una breve descripción de cada nivel [4]:

- Nivel físico: Se encarga de definir el medio de comunicación empleado en la transferencia de datos. Además, especifica bits de control y controla el medio.

- Nivel enlace de datos: Facilita la transmisión de paquetes de datos; en otras palabras, da un formato a nivel físico de los valores lógicos en la información. Sincronía, detección de errores, son algunas de las tareas en esta capa.
- Nivel de red: Define el enrutamiento de los datos entre redes; por lo tanto, es donde los algoritmos de ruteo trabajan. Las tareas que lleva a cabo son mayormente la gestión de conexiones, envío de mensajes al nivel anterior o al siguiente.
- Nivel de transporte: Es el nivel intermedio entre los anteriores orientados a comunicaciones y los tres siguientes que se dedican al procesamiento. Dirige los dispositivos de red mediante la asignación de direcciones, soporta multicanalización, gestiona la habilitación de conexiones y determina el protocolo.
- Nivel sesión: Ofrece los servicios de organización y sincronización de diálogos entre usuarios y el manejo del intercambio de datos.
- Nivel presentación: Define el formato y asigna la sintaxis de los datos para su envío a través de la red, opera el intercambio.
- Nivel aplicación: Es el nivel donde priman los programas de aplicación, a fin de interactuar directamente con el usuario para informar del manejo de red, protocolos, etc.

ANEXO C:

TABLA A-2 Tipos de sensores utilizados en redes inalámbricas [5].

Tipos de sensores para medición en redes inalámbricas		
	Medida	Tipo de transductor
Propiedades físicas	Presión	Piezo-resistivo, capacitivo
	Temperatura	Termistor, termo-mecánico, termocupla
	Humedad	Resistivo, capacitivo
	Caudal	De cambio de presión, termistor
Propiedades de movimiento	Posición	GPS, sensor de contacto, E-mag
	Velocidad	Efecto Doppler, efecto Hall, optoelectrónico
	Velocidad angular	Encoder óptico
	Aceleración	Piezo-resistivo, piezoeléctrico, fibra óptica
Propiedades de contacto	Tensión	Piezo-resistivo
	Fuerza	Piezoeléctrico, piezo-resistivo
	Torque	Piezo-resistivo, optoelectrónico
	Deslizamiento	Torque dual
	Vibración	Piezo-resistivo, piezoeléctrico, fibra óptica, sonido, ultrasonido
Presencia	Táctil o de contacto	Switch de contacto, capacitivo
	Proximidad	De efecto Hall, capacitivo, magnético, sísmico, acústico, RF
	Distancia o rango	E-mag, magnética, de túnel
	Moción	E-mag, IR, acústica, sísmica
Bioquímica	Agentes bioquímicos	Transducción bioquímica
Identificación	Características personales	Visión
	Identificación personal	Huella digital, escaneo retinal, voz, calor y de análisis de movimiento visual

ANEXO D:

Aplicaciones del Smart Dust [15]:

- Redes de sensado de defensa: vigilancia en el campo de batalla, monitoreo de transporte, caza de objetivos, etc.

- Teclado virtual: Uso de los acelerómetros presentes en cada mote, los cuales van a estar pegados a las puntas de los dedos para que cada uno de estos realicen el movimiento deseado de tipeo de teclado.
- Control de inventario: Si uno desea monitorear las cajas de los productos que comercializa, bastará con elaborar una red de sensado donde cada mote controla el contenido de cada caja.
- Espacios de oficina inteligentes: Sensado de temperatura, humedad, entre otros conectados a nuestra ropa y todos ellos comunicados con el ambiente exterior para mejorar nuestra comodidad.

ANEXO E:

Tipos de redes [13]:

- Wireless PAN - Wireless Personal Area Network: Es una red que involucra pocos dispositivos como las redes ad-hoc. Ej: uso de dispositivos inalámbricos por medio de RF o Bluetooth.
- Wireless LAN – Local Area Network: Interconecta a varios dispositivos a través de redes Wi-Fi (IEEE 802.11.X) a comparación de las costosas redes 3G. En Ethernet la velocidad típica puede llegar a ser de 1 Gbps. Puede englobar cientos de usuarios.
- Wireless MAN – Metropolitan Area Network: Se refiere a las redes que interconectan redes LAN aledañas a altas velocidades, para lo cual requiere uso de routers, switches, fibra óptica, etc. Un ejemplo es el estándar WiMax (IEEE 802.16.X) la cual cumple con los requerimientos de alta tasa de transferencia y, a su vez, trabaja en ambientes interiores y exteriores.
- Wireless WAN: Son las redes que cubren enormes áreas exteriores. Su velocidad varía en función a la distancia entre puntos a ser conectados y depende de puntos de acceso, gateways y bridging relays inalámbricos, encargados de conectar varios segmentos de red. Ej: Internet y el Universal Mobile Telecommunications System (UMTS), el cual trabaja en celulares.
- Redes para dispositivos móviles o celulares:
 - i) GSM – Global System for Mobile Communications: El sistema troncal de las comunicaciones por telefonía celular. Cada dispositivo se conecta al sistema que lo redirecciona a la estación de soporte y a la switching station, encargada de dirigirla a la ubicación destino.

- ii) PCS – Personal Communication Service: Banda de radiofrecuencia empleada en el sur de Asia y Norteamérica.
- iii) D-AMPS Digital Advanced Mobile Phone Service: Versión moderna del Advance Mobile Phone System
- TAN – Tiny Area Network: Parecida a la red LAN pero a escala de 2 o 3 dispositivos.
- CAN – Campus Area Network: Como la red MAN, pero con limitado ancho de banda.

ANEXO F:

TABLA F-1. Comparación de algoritmos MAC en WLAN con estructura [14]

Algorithms	Carrier Sensing , RTS / CTS, Polling	NAV (virtual sensing)	DIFS / PIFS / SIFS	Back-off Algorithm	Back-off Timer	With ACK	Centralized
<i>CSMA/CA</i>	Carrier Sensing	No	----	BEB	No	Yes	No
<i>MACA</i>	RTS/CTS	No	----	BEB	No	No	No
<i>MACAW</i>	RTS/CTS	No	----	MILD	No	Yes	No
<i>IEEE 802.11 with CSMA/CA</i>	Carrier Sensing	No	DIFS / SIFS	BEB	Yes	Yes	No
<i>IEEE 802.11 with RTS/CTS</i>	Carrier Sensing & RTS/CTS	Yes	DIFS / SIFS	BEB	Yes	Yes	No
<i>IEEE 802.11 with Polling</i>	Polling	Yes	PIFS / SIFS	-----	-----	Yes	Yes

TABLA F-2. Comparación de algoritmos MAC con antenas omnidireccionales [14]

Algorithms	Carrier Sensing	RTS/CTS	Increased RTS/CTS Transmission time	Back-off Algorithm	Busy Tone based	Split channel
<i>CSMA/CA</i>	Yes	No	No	BEB	No	No
<i>MACA / MACAW</i>	No	Yes	No	BEB / MILD	No	No
<i>FAMA</i>	Yes	Yes	Yes	BEB	No	No
<i>DBTMA</i>	No	Yes	No	MILD	Yes	Yes
<i>IEEE 802.11 MAC</i>	Yes	Yes	No	BEB	No	No

TABLA F-3. Comparación de algoritmos MAC con antenas direccionales [14]

Algorithms	Basic Protocol	Carrier Sensing	Omni directional RTS/CTS	Directional RTS	Directional Busy Tone	On-Demand Location Tracking
<i>MAC/DA1</i>	802.11	Yes	CTS	Yes	No	No
<i>MAC/DA2</i>	802.11	Yes	RTS/CTS	Yes	No	Yes
<i>DBTMA/DA</i>	DBTMA	No	RTS/CTS	No	Yes	Yes

TABLA F-4. Comparación de algoritmos MAC en WSN [14]

Algorithms	Basic Concept	Carrier Sensing	Synchronization based	Sleep / listen schedule	Cluster approach	Energy efficient	Accommodative
<i>S-MAC</i>	Synchronization management	Yes	Yes	Yes	Yes	No	Good
<i>SIFT</i>	Event driven	Yes	No	No	No	No	Good
<i>DMAC</i>	Data gathering	No	No	Yes	Yes	Yes	Weak
<i>T-MAC</i>	Listening Time Threshold	Yes	Yes	No fixed schedule	Yes	Yes	Good
<i>DS-MAC</i>	Dynamic duty cycle	Yes	Yes	Yes	Yes	Yes	Good

ANEXO G:

Para un nodo genérico u

1: Inicialización

2: $N' = \{u\}$

3: Para todo nodo v

4: Si v es vecino de u

5: Entonces $D(v) = c(u,v)$

6: Si no $D(D(v)) = f$

7:

- 8: Lazo
- 9: Encuentra w que no este en N' tal que $D(w)$ es el menor
- 10: Añade w a N'
- 11: Actualiza $D(v)$ para cada vecino v de w y que no este en N'
- 12: $D(v) = \min[D(v), D(w) + c(w,v)]$
- 13: /* El nuevo coste de v es o el antiguo coste de v o el costo del camino
- 14: más corto conocido a w más el costo de w a v */
- 15: Hasta todos los nodos en N

ANEXO H:

- 1: Inicialización
- 2: Para todos los destinos de N
- 3: $D_x(y) = c(x,y)$ /* Si y no es un vecino que $c(x,y) = f$ */
- 4: Para cada vecino w
- 5: $D_x(y) = f$, para todos los destinos y en N
- 6: Para cada vecino w
- 7: Mandar vector distancia $D_x = [D_x(y):y \text{ en } N]$ a w
- 8:
- 9: Lazo
- 10: Esperar (hasta que se vea el cambio del coste de enlace en un vecino w o
- 11: hasta que reciba un vector distancia de algún vecino w)
- 12:
- 13: Para cada y en N :
- 14: $D_x(y) = \min_v[c(x,v) , D_v(y)]$
- 15:
- 16: Si $D_x(y)$ cambia para cualquier destino y
- 17: mandar vector distancia $D_x = [D_x(y):y \text{ en } N]$ a todos
- 18: los vecinos
- 19: Para siempre

ANEXO I:

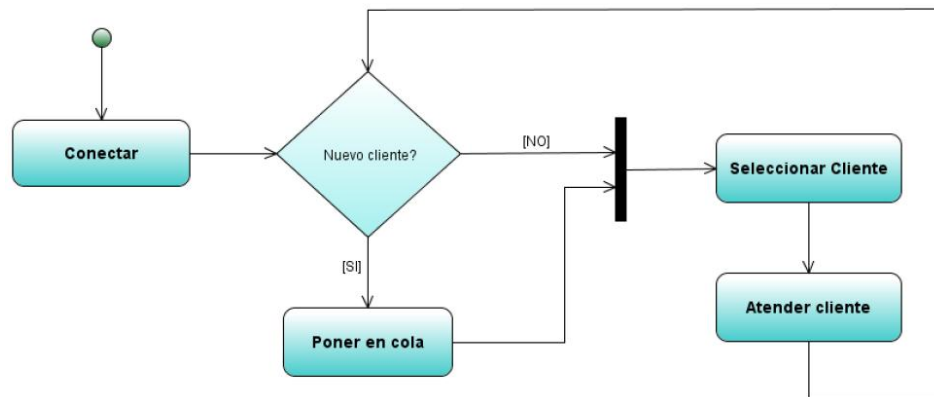


FIGURA 2-7 Diagrama de flujo del funcionamiento del algoritmo [18]

ANEXOS J:

TABLA J-1. Tabla de parámetros del flooding en topología aleatoria. Elaboración propia.

PARÁMETROS	Número de nodos (N)				
	10	20	30	40	50
FLOODING	10	20	30	40	50
BASICOS	Parámetros generales de desempeño				
Complejidad temporal	162.8	249.6	411.6	540.2	591.6
Complejidad de mensajes	325.6	499.2	823.2	1080.4	1183.2
Complejidad de conexiones	21.4	92.6	214.4	379.6	586.6
Complejidad espacial	1628	4992	12348	21608	29580
Probabilidad de fin	[97;234]	[190;300]	[311;498]	[519;567]	[504;780]
Eficacia (%)	76%	84%	83%	84%	83%
AVANZADOS	Parámetros adicionales ligados al algoritmo empleado				
Tasa de difusión	2	2	2	2	2
Total de población afectada	7.6	16.8	24.8	33.4	41.4
Distribución de grado	2.14	4.63	7.147	9.49	11.732
Cercanía(área 1 x 1)	0.2954	0.30316	0.30772	0.30878	0.31026
Intermediación	100%	100%	100%	100%	100%
DEFINICIONES PROPIAS	Parámetros definidos por el autor				

Densidad mínima (respecto a 1000 x 1000)	0.5	0.5	0.5	0.5	0.5
Eficacia propia (%)	50.33%	51.91%	58.38%	61.09%	62.77%
Rango de finalización	0.1	0.1	0.1	0.1	0.1

TABLA J-2. Tabla de parámetros de RGA en topología aleatoria. Elaboración propia.

PARÁMETROS	Número de nodos (N)				
	10	20	30	40	50
BASICOS	Parámetros generales de desempeño				
Complejidad temporal	259.2	845.2	5187.6	7556	7316.8
Complejidad de mensajes	518.4	1690.4	10375.2	15112	14633.6
Complejidad de conexiones	21.4	92.6	214.4	379.6	586.6
Complejidad espacial	2592	16904	155628	302240	365840
Probabilidad de fin	[60;429]	[645;1194]	[1189;7402]	[3544;11635]	[3290;12163]
Eficacia (%)	72%	59%	53%	60%	53%
AVANZADOS	Parámetros adicionales ligados al algoritmo empleado				
Tasa de difusión	2	2	2	2	2
Total de población afectada	7.2	11.8	15.8	24	26.4
Distribución de grado	2.14	4.63	7.147	9.49	11.732
Cercanía(área 1 x 1)	0.2954	0.30316	0.30772	0.30878	0.31026
Intermediación	100%	100%	100%	100%	100%
DEFINICIONES PROPIAS	Parámetros definidos por el autor				
Densidad mínima (respecto a 1000 x 1000)	0.5	0.5	0.5	0.5	0.5
Eficacia propia (%)	58.60%	83.05%	67.79%	52.33%	64.25%
Rango de finalización	0.1	0.1	0.1	0.1	0.1

TABLA J-3. Tabla de parámetros de DRG en topología aleatoria. Elaboración propia.

PARÁMETROS	Número de nodos (N)				
	10	20	30	40	50
BASICOS	Parámetros generales de desempeño				
Complejidad temporal	17.2	20.6	16.8	13.4	12.6
Complejidad de mensajes	149.2	371.2	443.6	478	600.8
Complejidad de conexiones	21.4	92.6	214.4	379.6	586.6
Complejidad espacial	172	412	504	536	630
Probabilidad de fin	[11;22]	[3;38]	[8;32]	[11;17]	[9;13]
Eficacia (%)	100%	100%	100%	100%	100%
AVANZADOS	Parámetros adicionales ligados al algoritmo empleado				
Tasa de difusión	5.337209302	10.0097087	14.202381	18.83582	24.84127
Total de población afectada	10	20	30	40	50
Distribución de grado	2.14	4.63	7.1466667	9.49	11.732
Cercanía(área 1 x 1)	0.2954	0.30316	0.30772	0.30878	0.31026
Intermediación	100%	100%	100%	100%	100%
DEFINICIONES PROPIAS	Parámetros definidos por el autor				
Densidad mínima (respecto a 1000 x 1000)	0.5	0.5	0.5	0.5	0.5
Eficacia propia (%)	50.33%	51.91%	58.38%	61.09%	62.77%
Rango de finalización	0.1	0.1	0.1	0.1	0.1

TABLA J-4. Tabla de parámetros del RGAM en varias topologías. Elaboración propia.

PARÁMETROS	Número de nodos (N)				
	10	20	30	40	50
RGAM	TOPOLOGÍA LINEAL				
Complejidad temporal	260	2291.8	6093.8	13792.8	24006
Complejidad de mensajes	520	4583.6	12187.6	27585.6	48012
Complejidad de conexiones	9	19	29	39	49
	TOPOLOGÍA ANILLO				
Complejidad temporal	108.6	673.8	2099.4	4886.8	8489.8
Complejidad de mensajes	217.2	1347.6	4198.8	9773.6	16979.6
Complejidad de conexiones	10	20	30	40	50
	TOPOLOGÍA GEOMÉTRICA ALEATORIA (R = 0.5)				
Complejidad temporal	80	447.8	1148.4	2401.2	3164
Complejidad de mensajes	160	895.6	2296.8	4802.4	6328
Complejidad de conexiones	21.4	92.6	214.4	379.6	586.6

ANEXO K:

TABLA K. Hoja de datos de Xbee series 2 [26].

1.2. Specifications

Table 1-01. Specifications of the XBee Series 2 OEM RF Module (PRELIMINARY)

Specification	XBee Series 2
Performance	
Indoor/Urban Range	up to 133 ft. (40 m)
Outdoor RF line-of-sight Range	up to 400 ft. (120 m)
Transmit Power Output (software selectable)	2mW (+3dBm)
RF Data Rate	250,000 bps
Serial Interface Data Rate (software selectable)	1200 - 230400 bps (non-standard baud rates also supported)
Receiver Sensitivity	-95 dBm (1% packet error rate)
Power Requirements	
Supply Voltage	2.8 – 3.4 V
Operating Current (Transmit)	40mA (@ 3.3 V)
Operating Current (Receive)	40mA (@ 3.3 V)
Power-down Current	< 1 uA @ 25°C
General	
Operating Frequency Band	ISM 2.4 GHz
Dimensions	0.960" x 1.087" (2.438cm x 2.761cm)
Operating Temperature	-40 to 85° C (industrial)
Antenna Options	Integrated Whip, Chip, RPSMA, or U.FL Connector
Networking & Security	
Supported Network Topologies	Point-to-point, Point-to-multipoint, Peer-to-peer & Mesh
Number of Channels (software selectable)	16 Direct Sequence Channels
Addressing Options	PAN ID and Addresses, Cluster IDs and Endpoints (optional)

De la tabla anterior se resalta el rango de alcance de la radio (40 mts. en lugares cerrados y 120 mts. en ambientes libres) y el consumo cuando recibe o transmite información (40 mA).

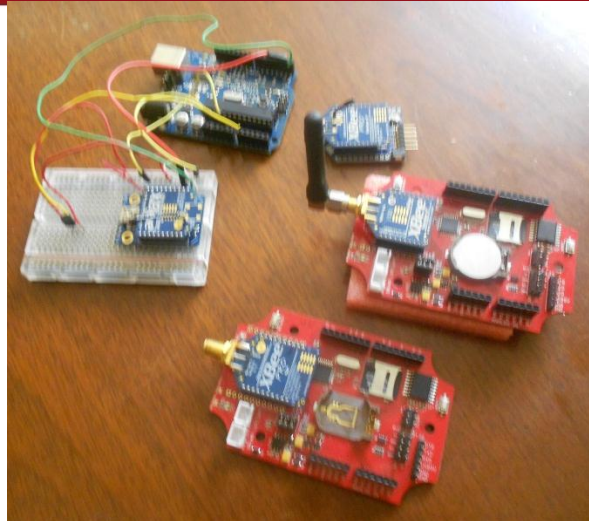


FIGURA M-1: Componentes de la red implementada. Elaboración propia.

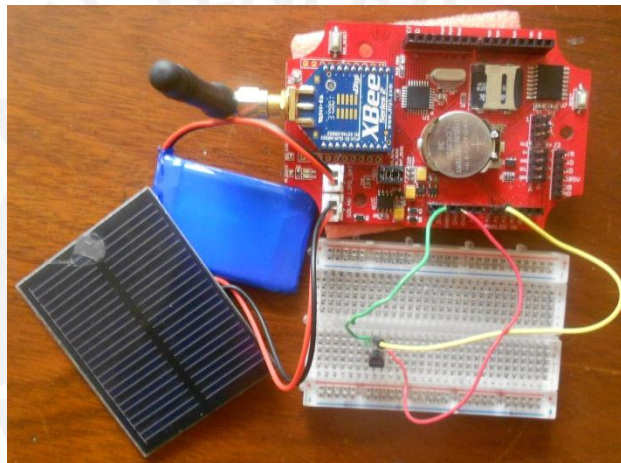


FIGURA M-2: Circuito del nodo sensor. Elaboración propia.

ANEXO N:

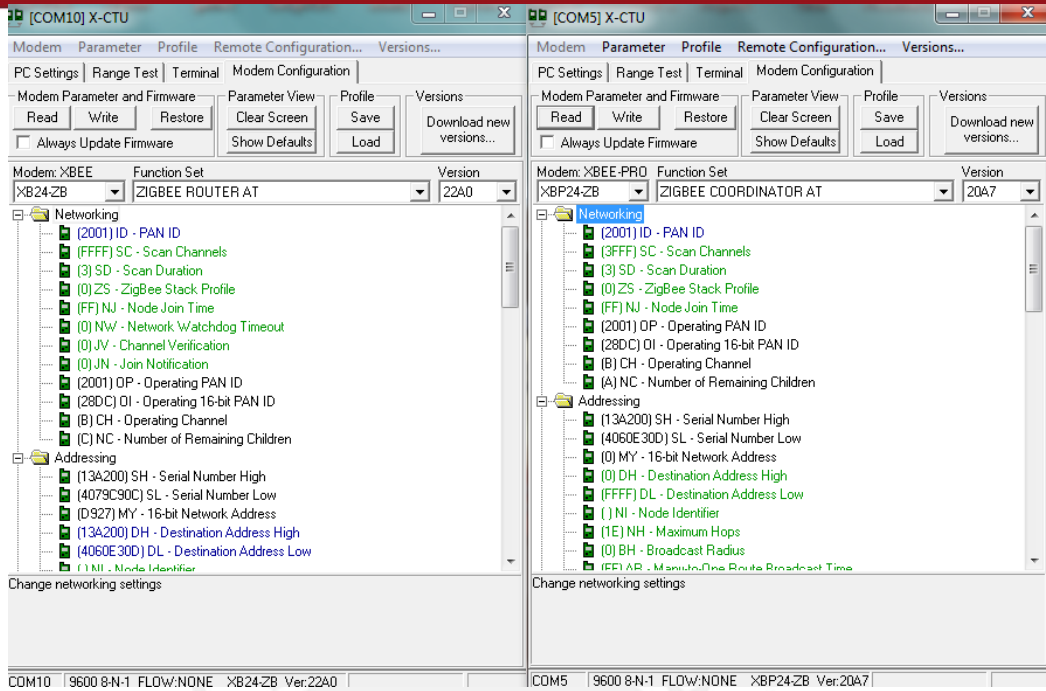


FIGURA N-1. Configuración del Xbee en modo unicast (izq.) y modo broadcast (der.)

En esta prueba se muestra que cada módulo o nodo sensor puede recibir datos de la PC y realizar el mismo la tarea de computación distributiva sin ningún inconveniente como se observa en la figura, además del hecho que no altera la transmisión del resto de los participantes de la red:

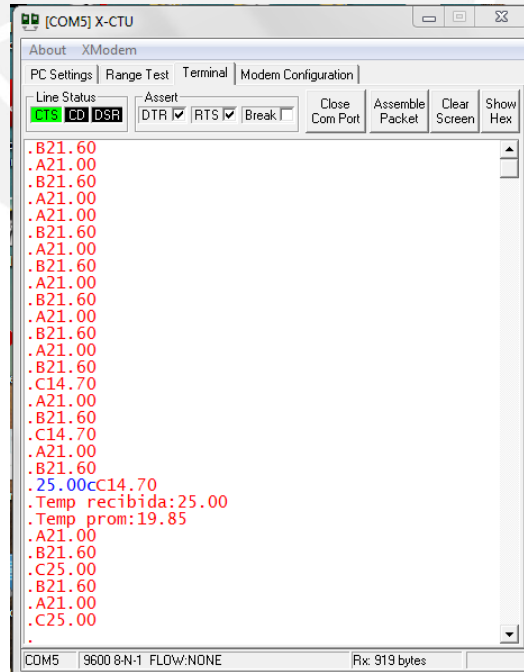


FIGURA N-2. Resultados del procesamiento remoto.

ANEXO O:

TABLA O. Resultados de medición de consumo de corriente

Tipo de módulo	Consumo por modo de trabajo		
	Stand-by	Programa	Programa + Xbee
Seeeduino Stalker (nodo A)	8.5 mA	10.1 mA	51.5 mA
Seeeduino Stalker (nodo B)	8.5 mA	10.3 mA	50.8 mA
Arduino Diecimila (nodo C)	4.3 mA	6.3 mA	46.1 mA

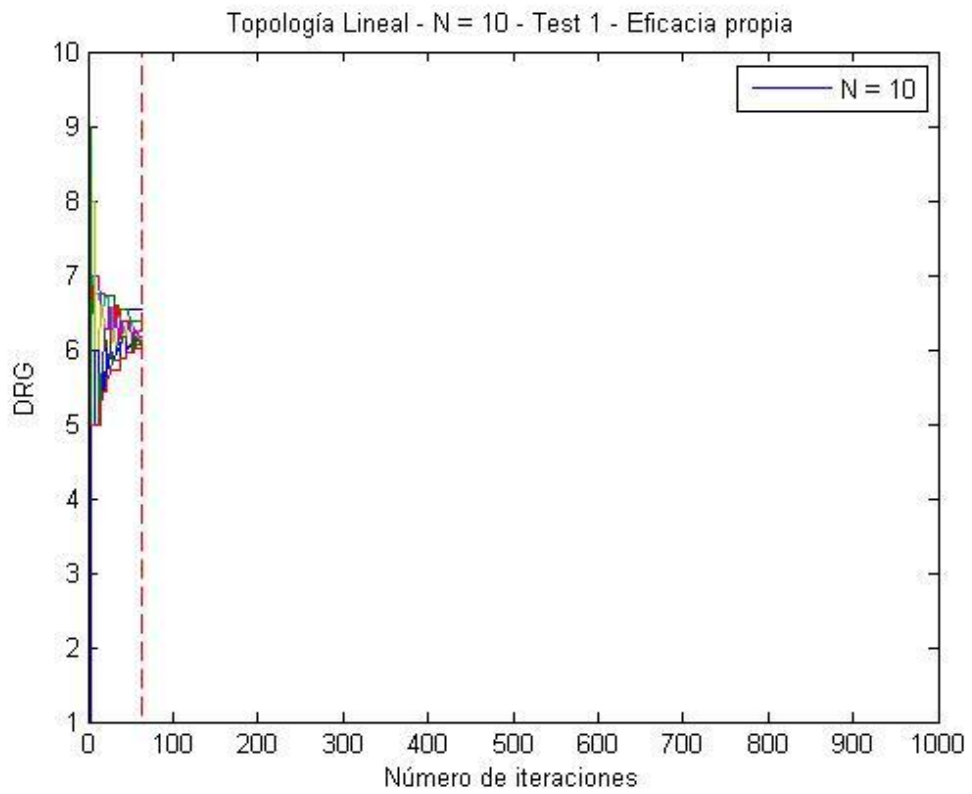
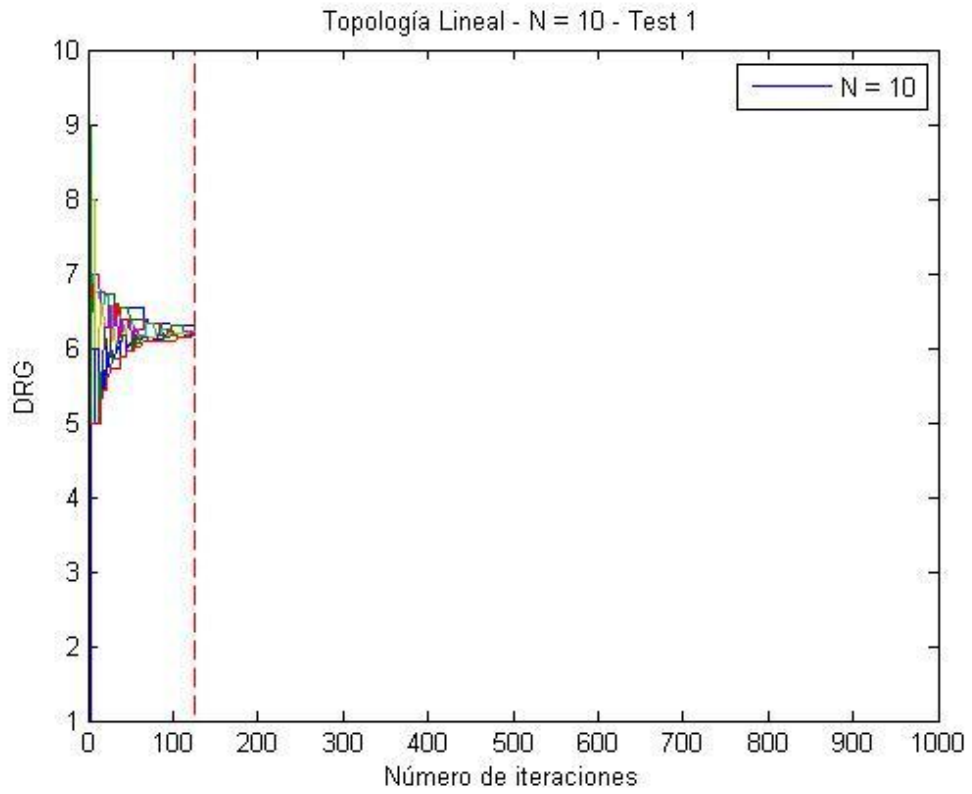
Con las 5 muestras de medición ponderadas y una vez obtenido el consumo promedio de corriente, este se multiplica por 3.3 V y se obtiene aproximadamente la potencia consumida en tiempo real.

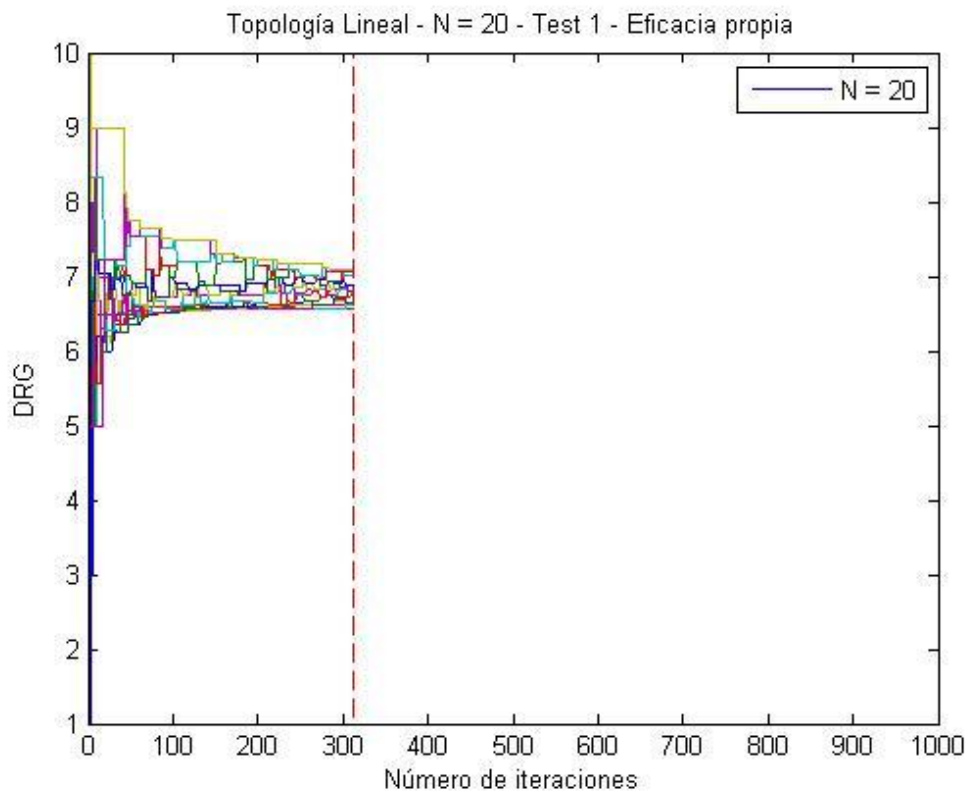
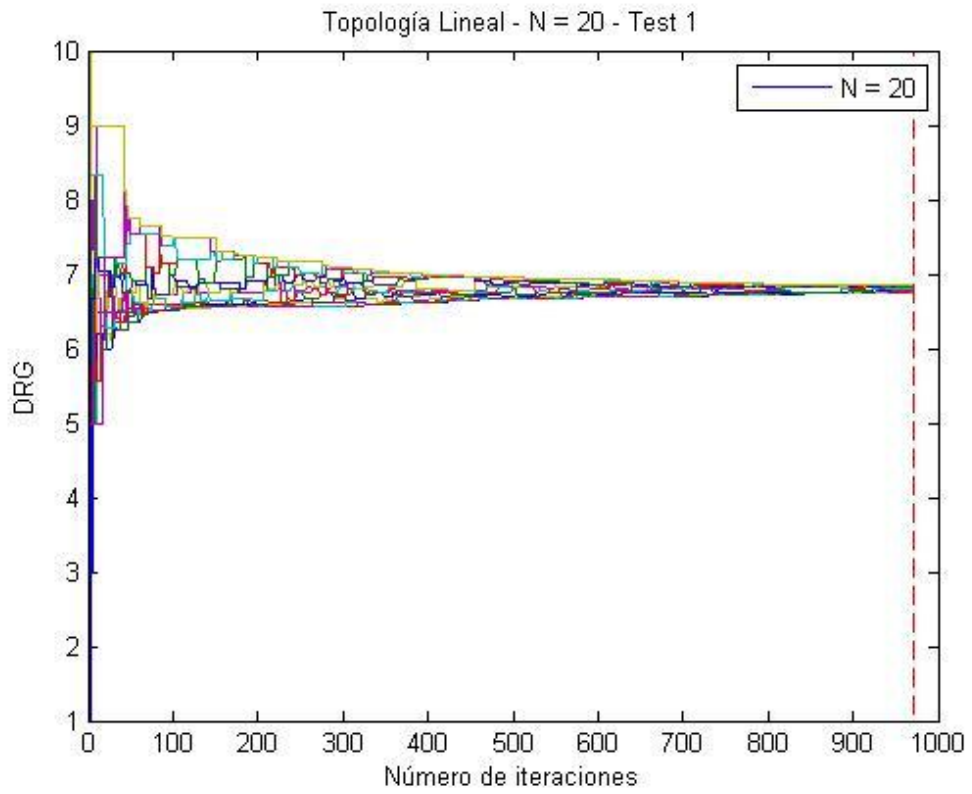
PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

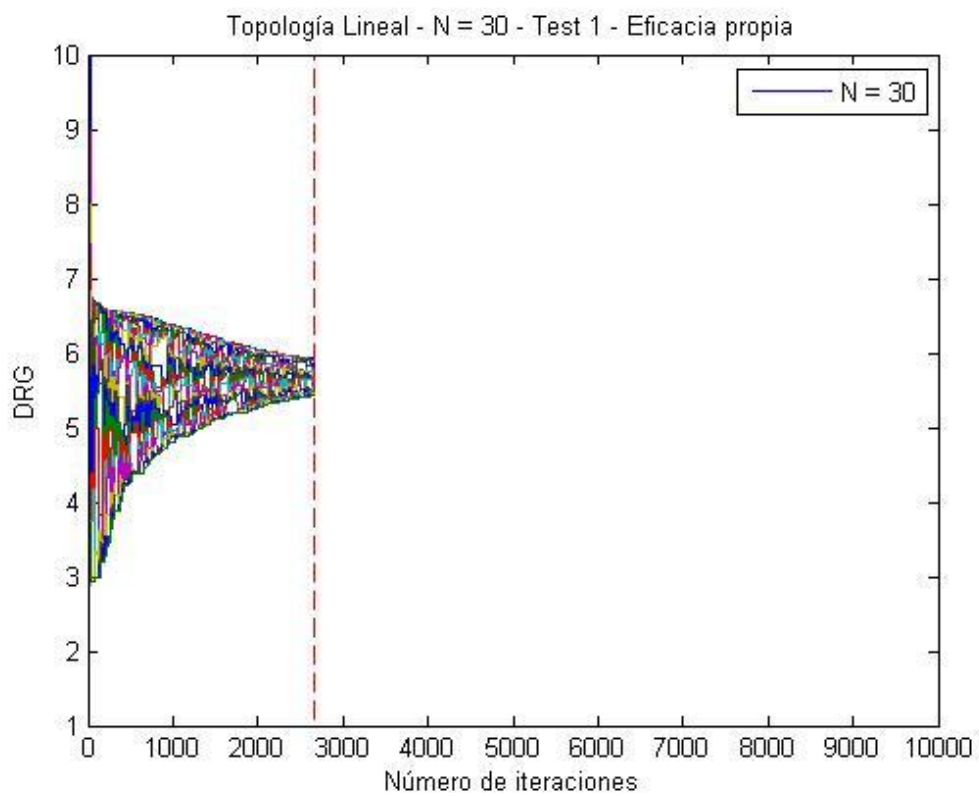
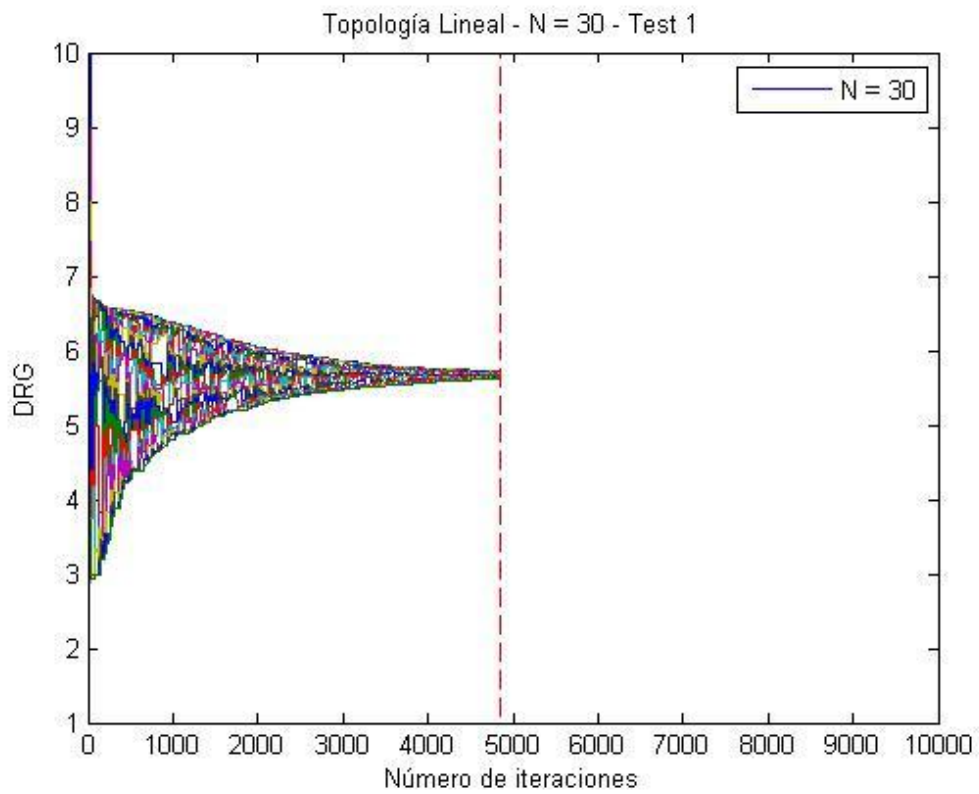
**IMPLEMENTACIÓN Y EVALUACIÓN DE UNA RED INALÁMBRICA
DE SENSORES A TRAVÉS DE ALGORITMOS DE
TRANSMISIÓN BASADOS EN GOSSIPING POR RF**

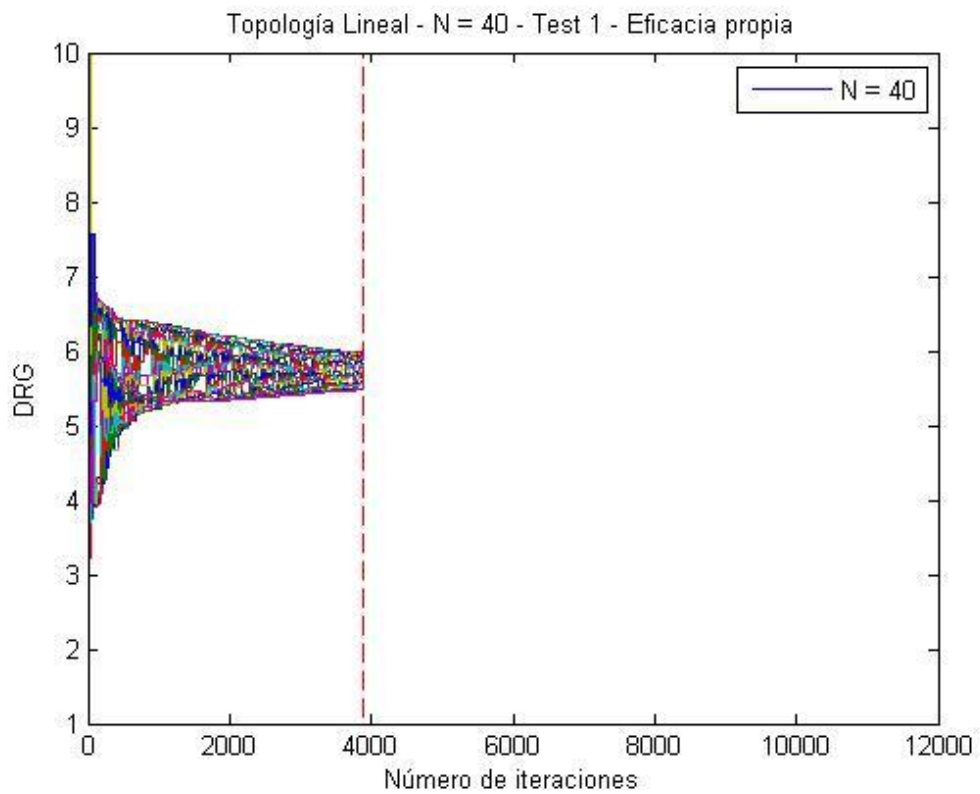
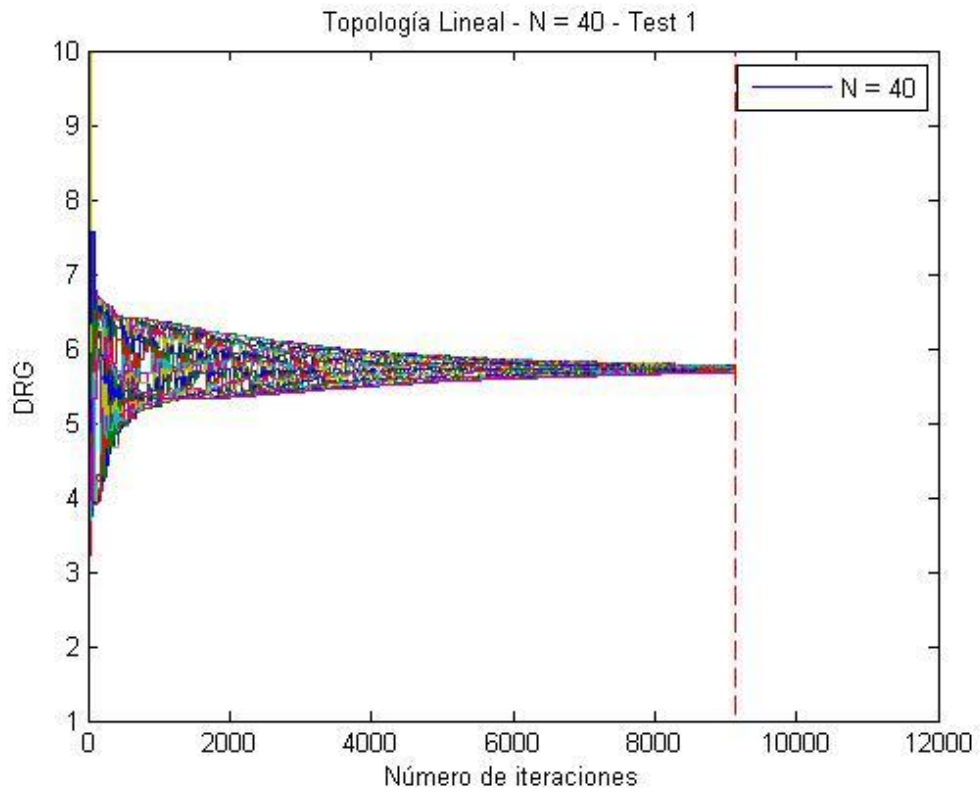
ANEXO 2

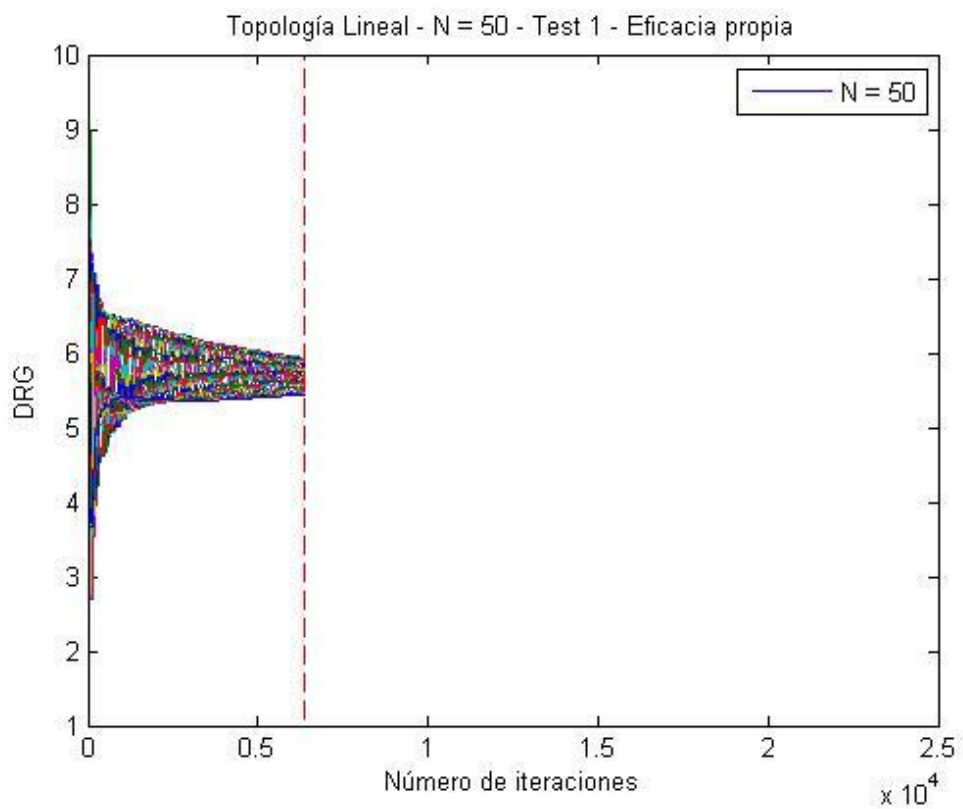
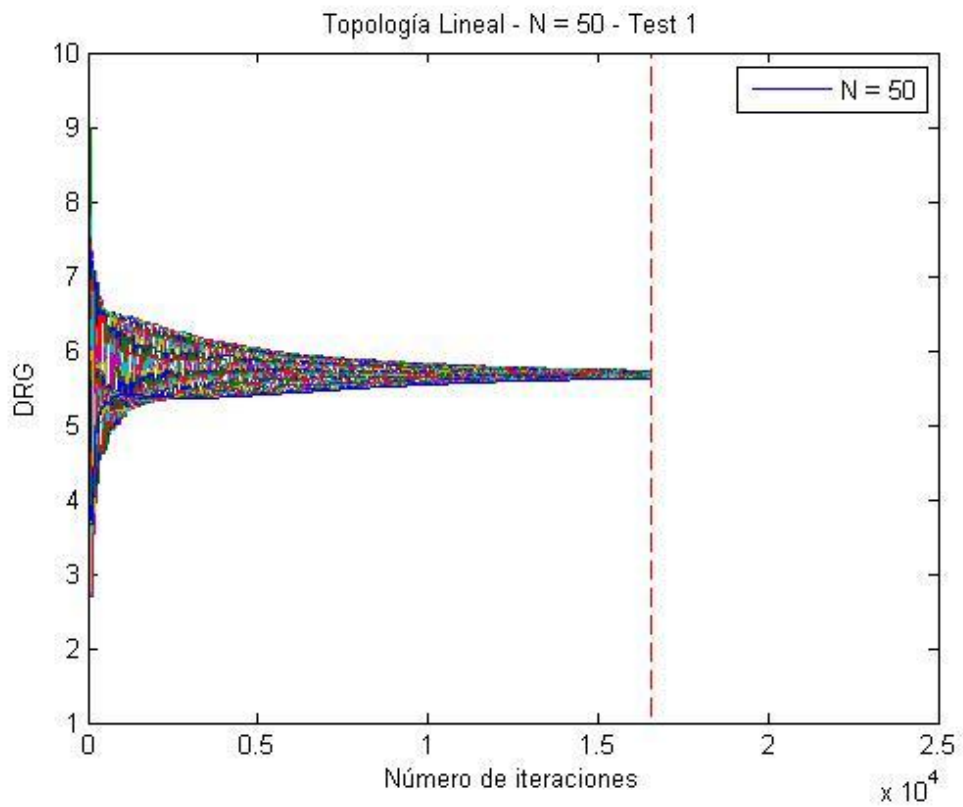
RESULTADOS DE SIMULACIÓN DEL DRG

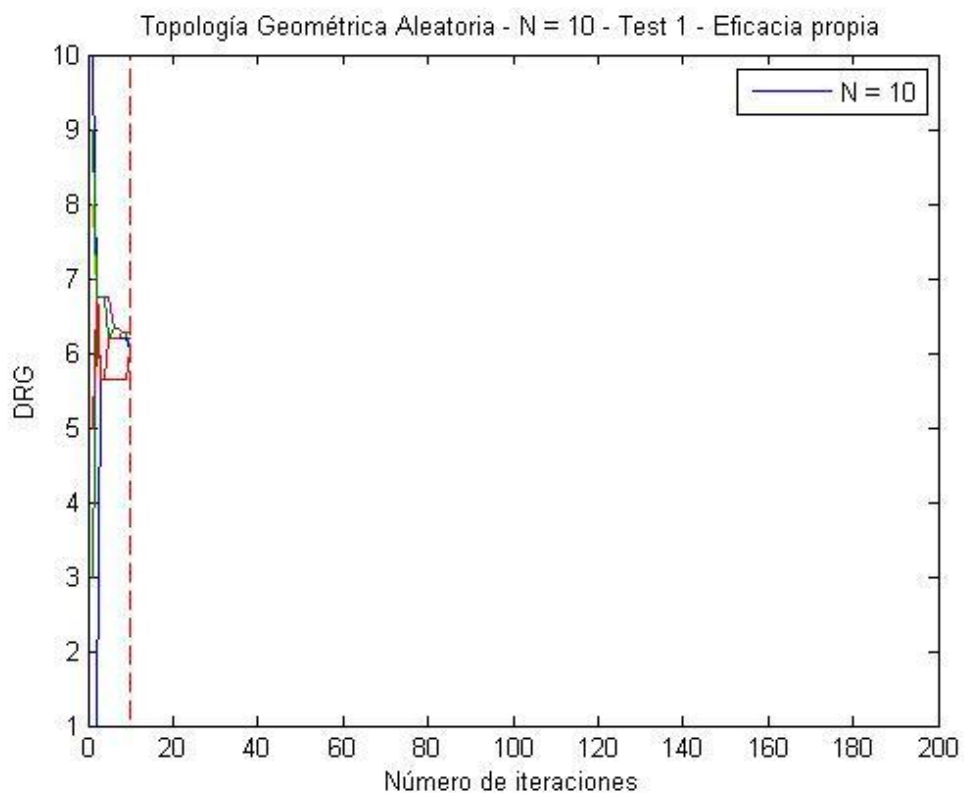
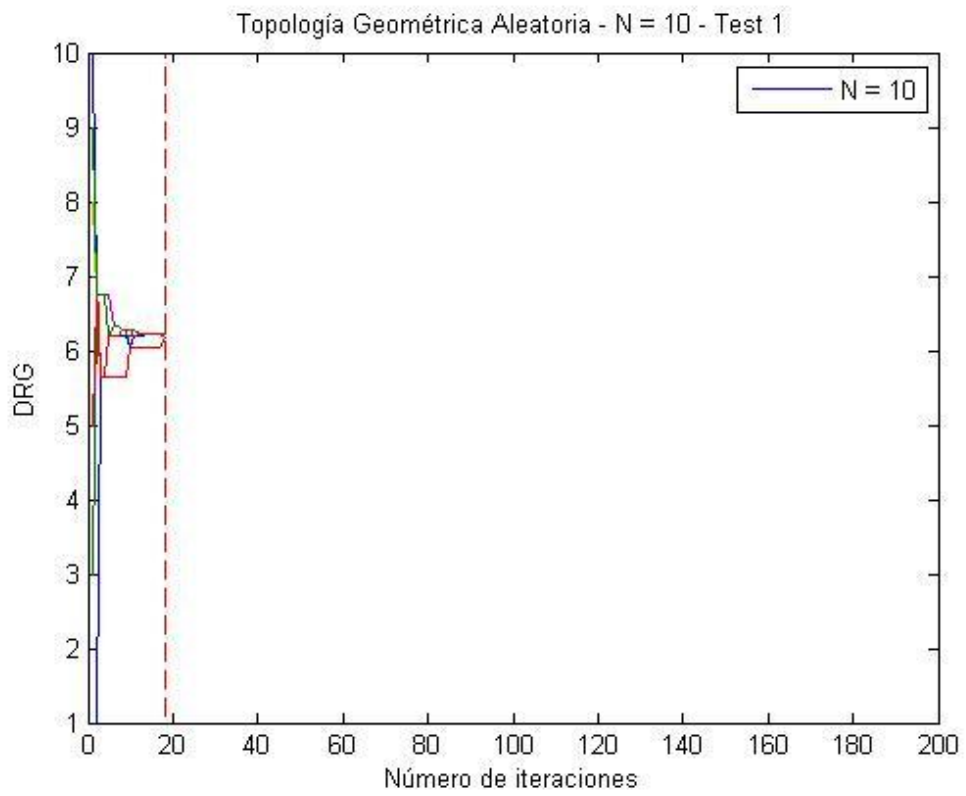


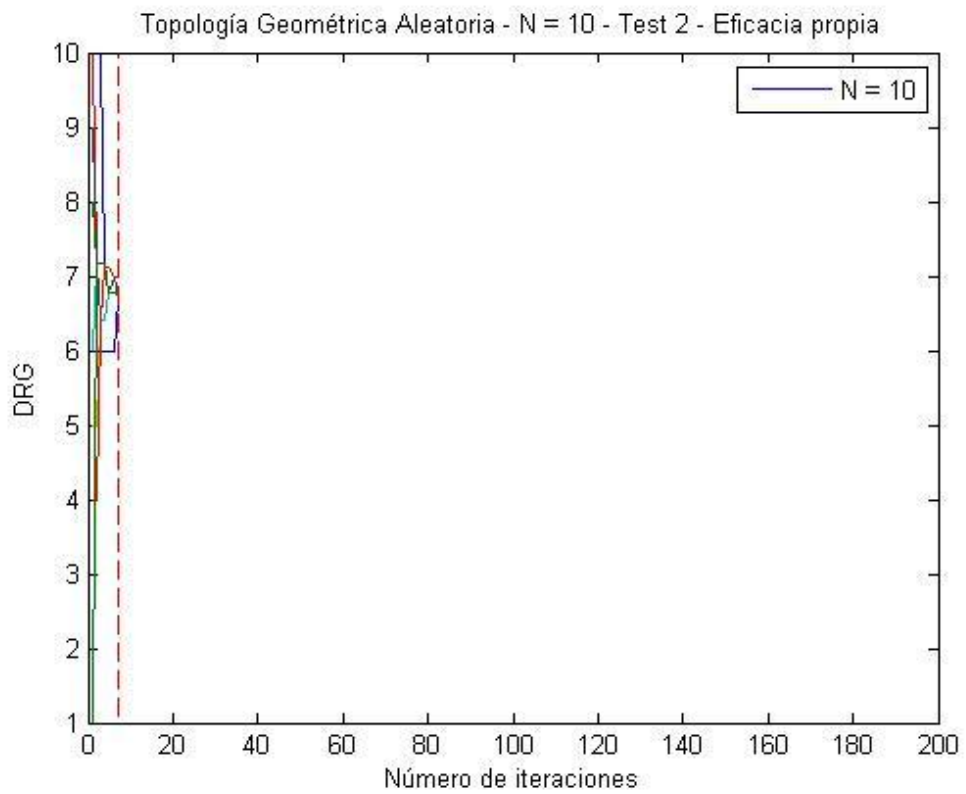
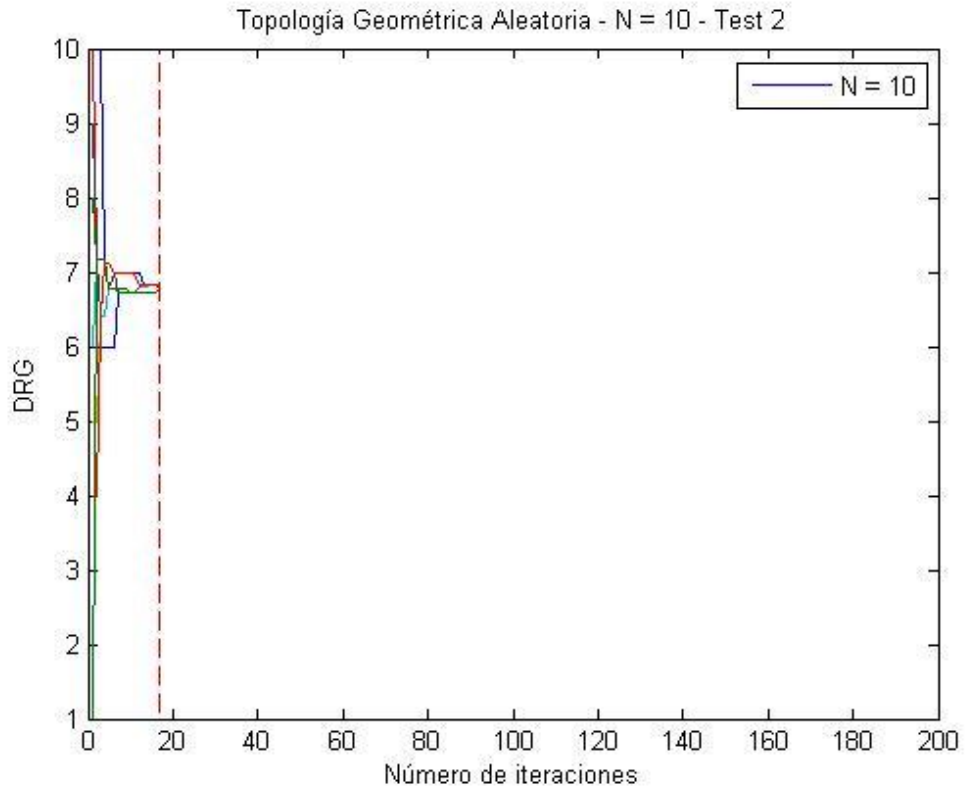


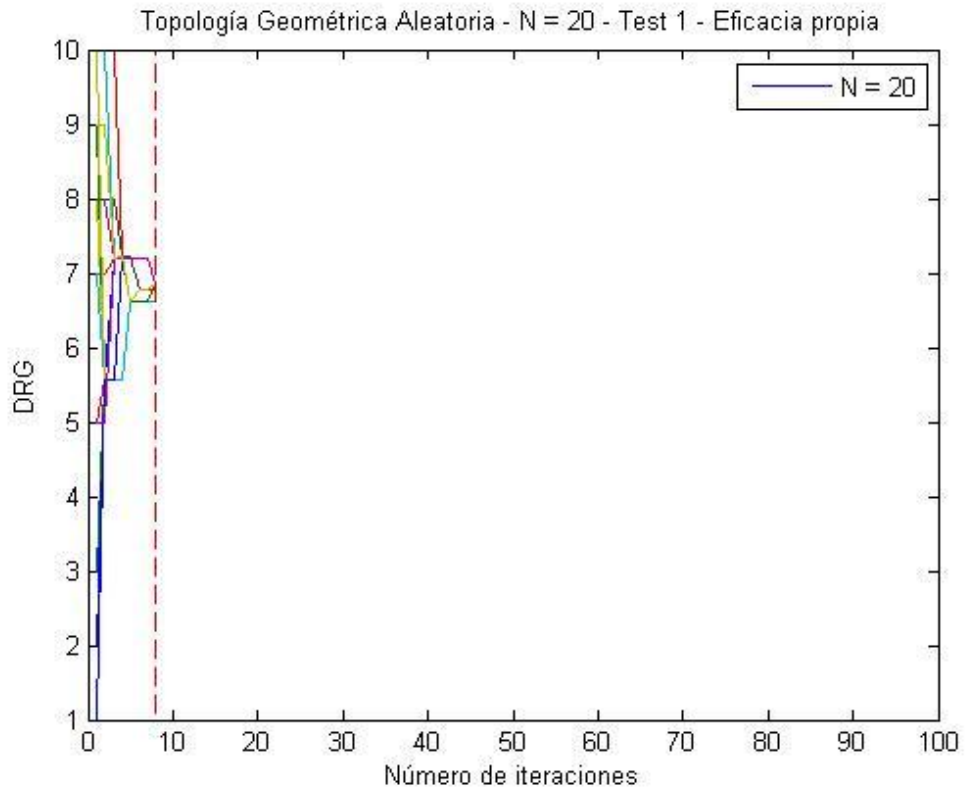
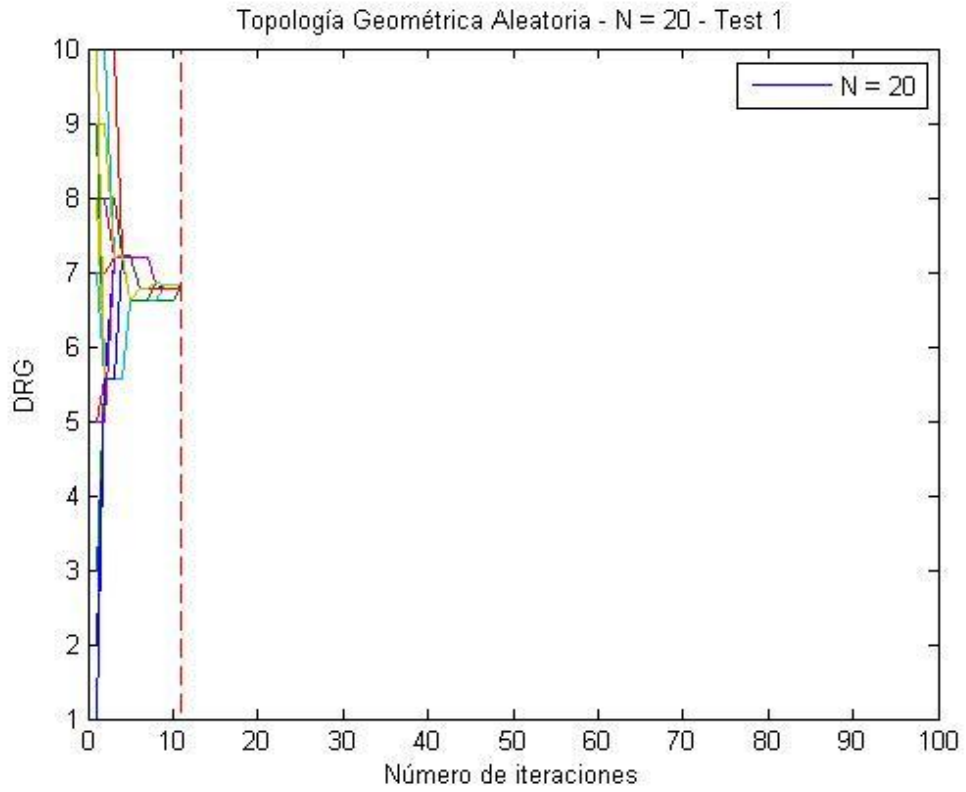


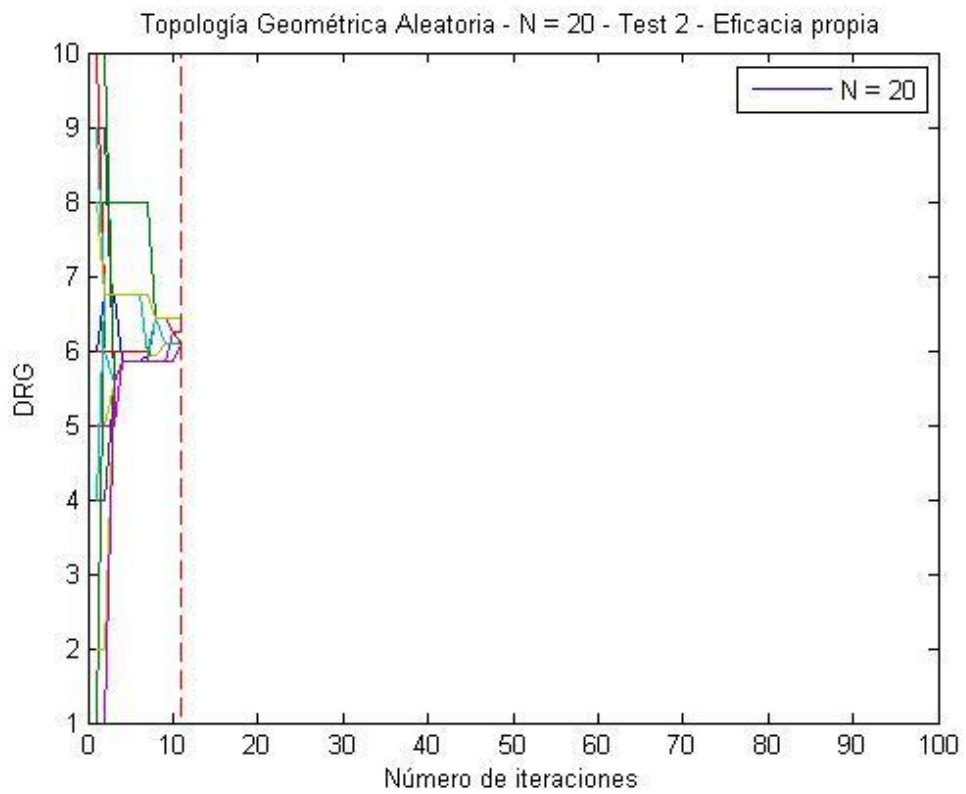
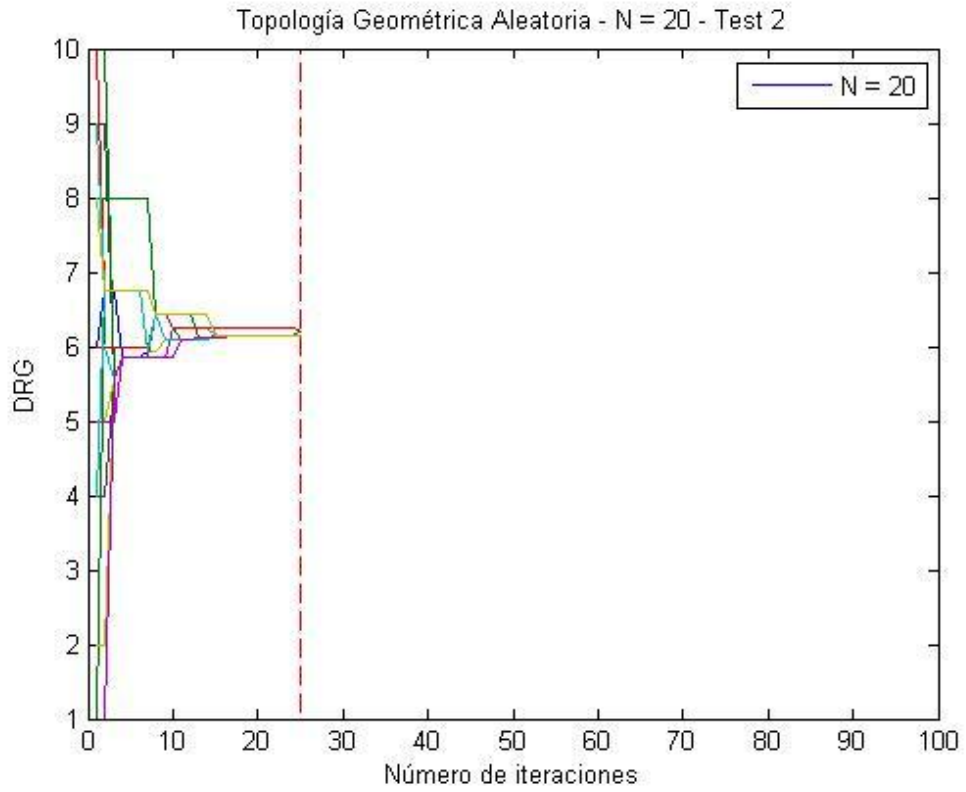


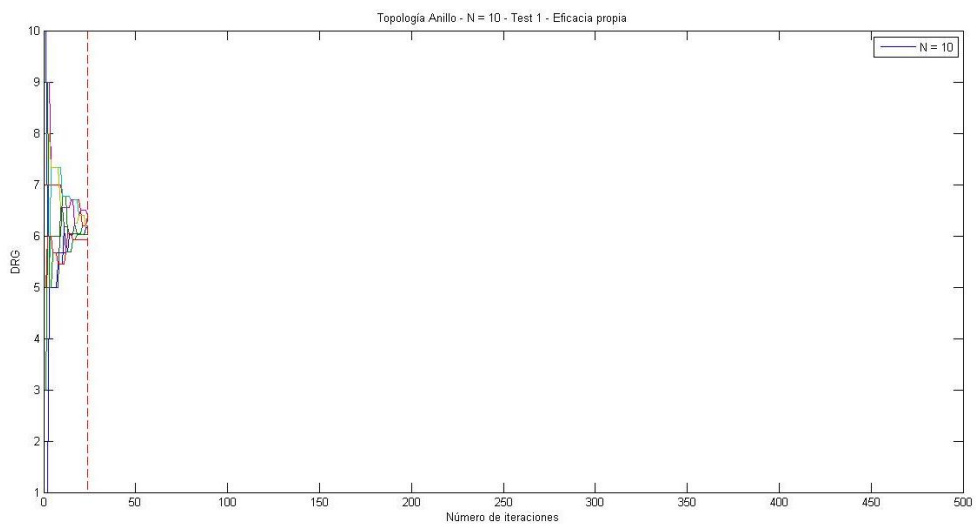
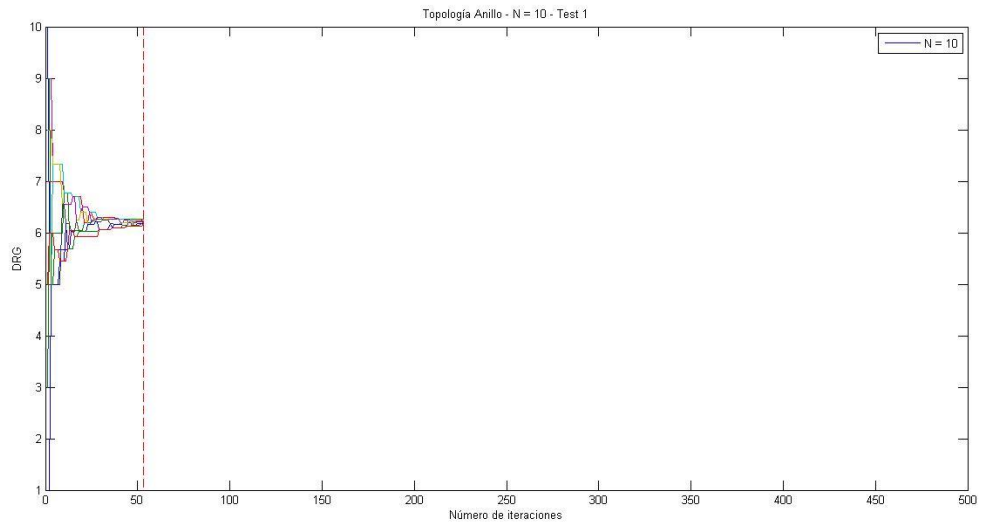


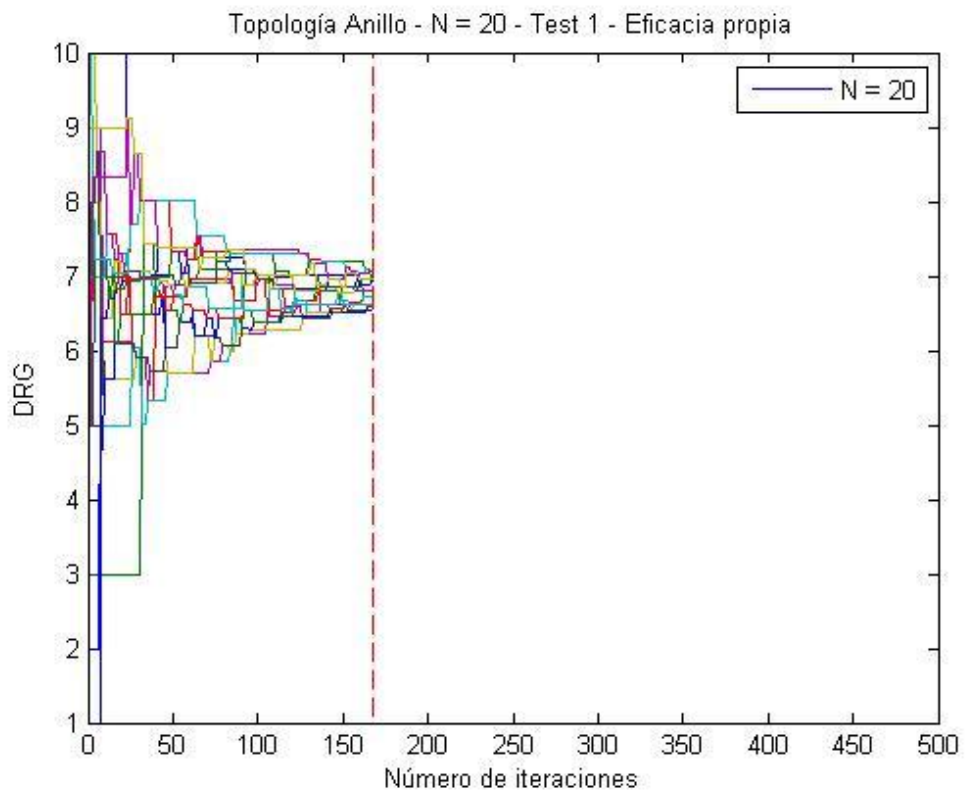
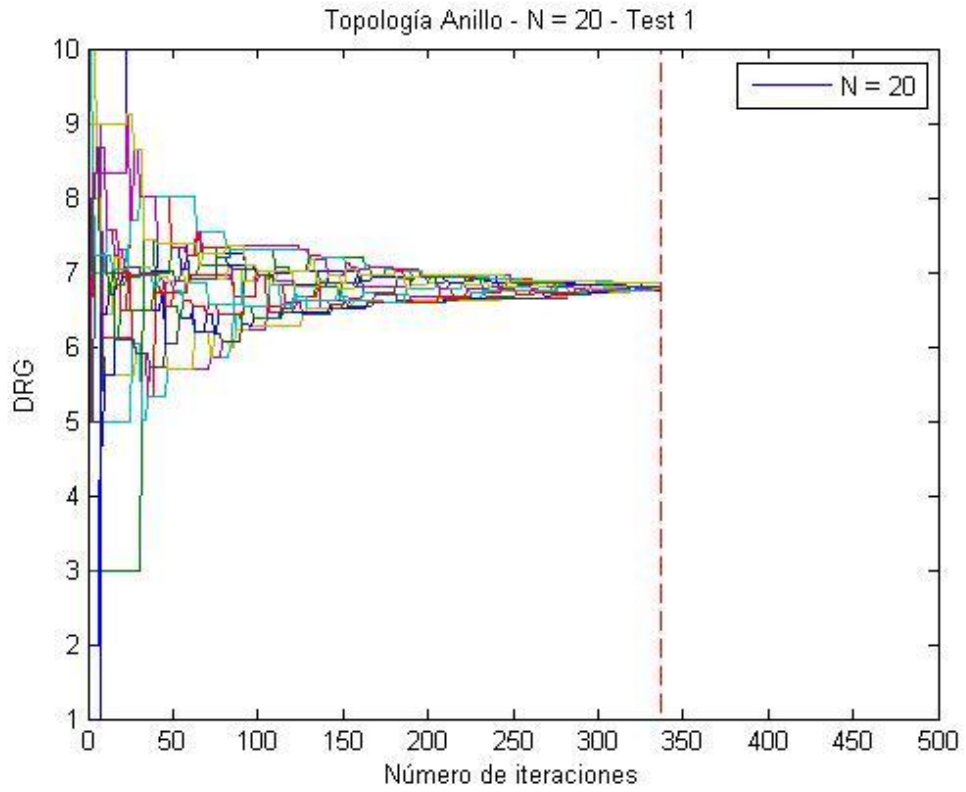


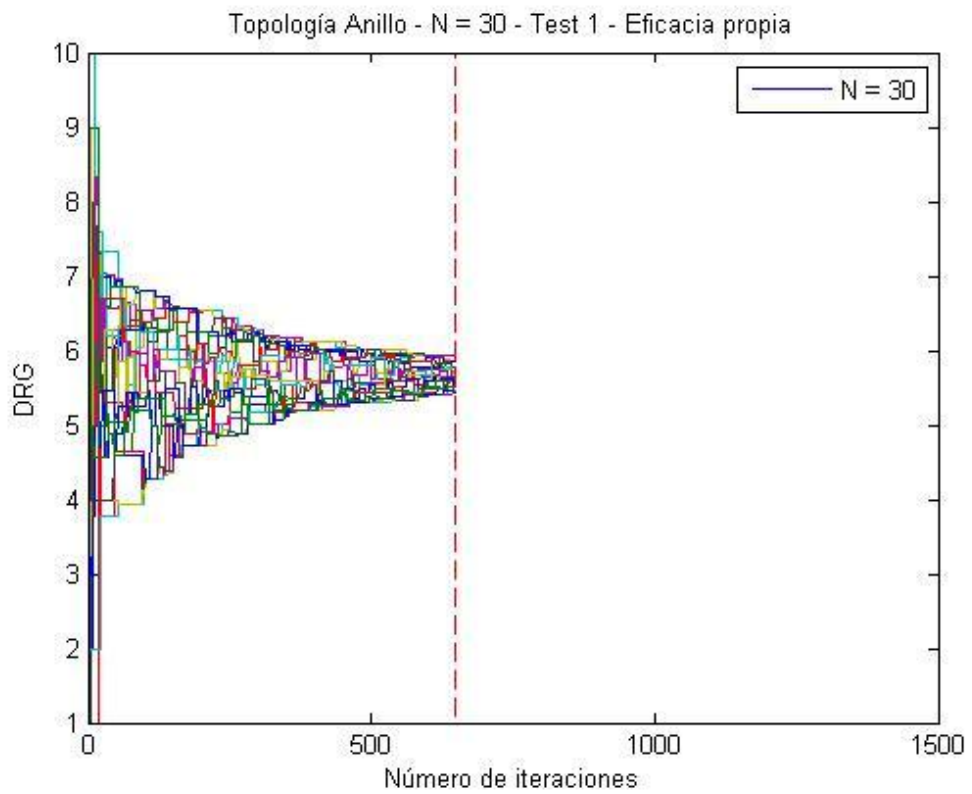
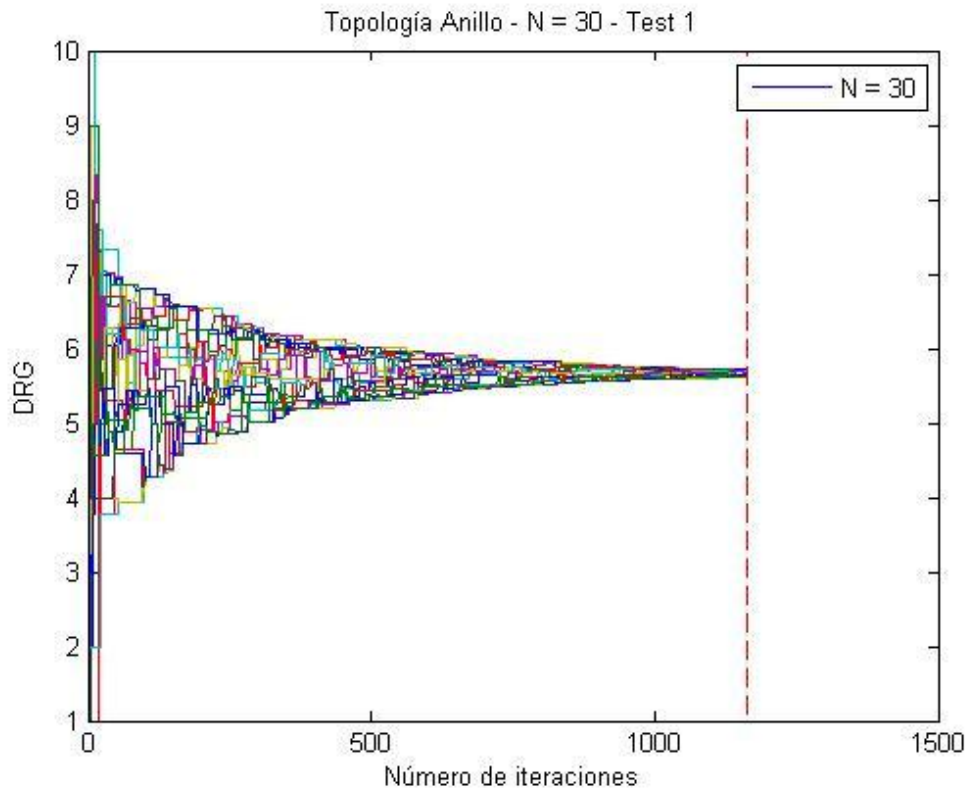


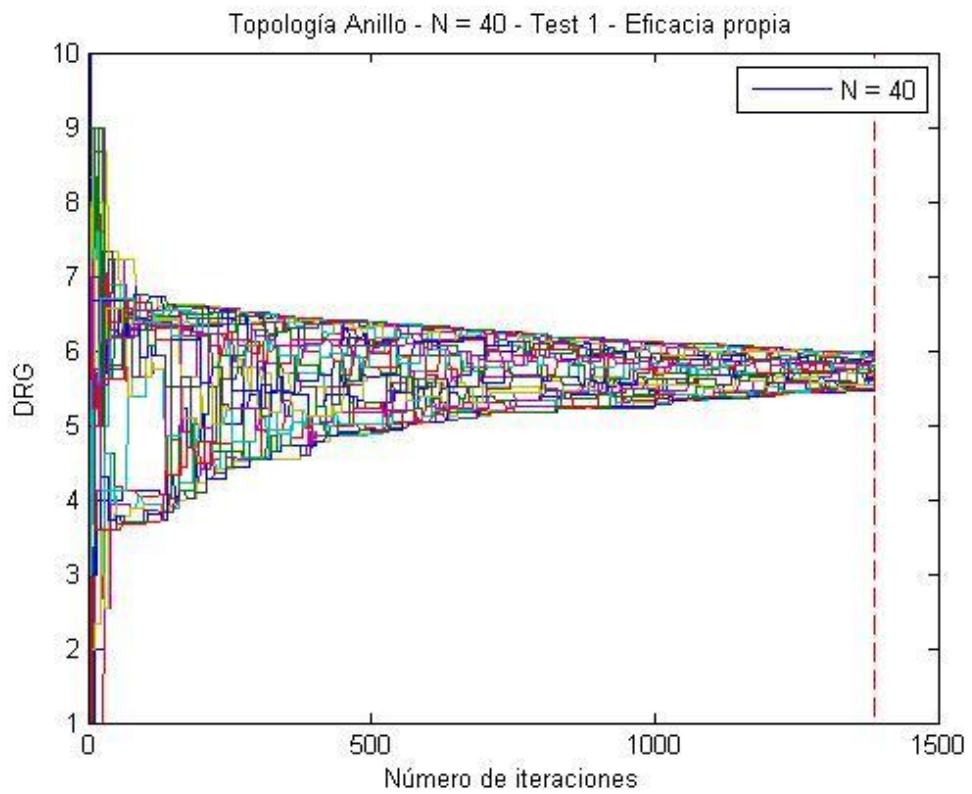
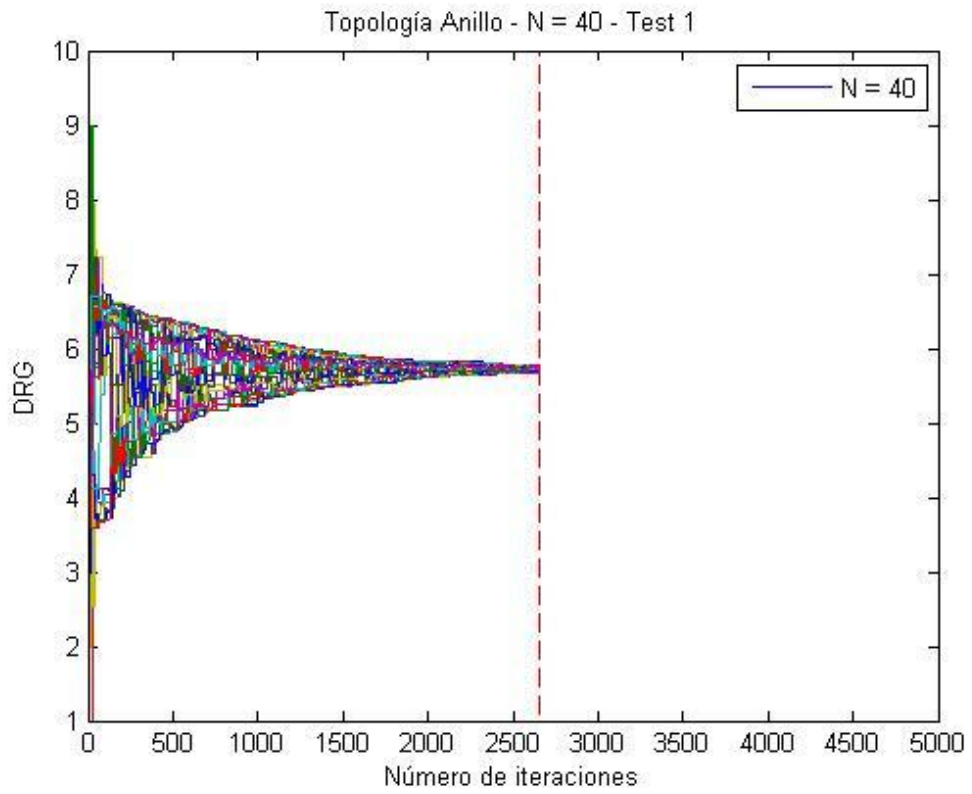


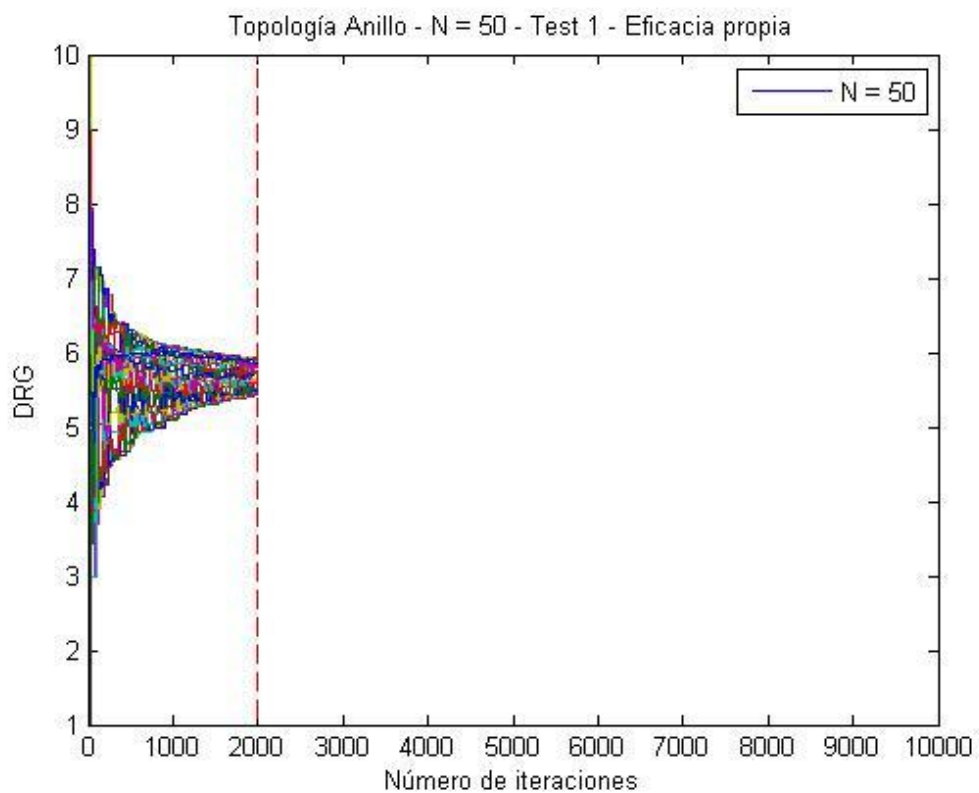
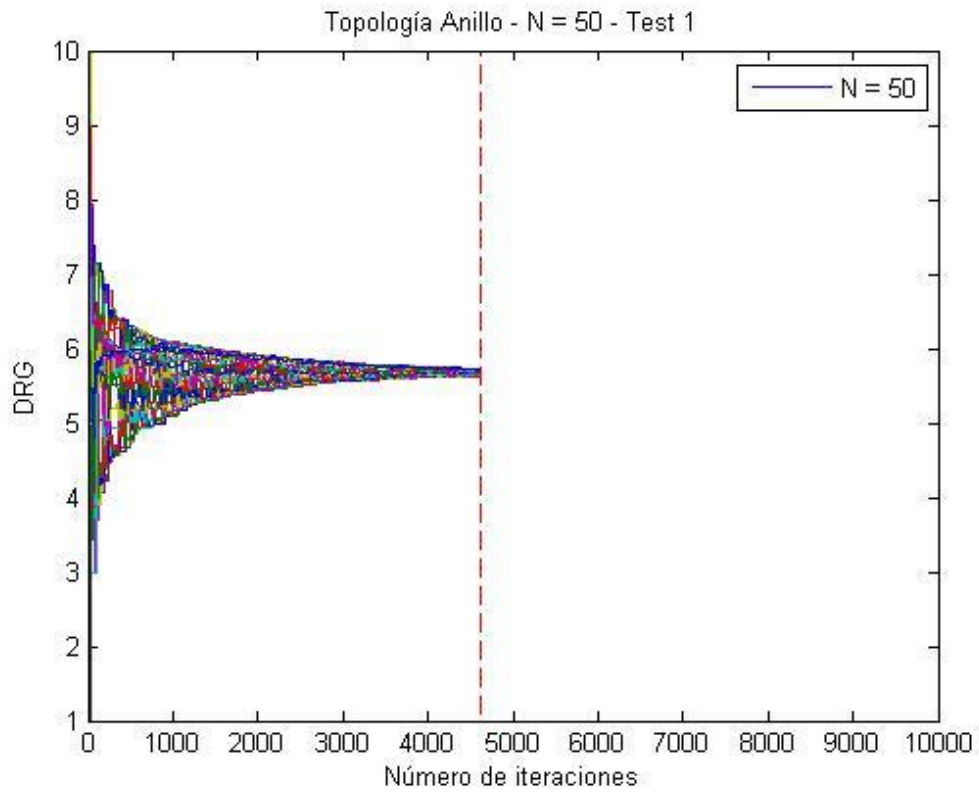










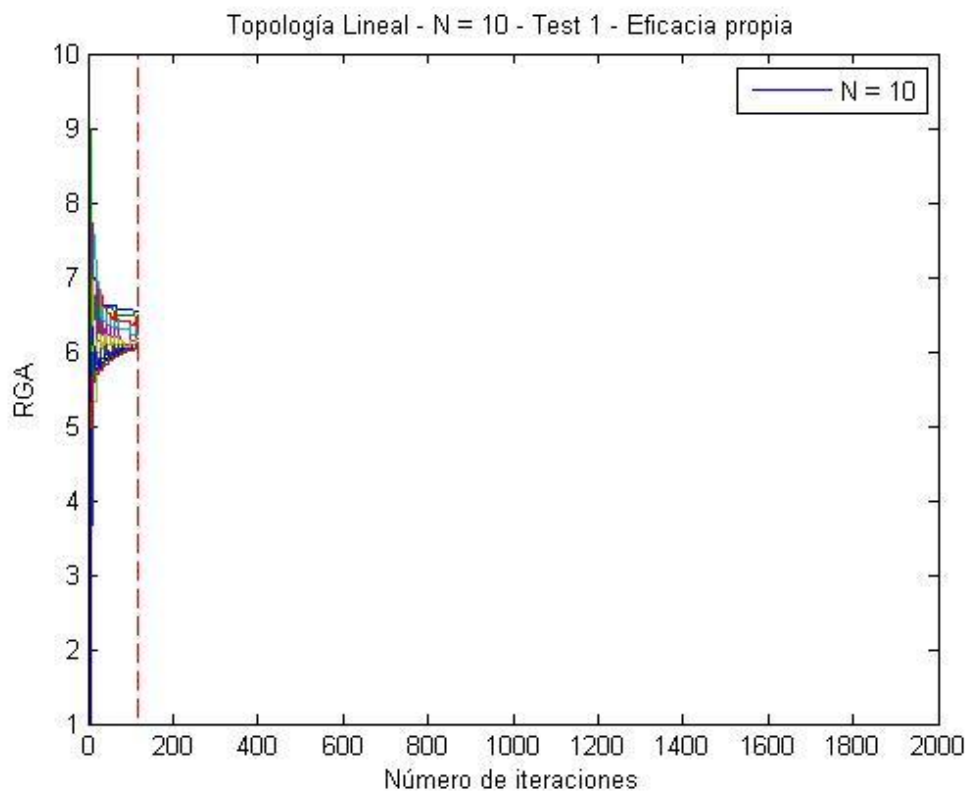
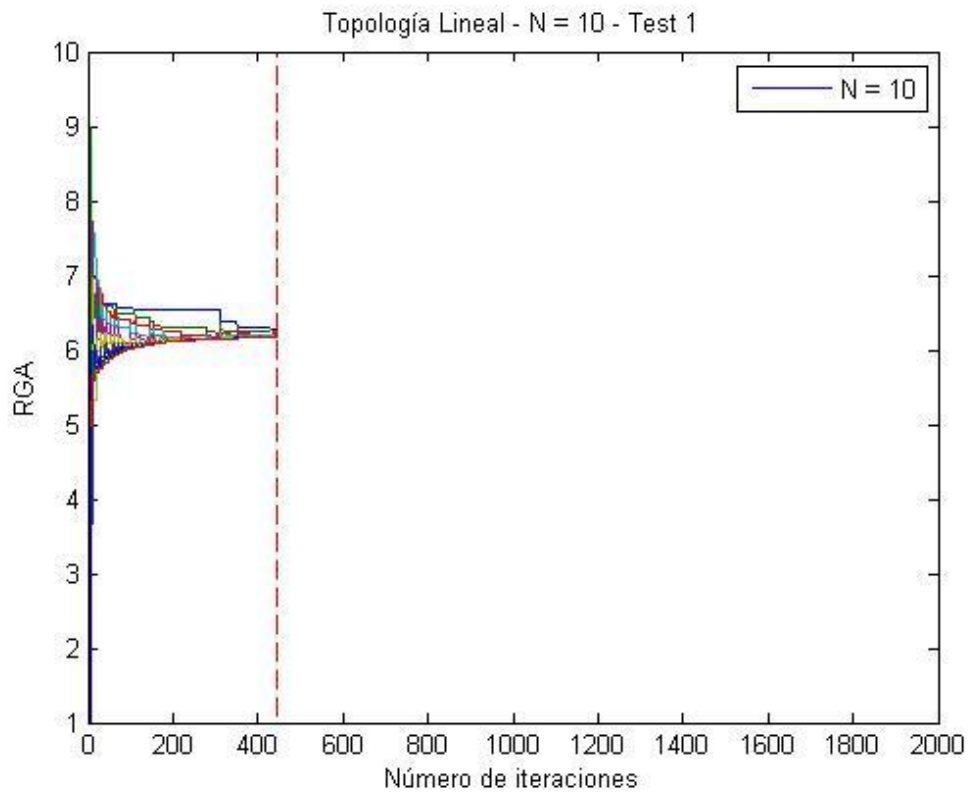


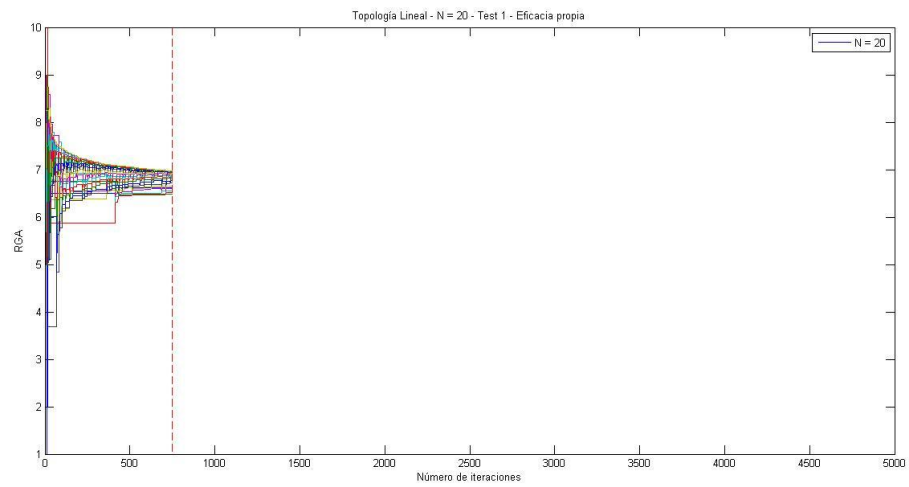
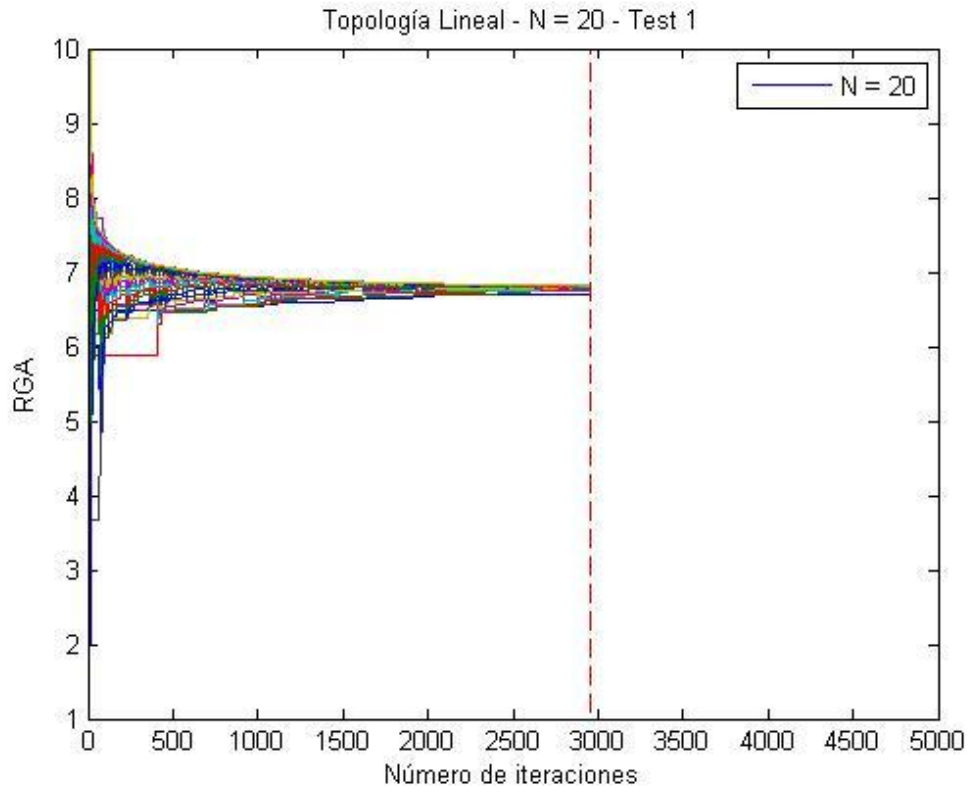
PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

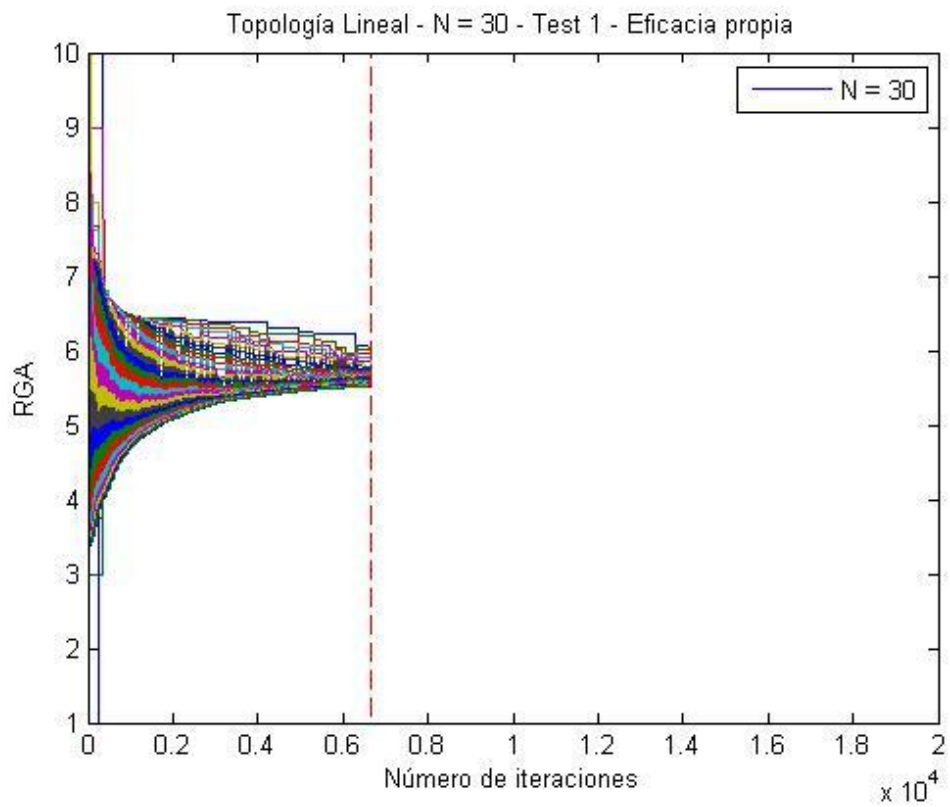
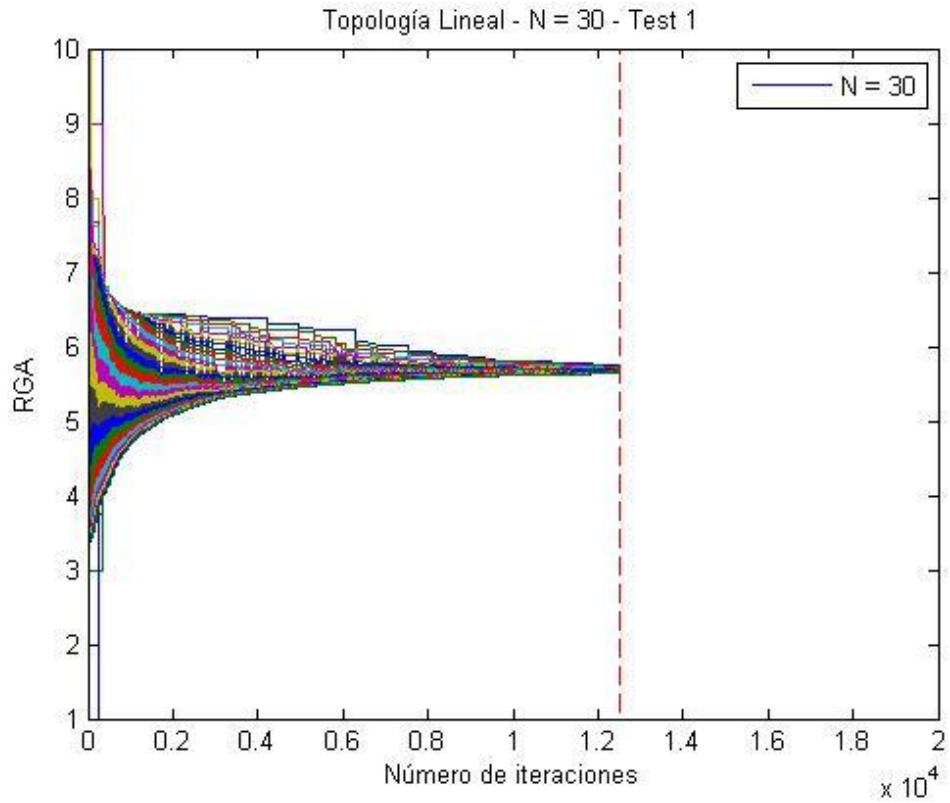
**IMPLEMENTACIÓN Y EVALUACIÓN DE UNA RED INALÁMBRICA
DE SENSORES A TRAVÉS DE ALGORITMOS DE
TRANSMISIÓN BASADOS EN GOSSIPING POR RF**

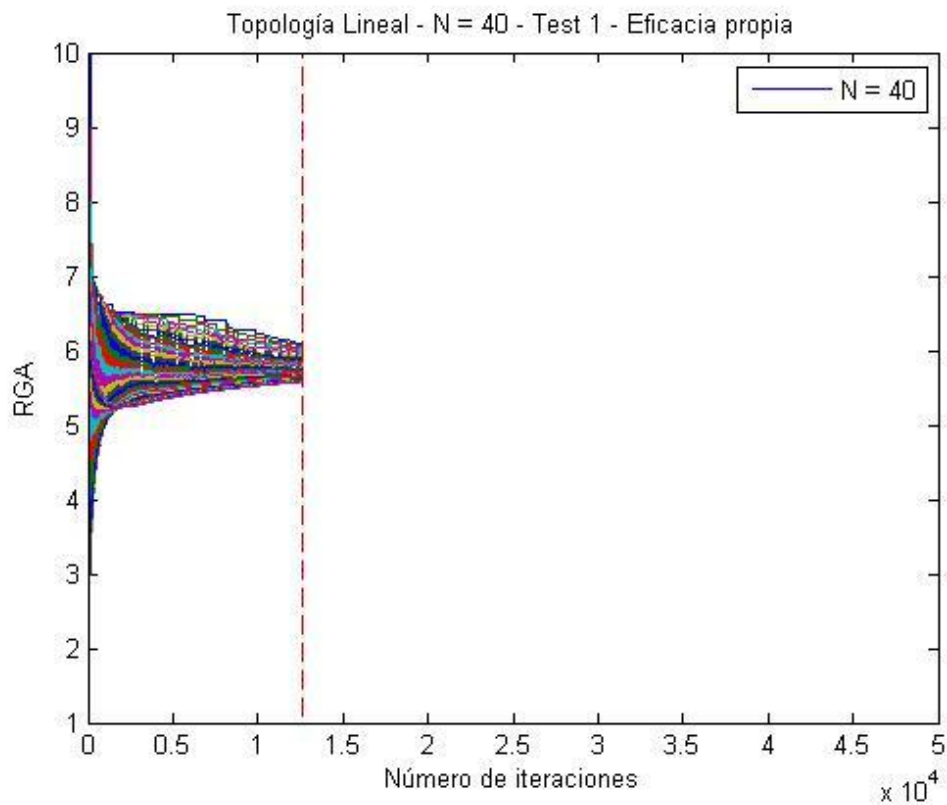
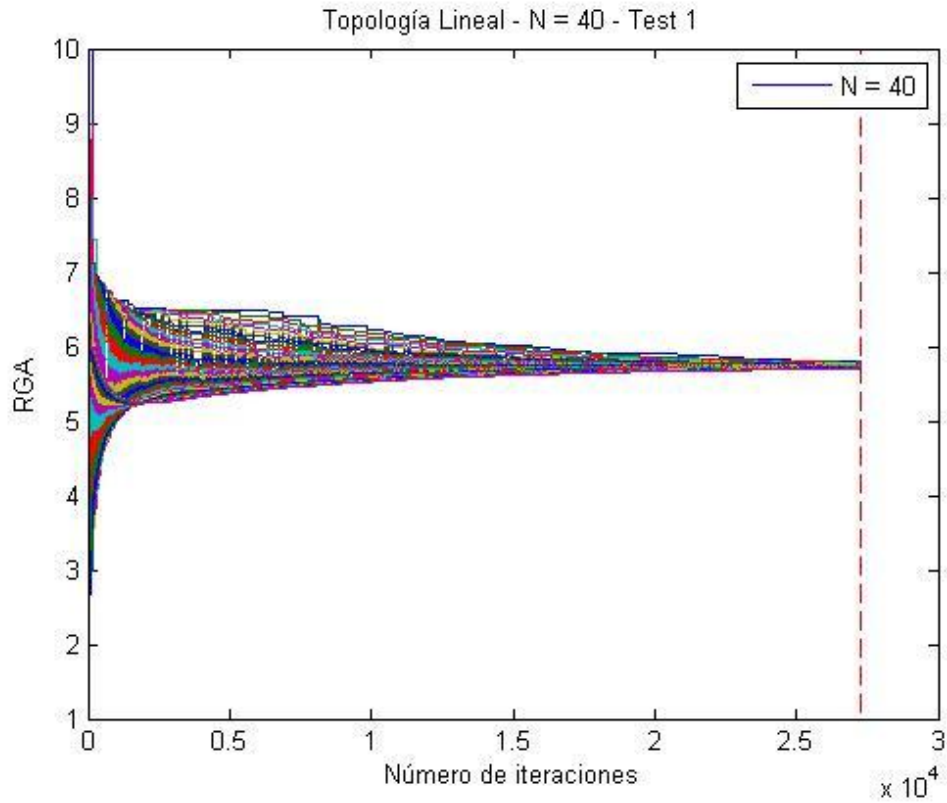
ANEXO 3

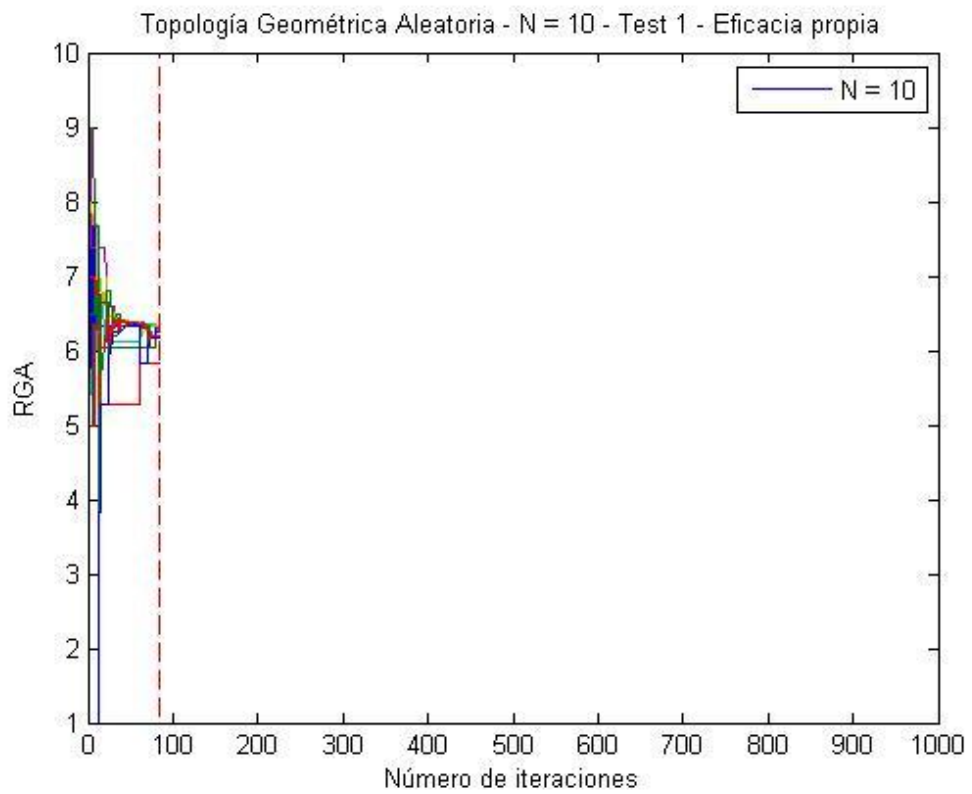
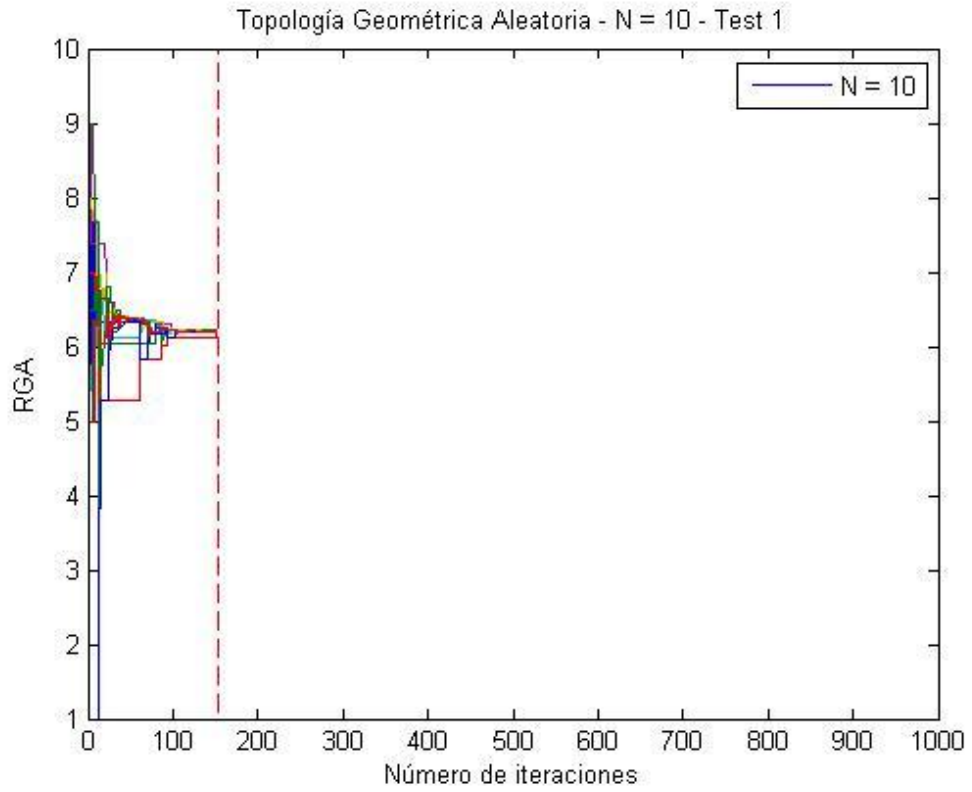
RESULTADOS DE SIMULACIÓN DEL RGA

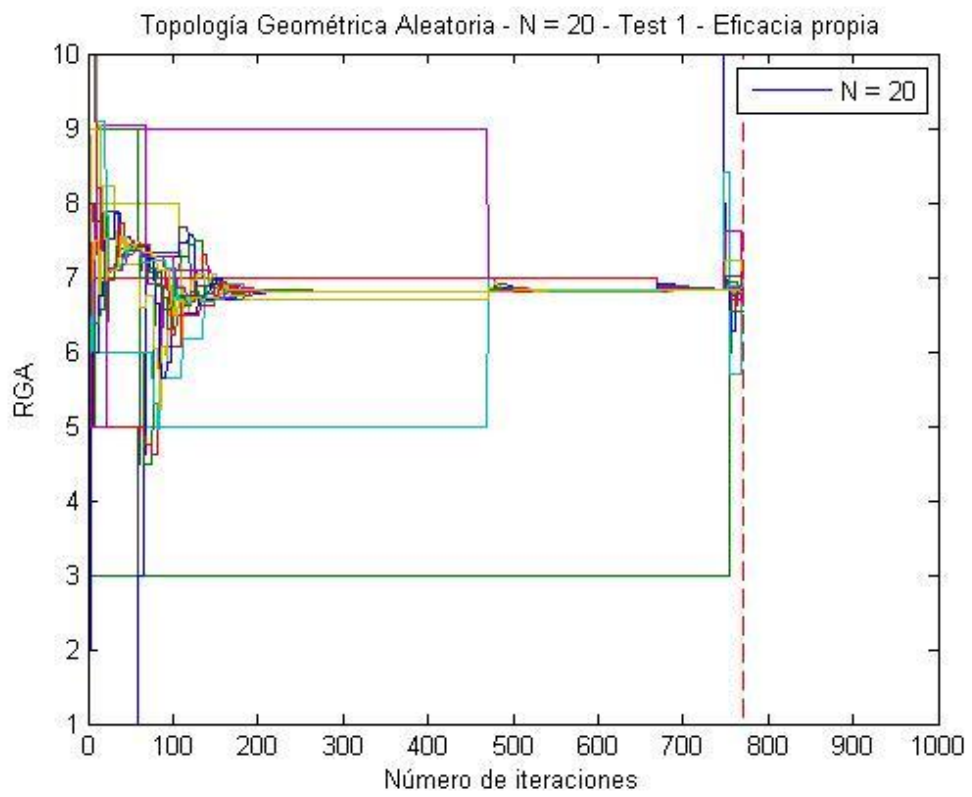
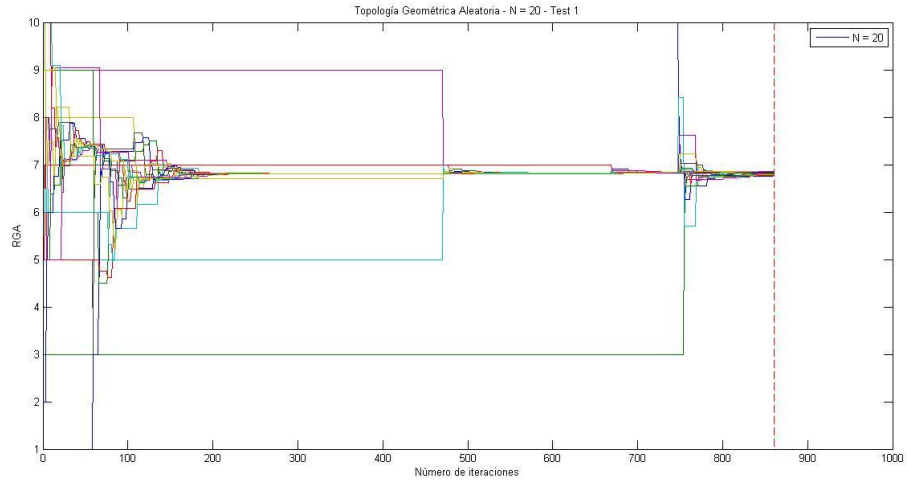


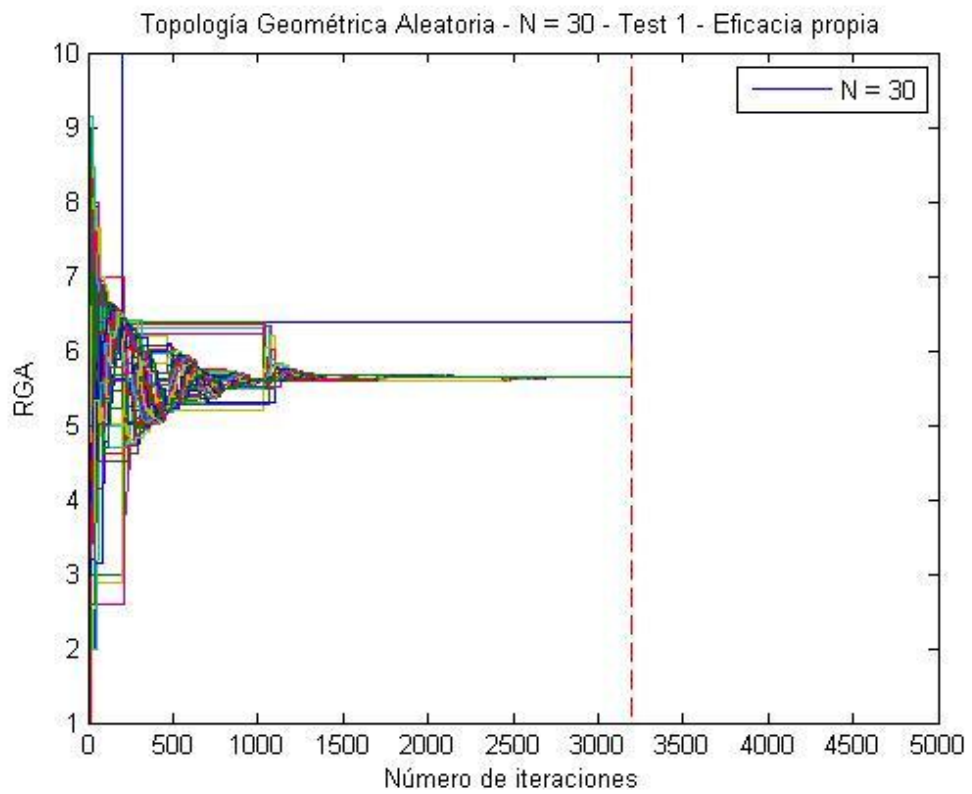
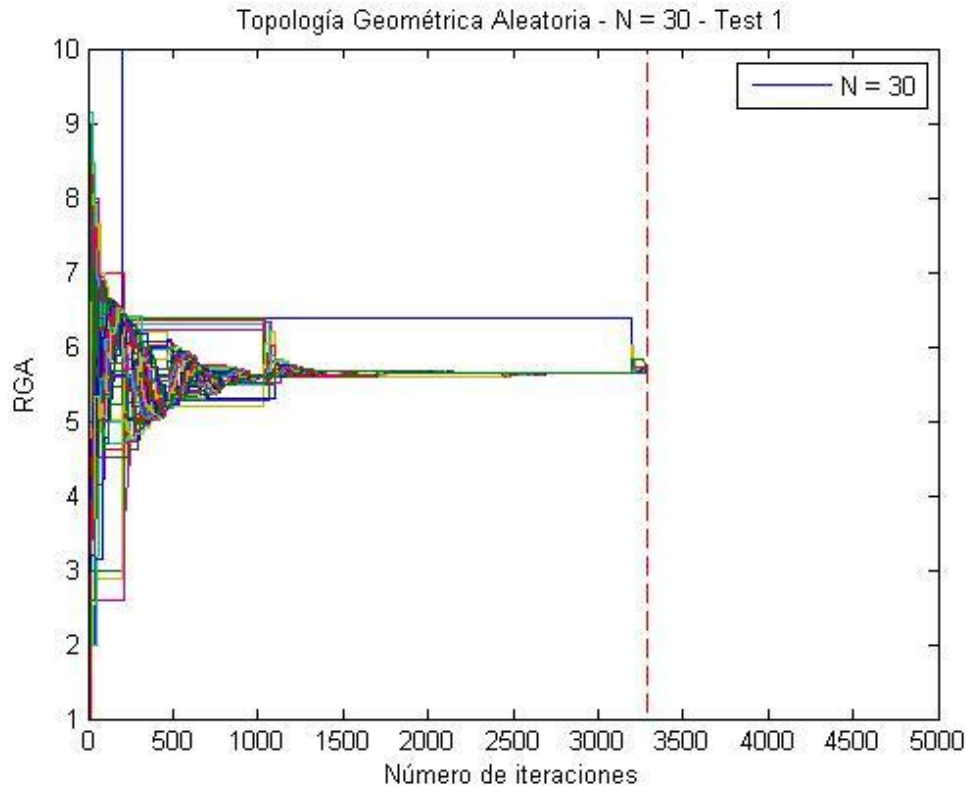


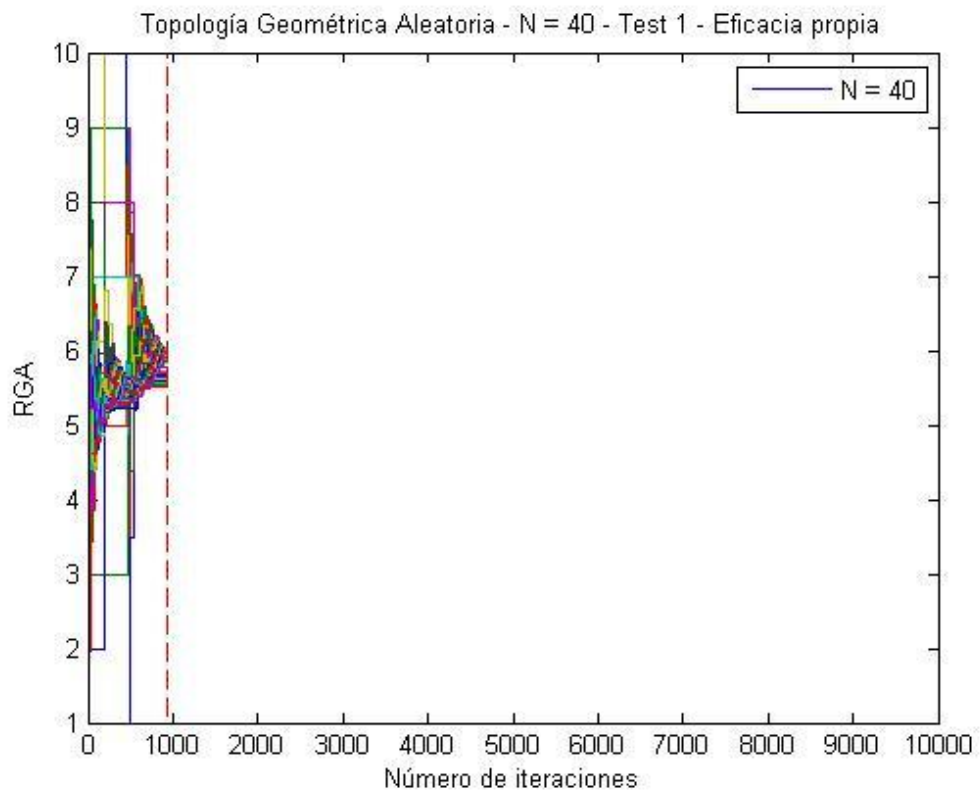
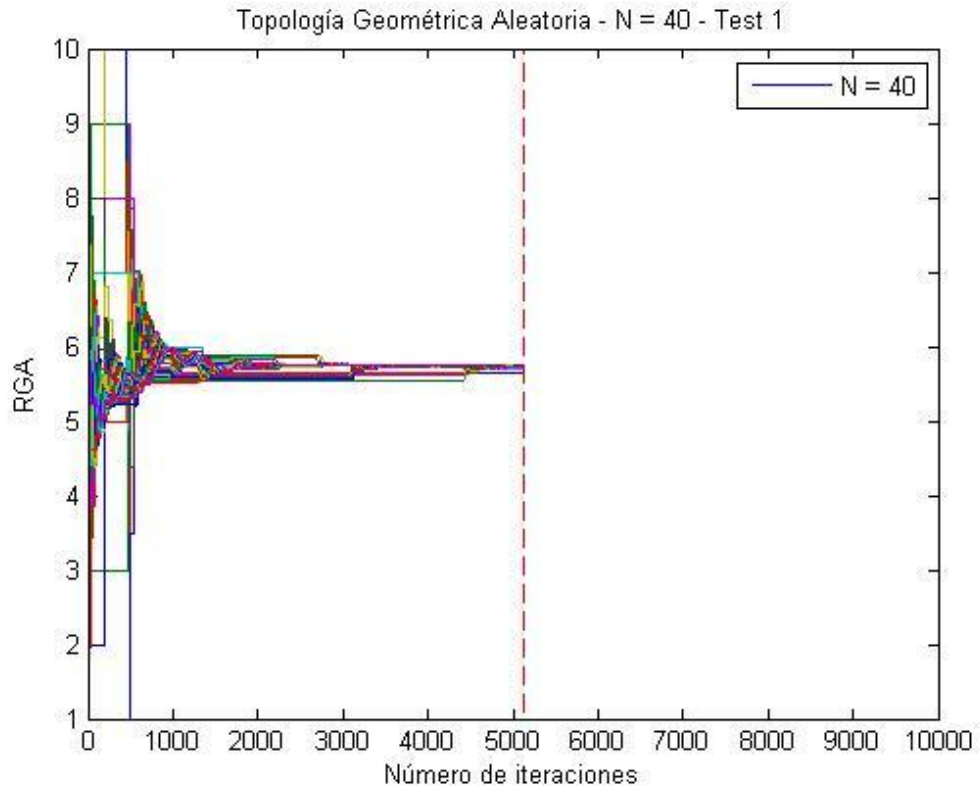


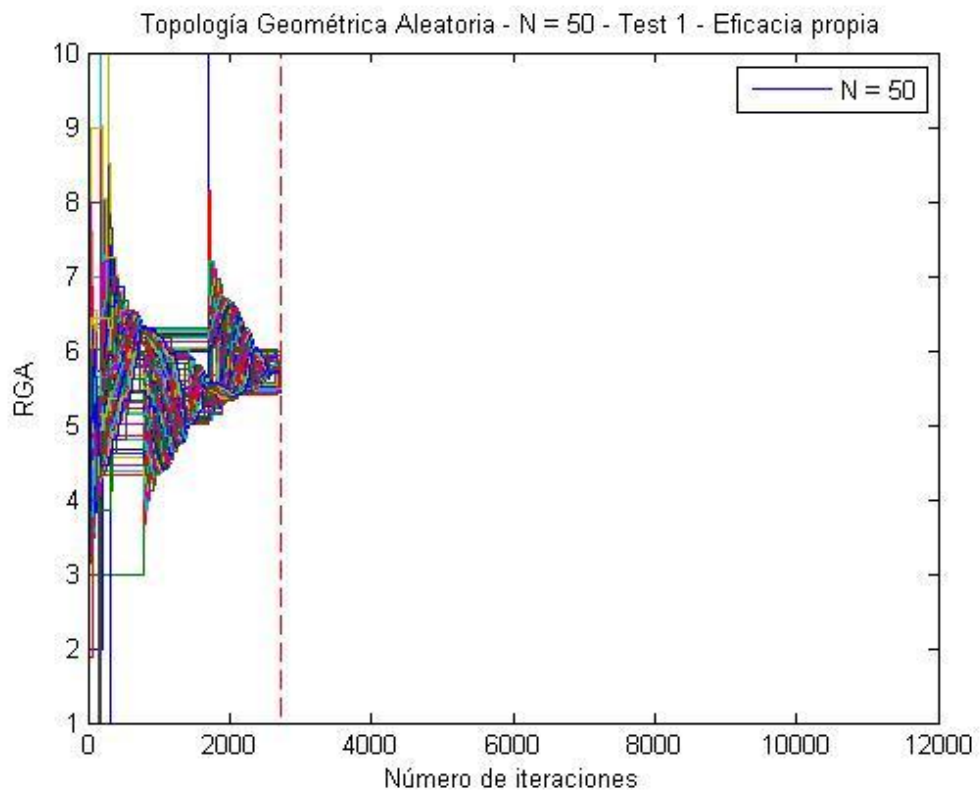
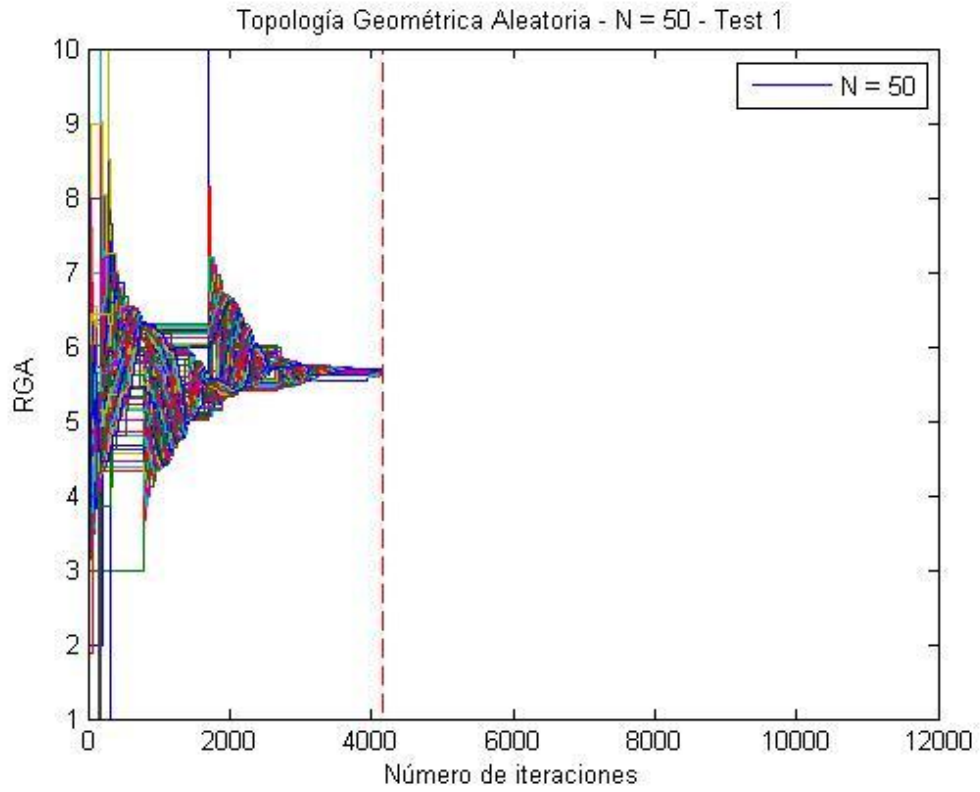


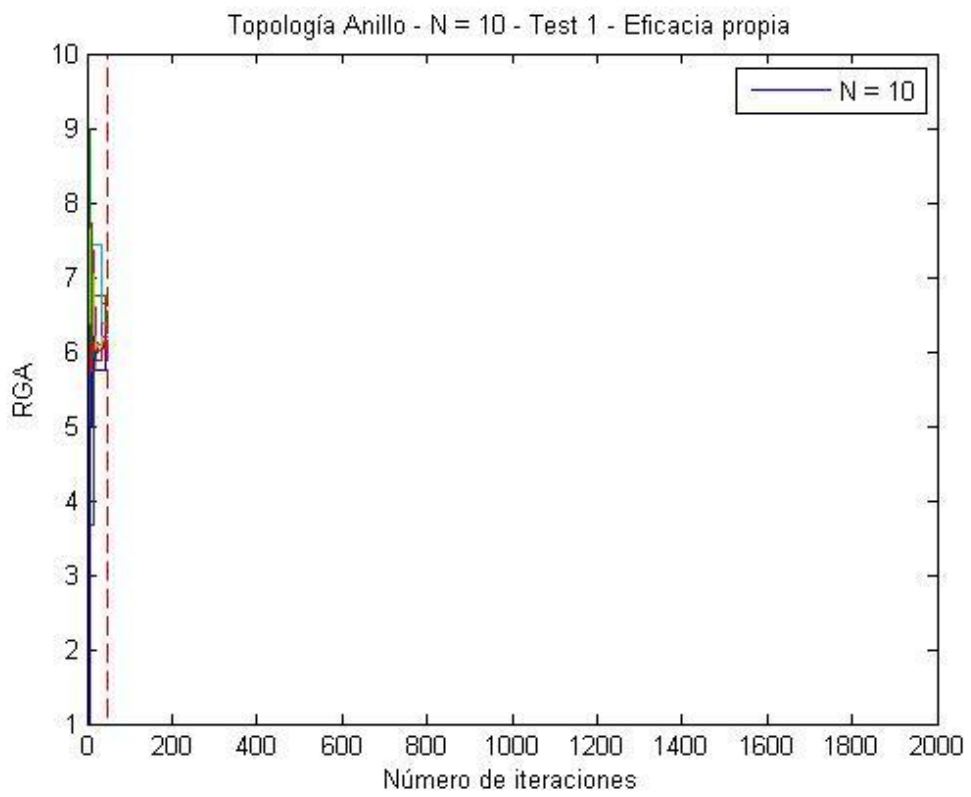
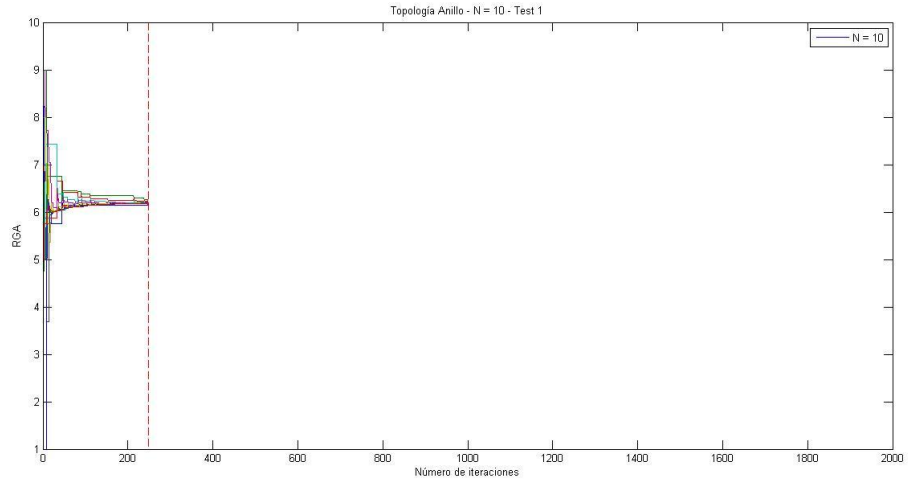


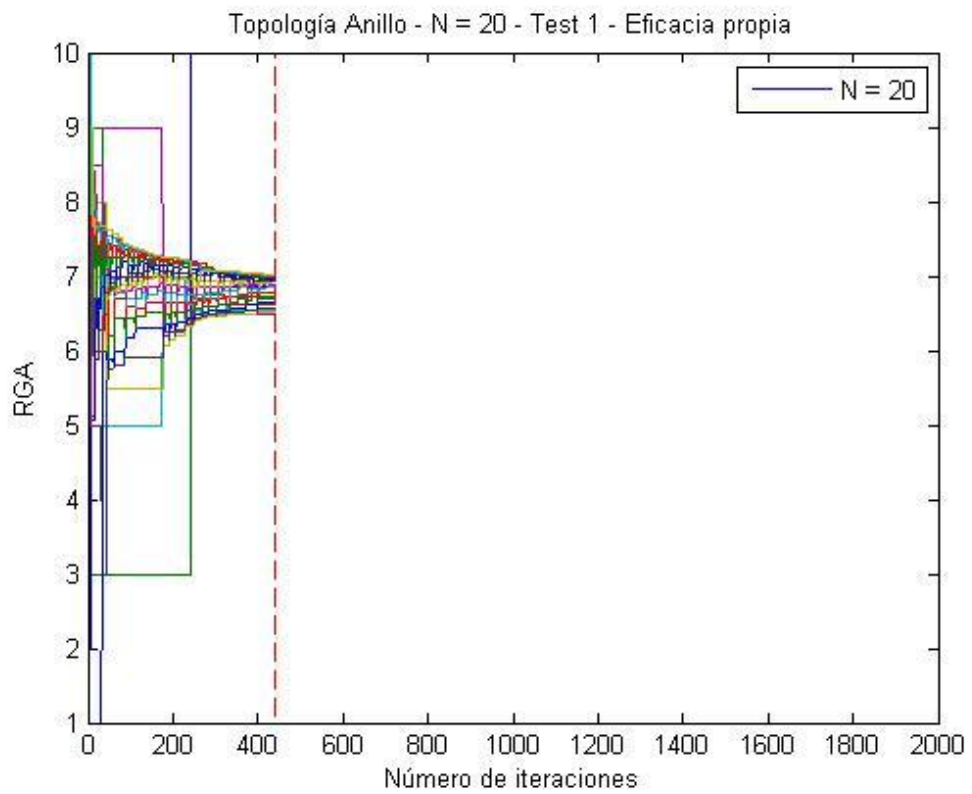
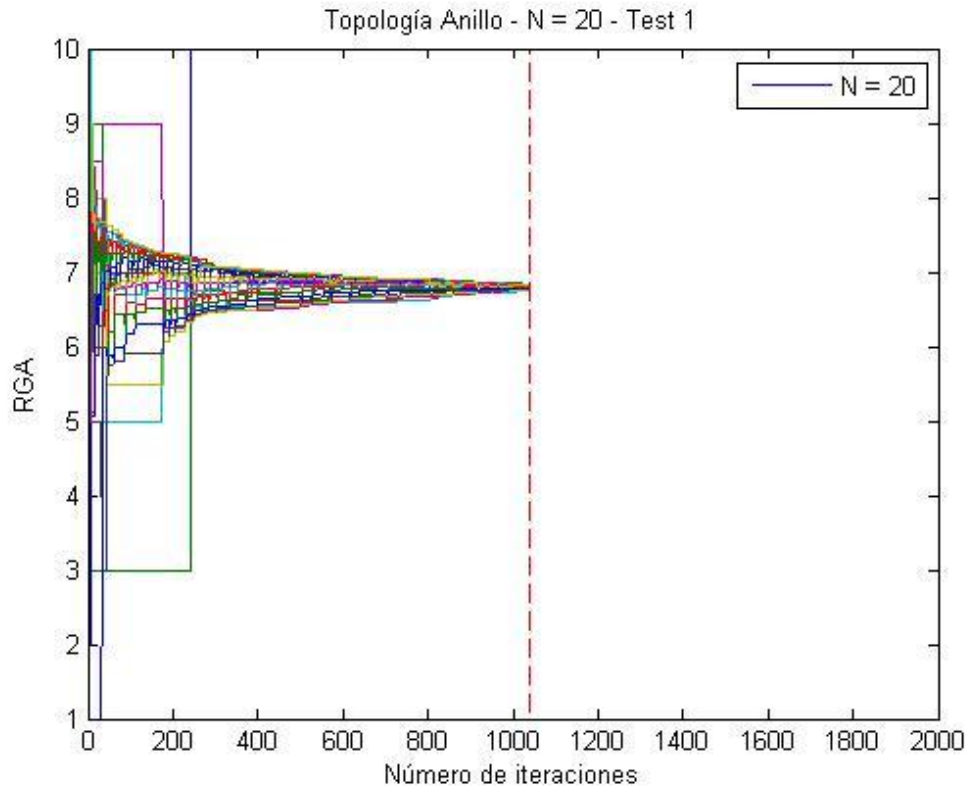


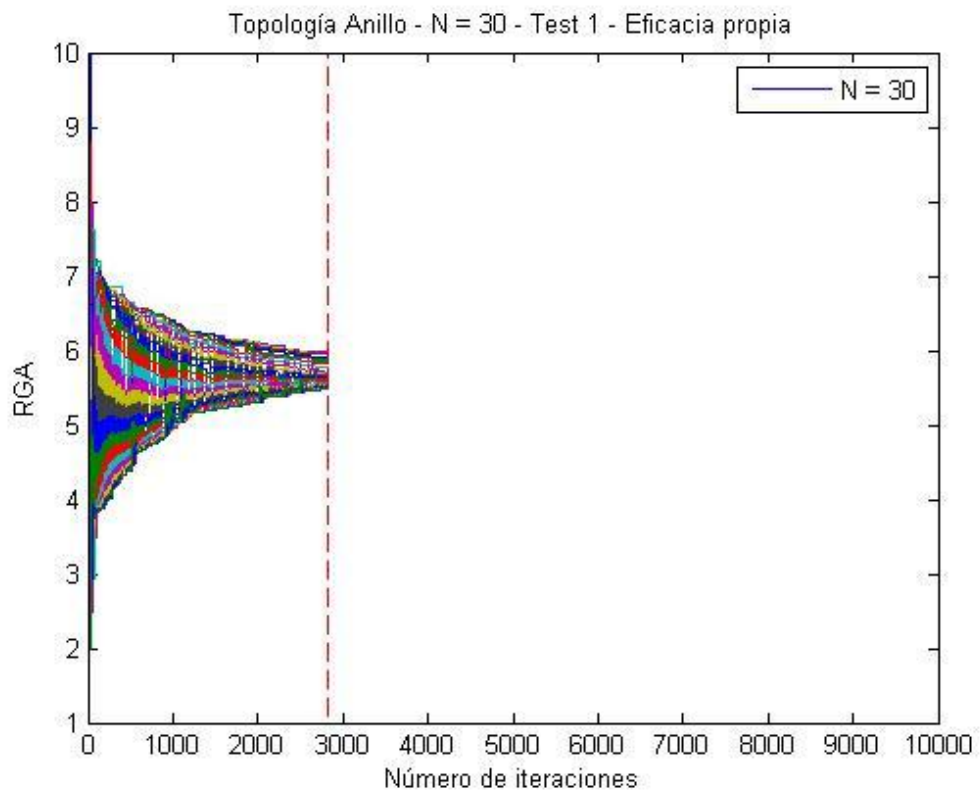
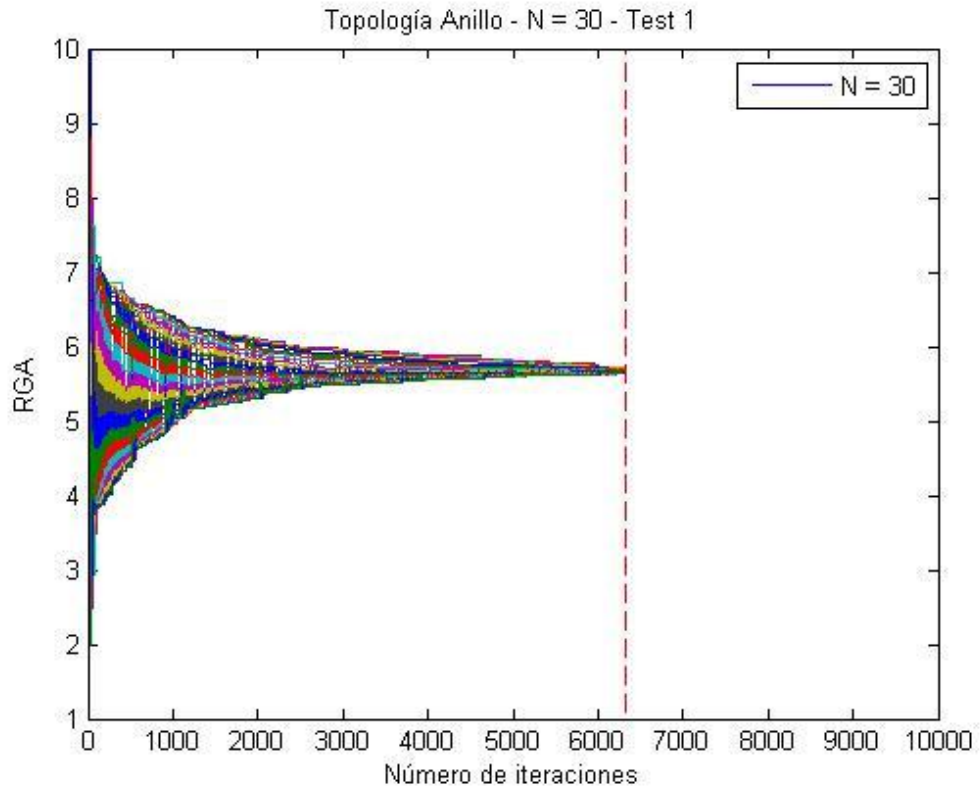


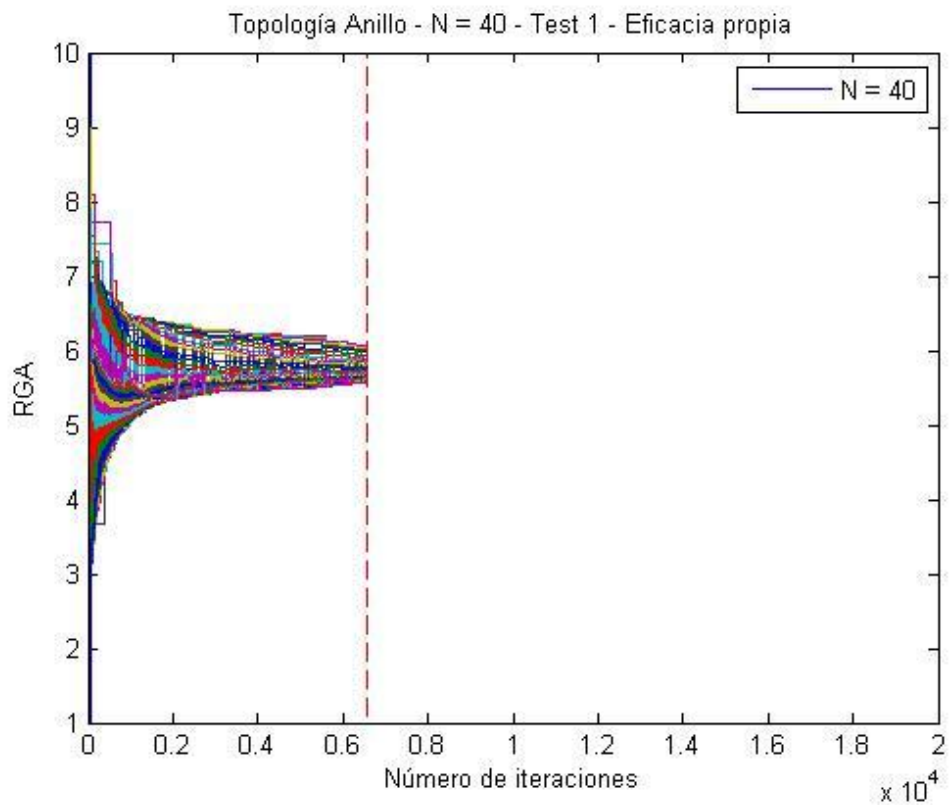
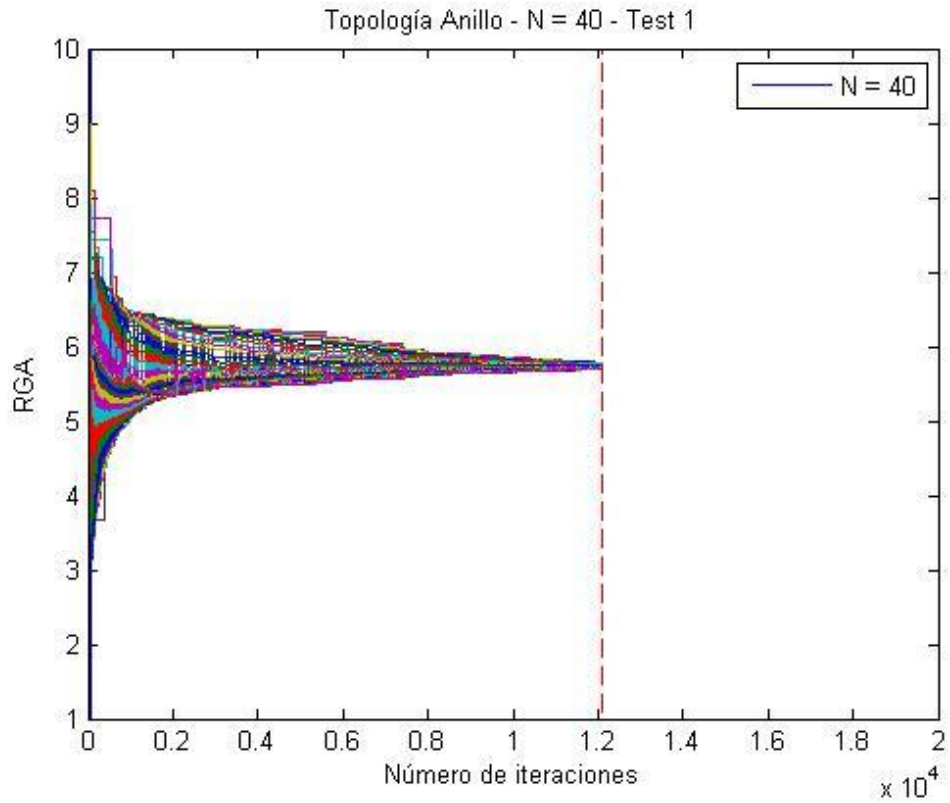










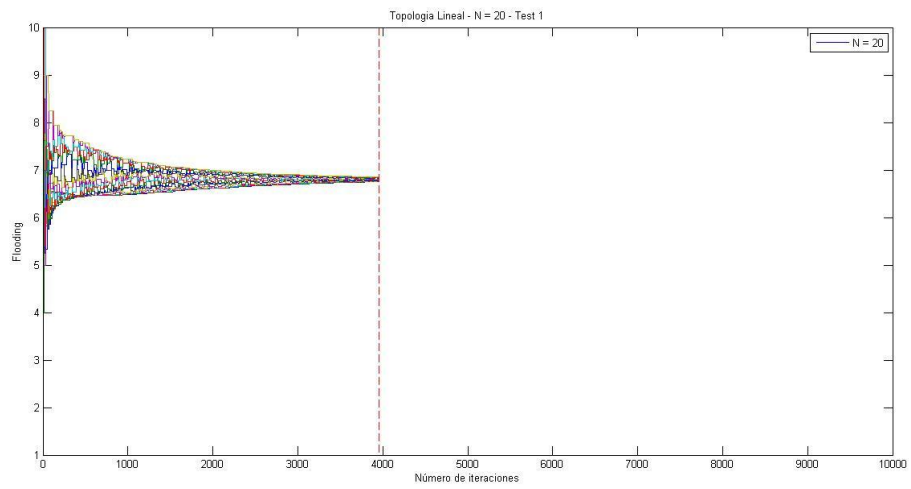
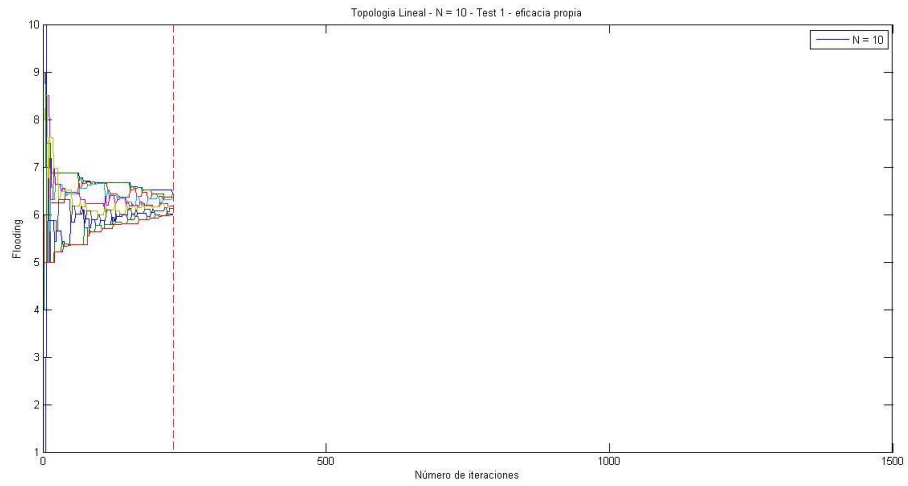
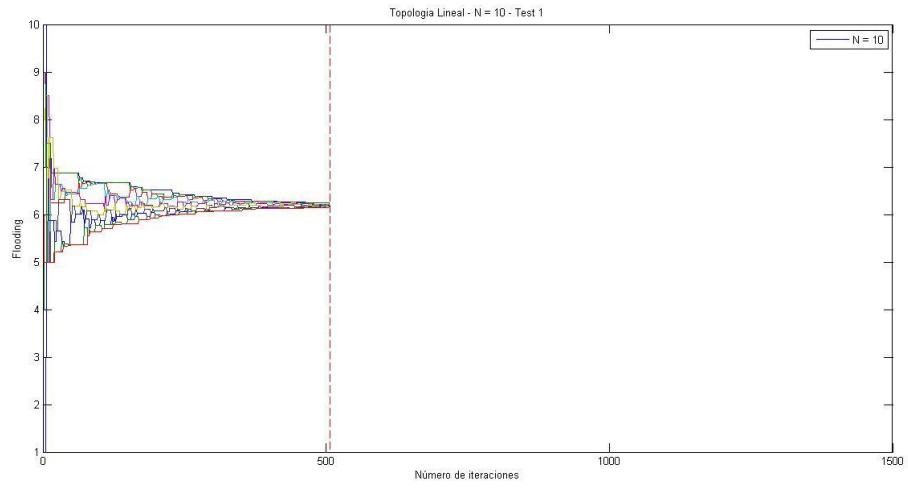


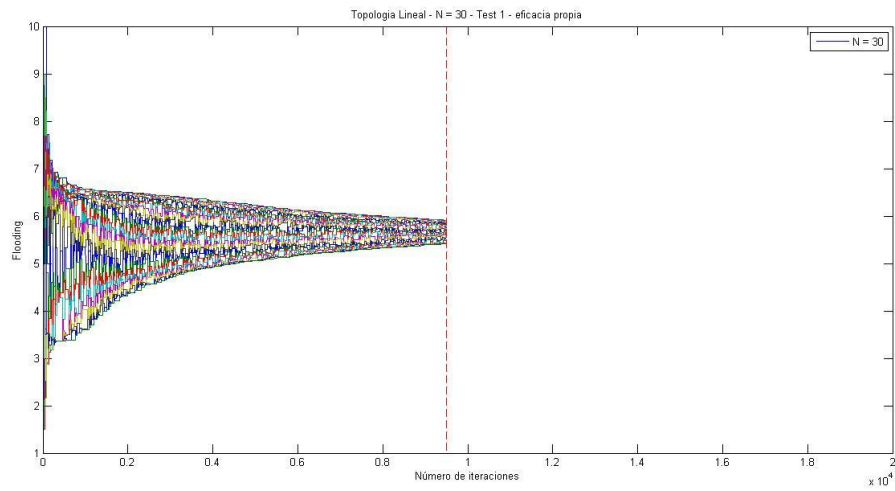
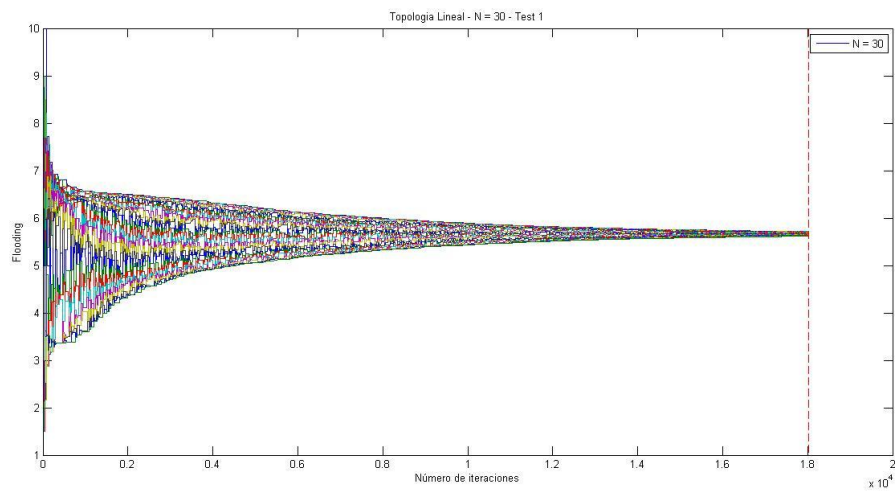
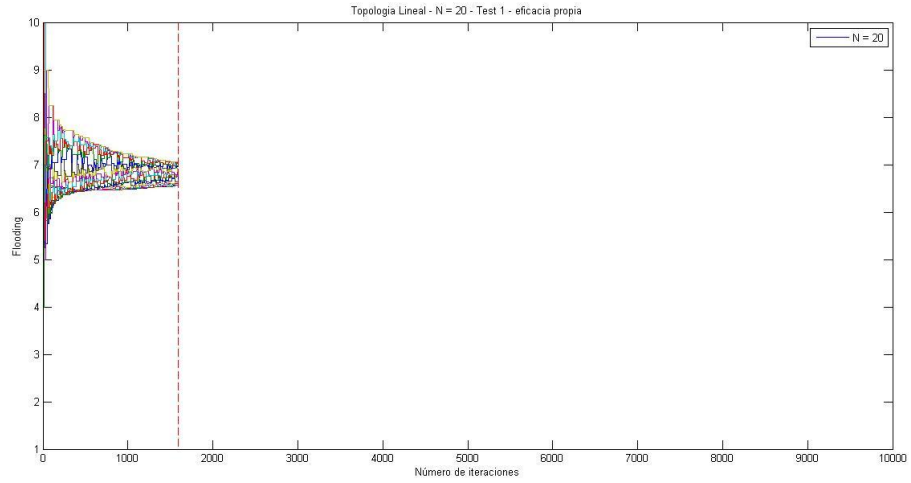
PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

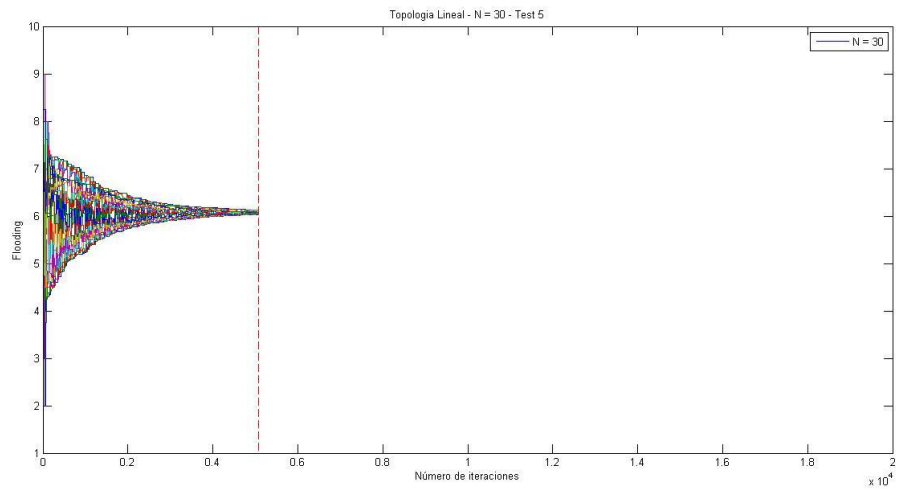
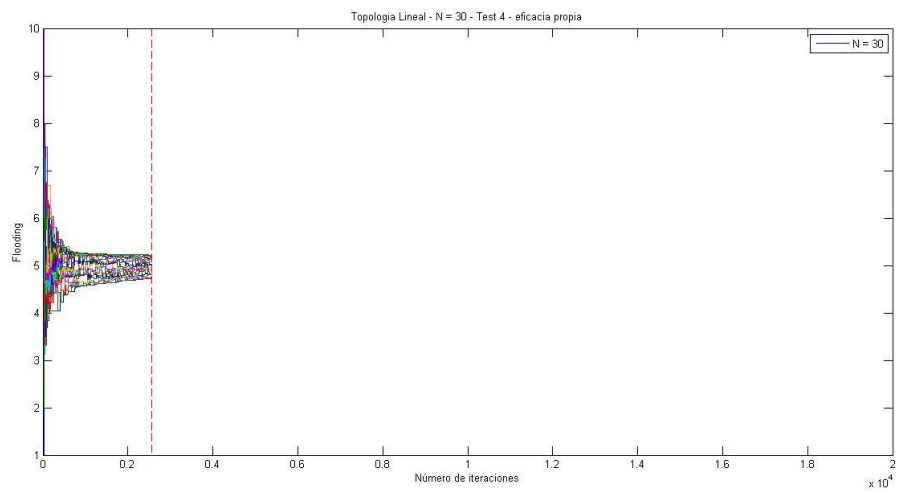
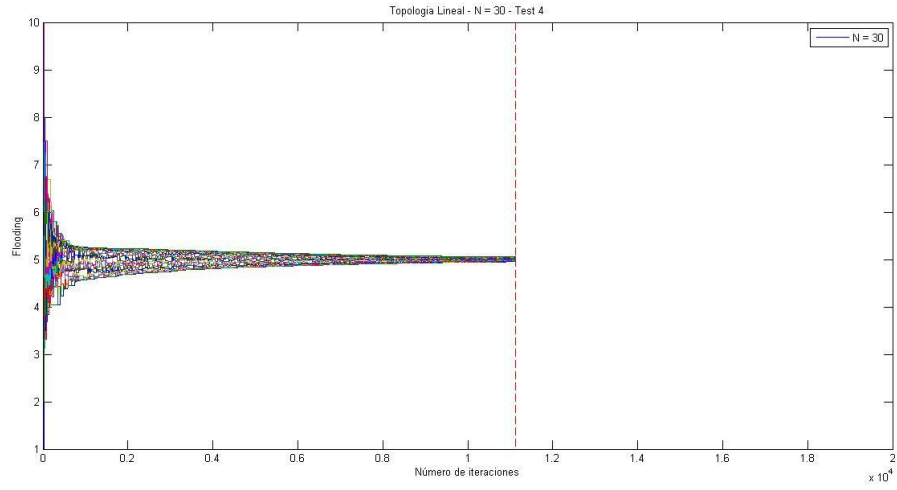
**IMPLEMENTACIÓN Y EVALUACIÓN DE UNA RED INALÁMBRICA
DE SENSORES A TRAVÉS DE ALGORITMOS DE
TRANSMISIÓN BASADOS EN GOSSIPING POR RF**

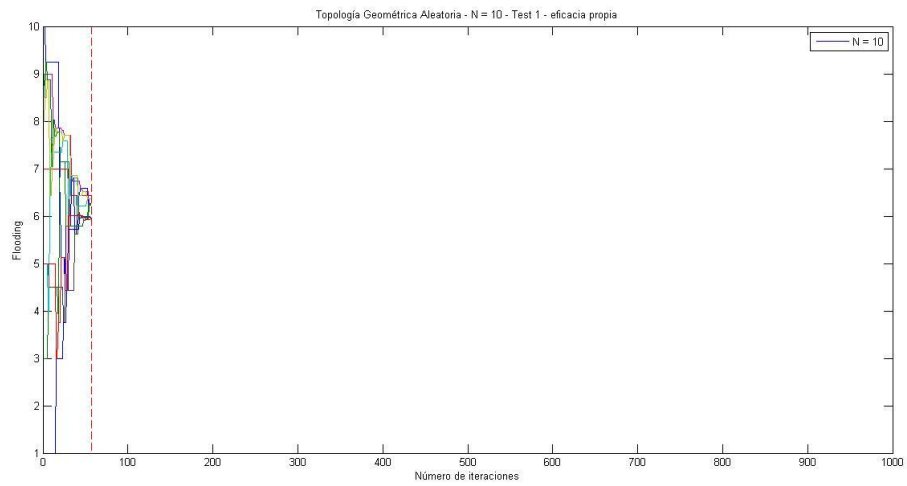
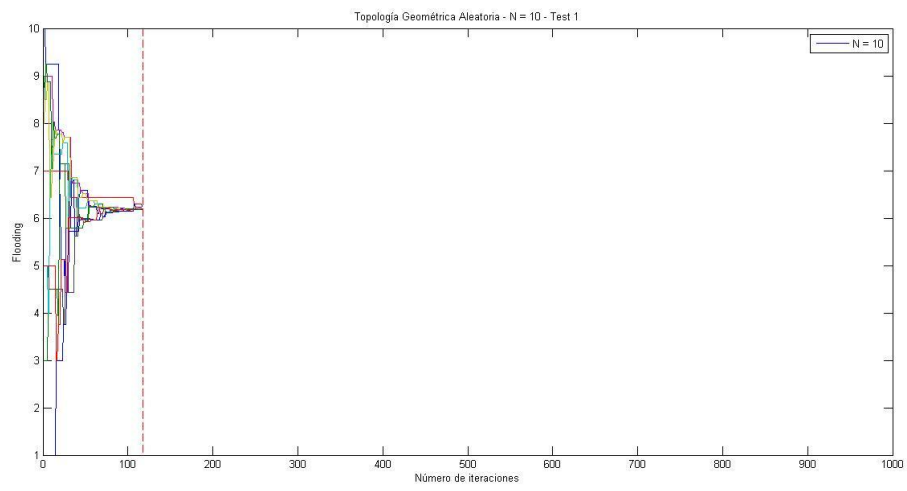
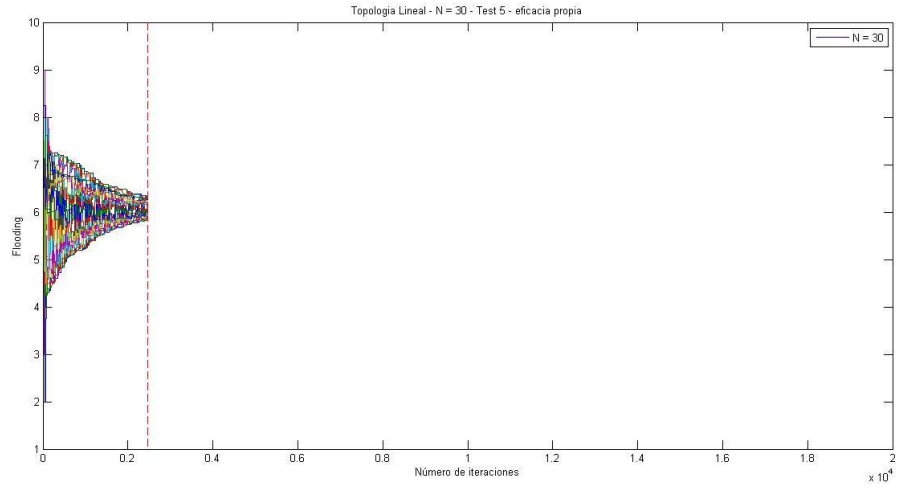
ANEXO 4

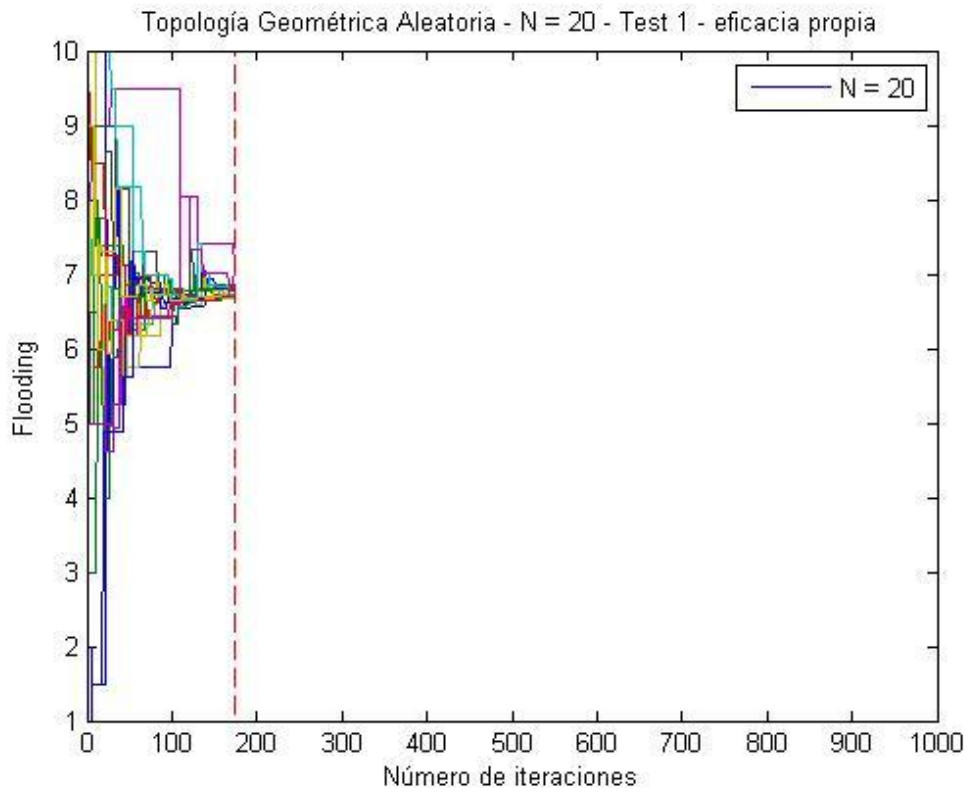
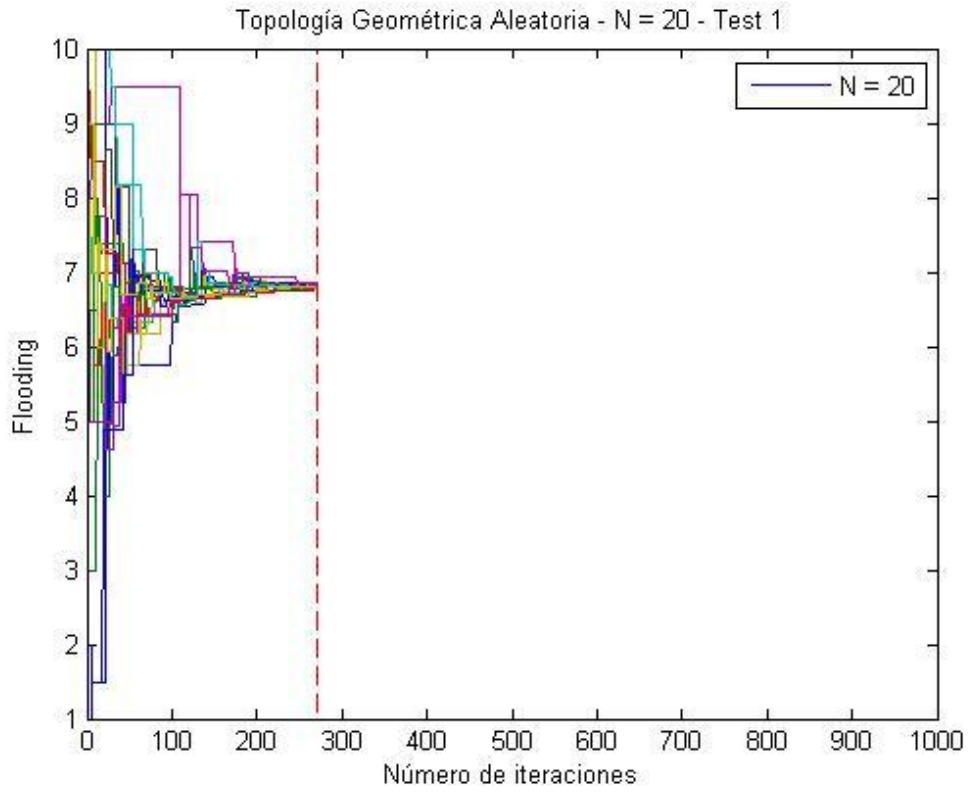
RESULTADOS DE SIMULACIÓN DEL FLOOD

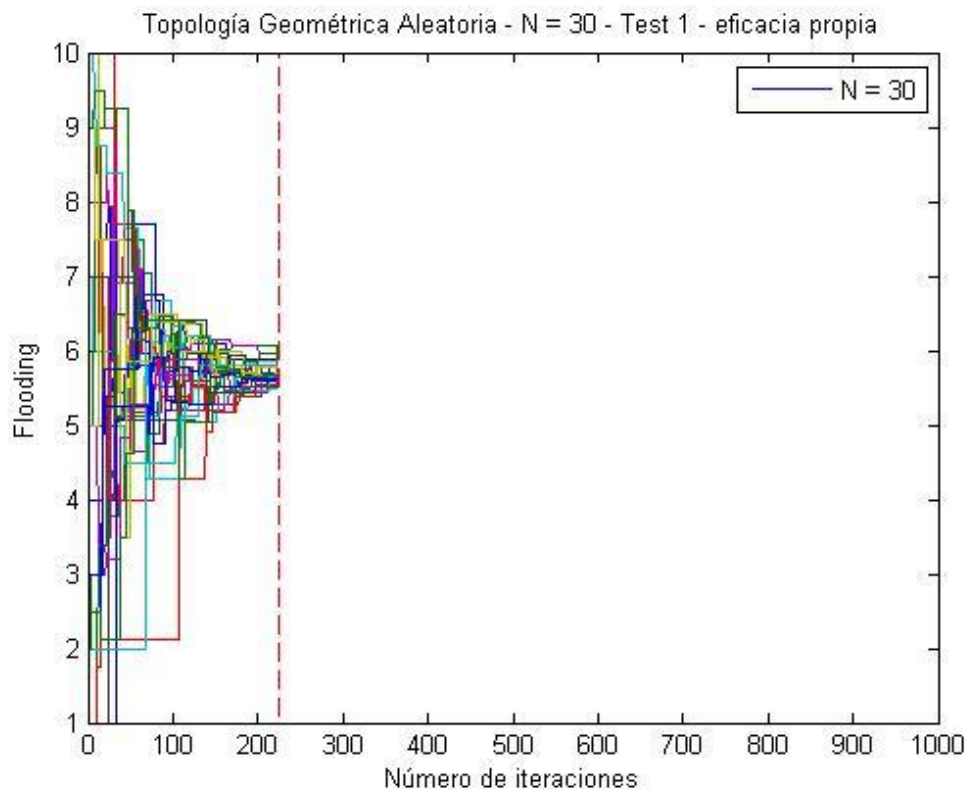
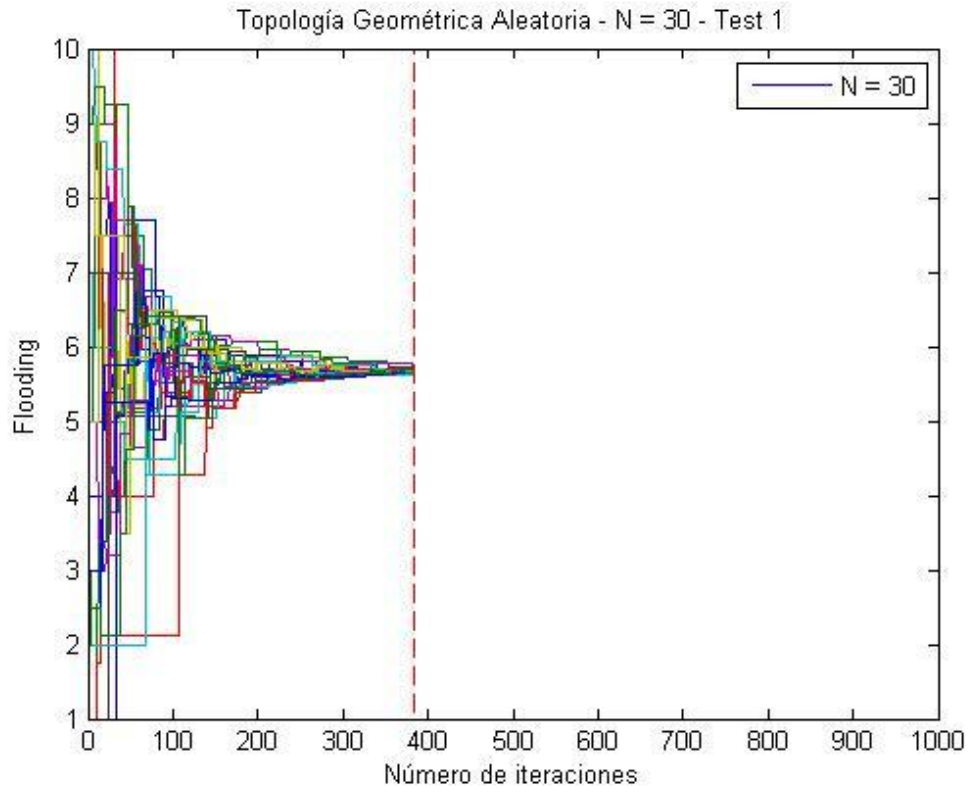


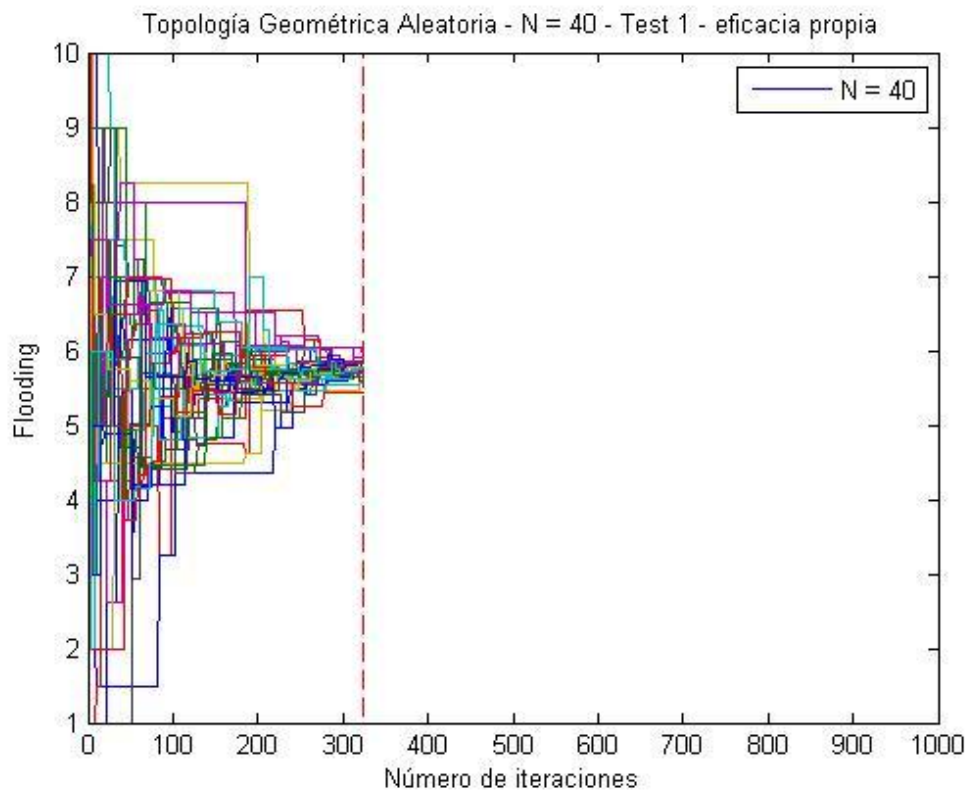
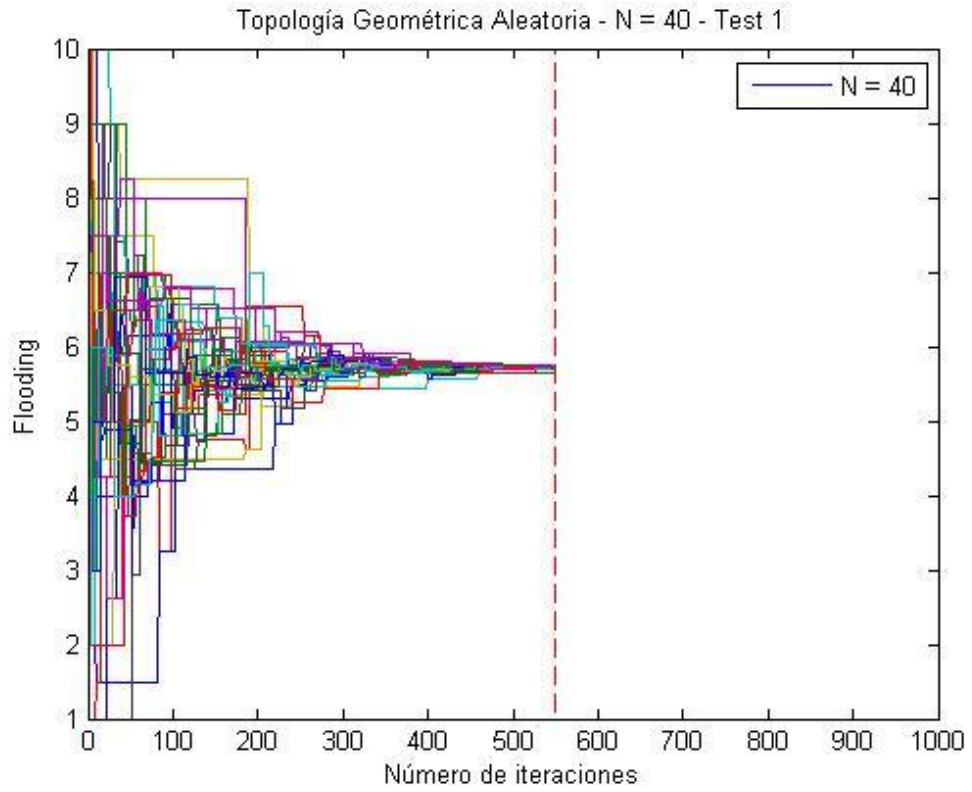


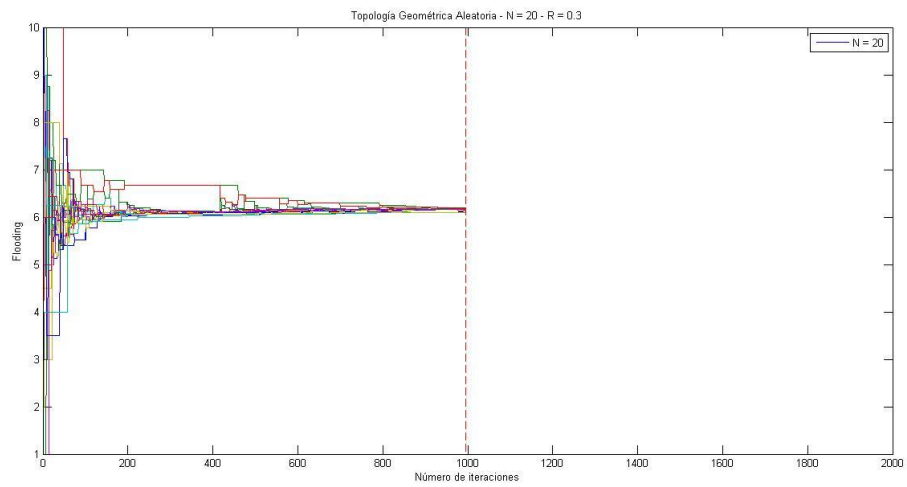
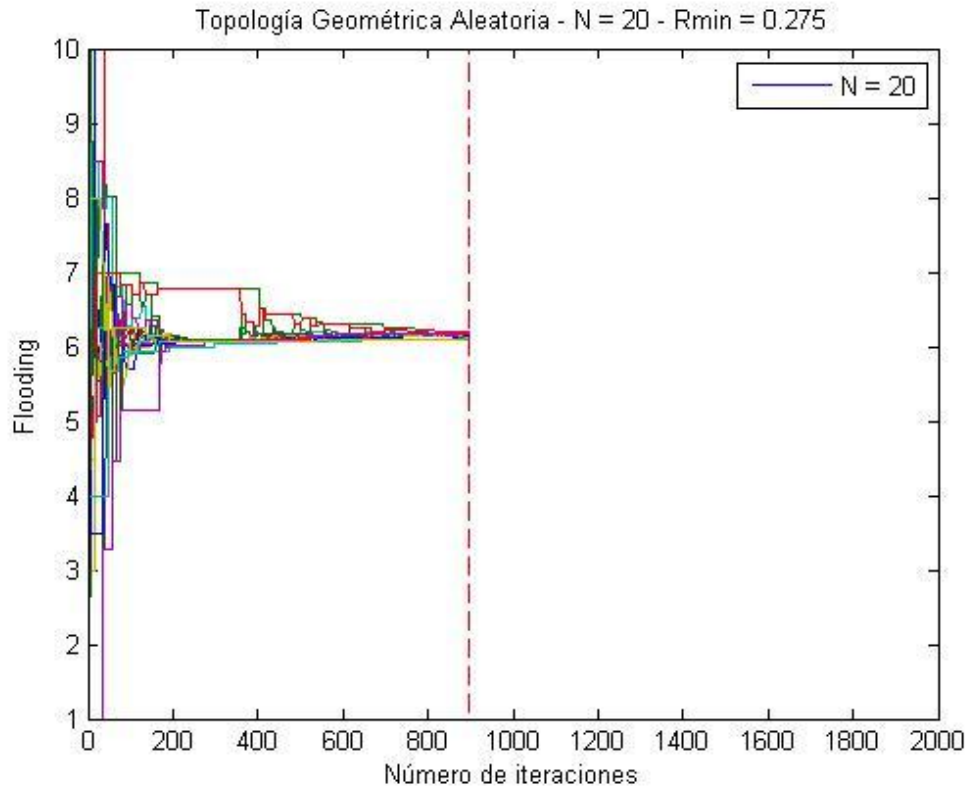


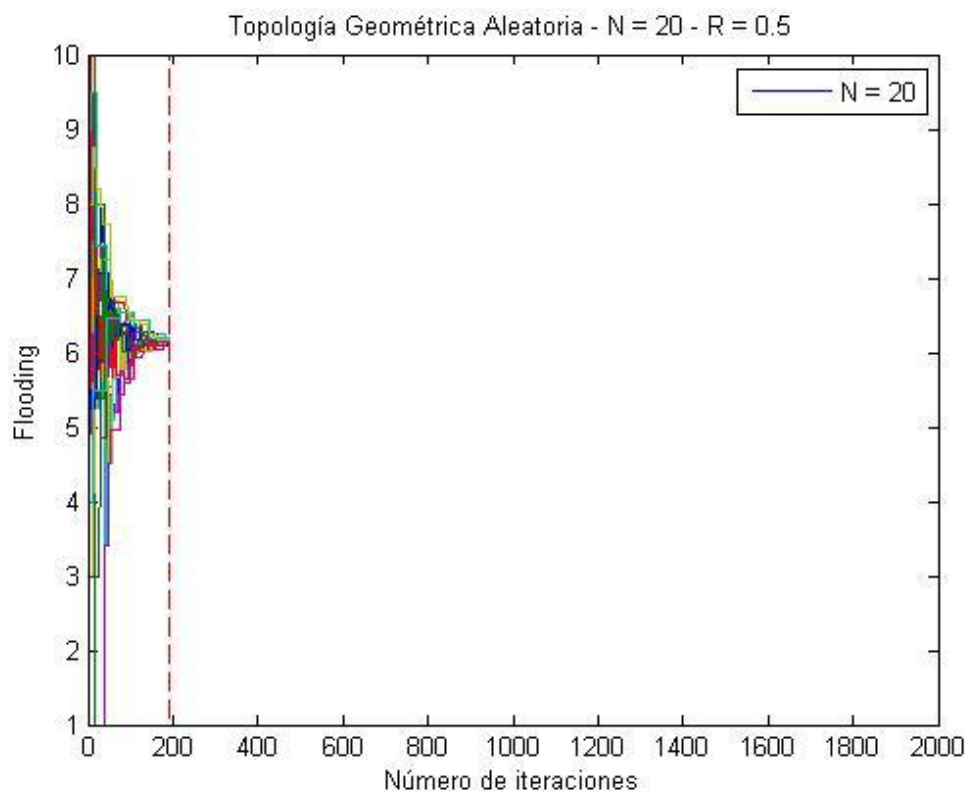
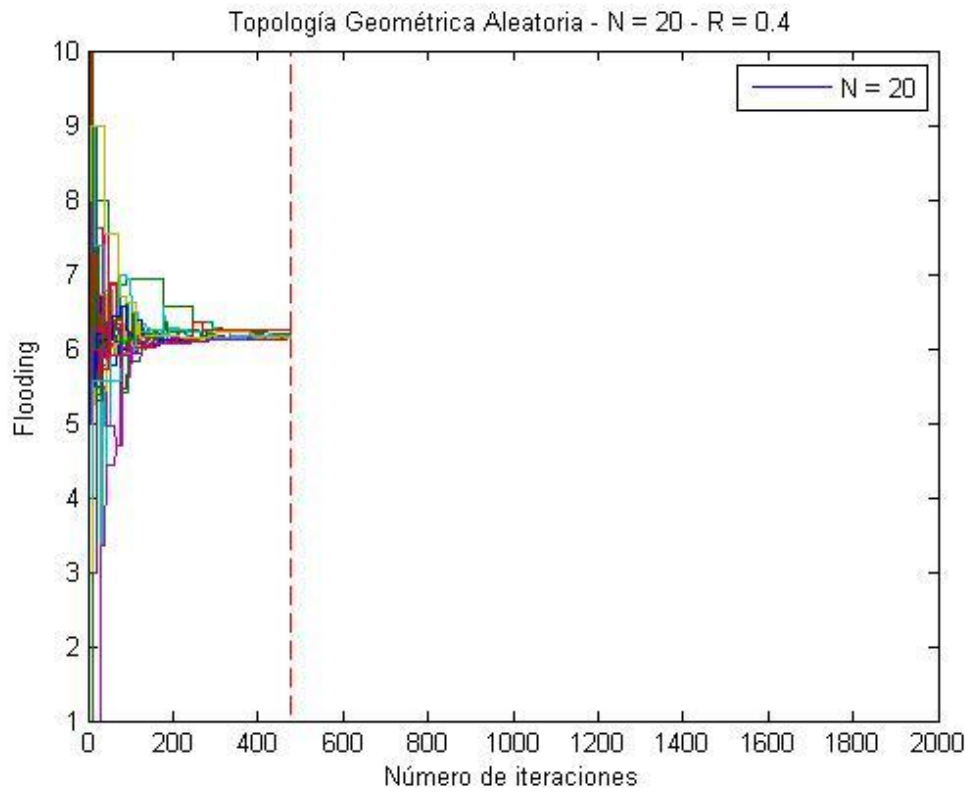


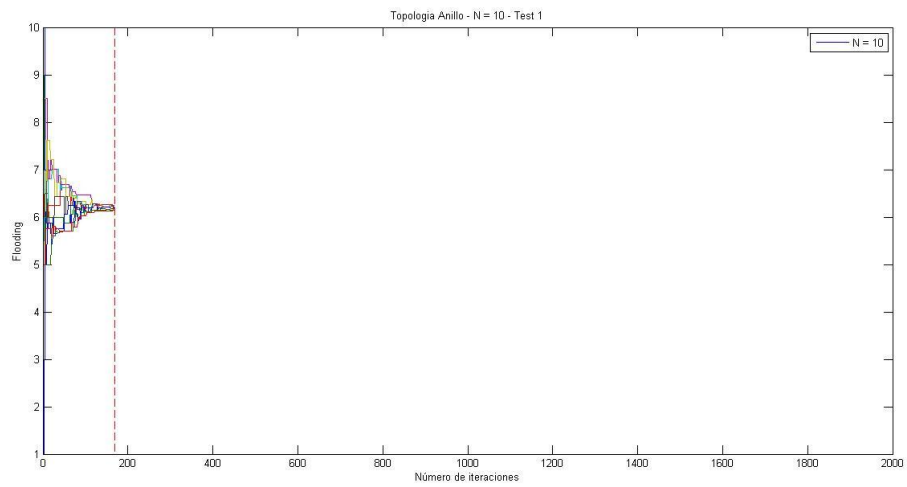
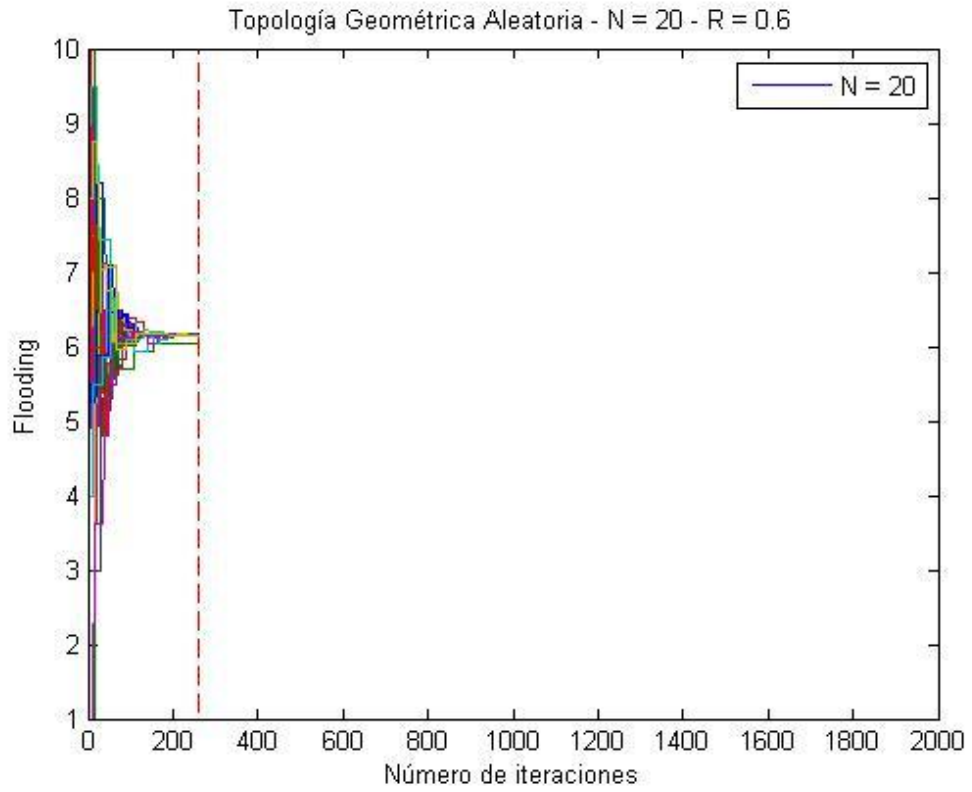


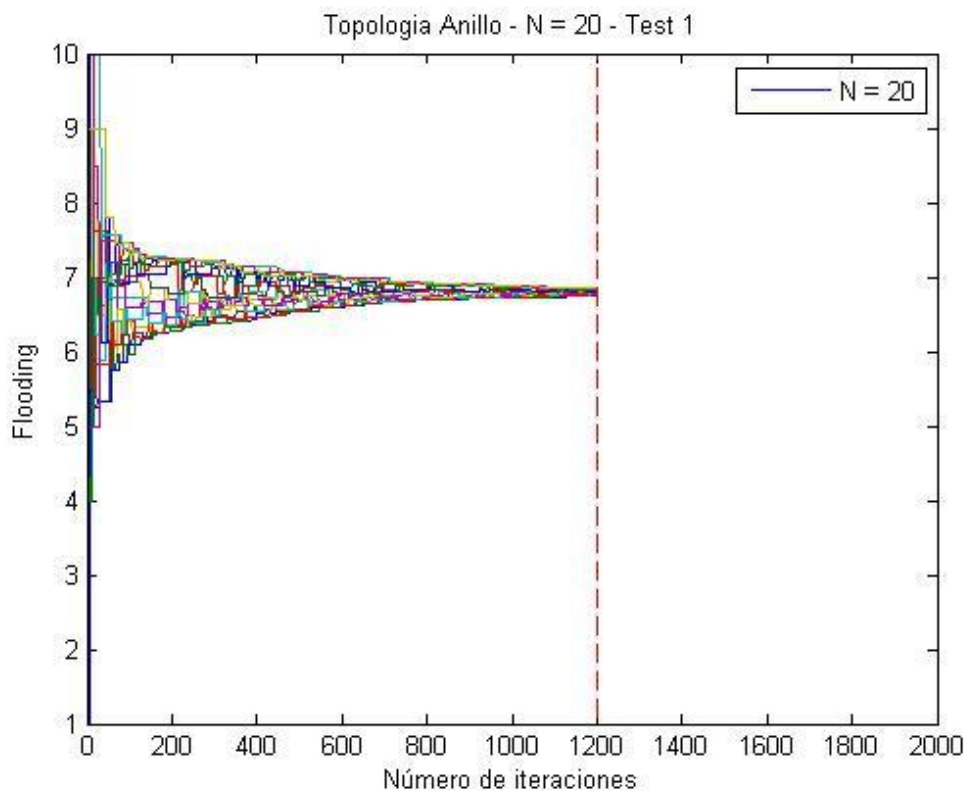
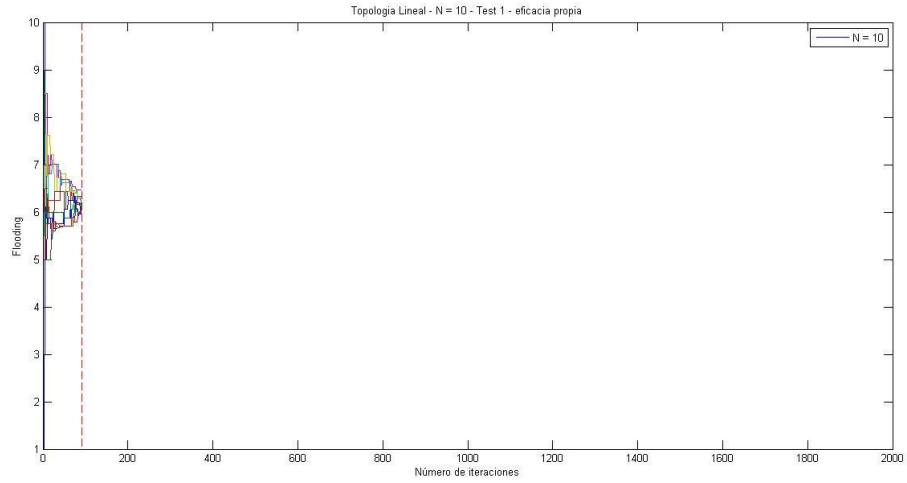


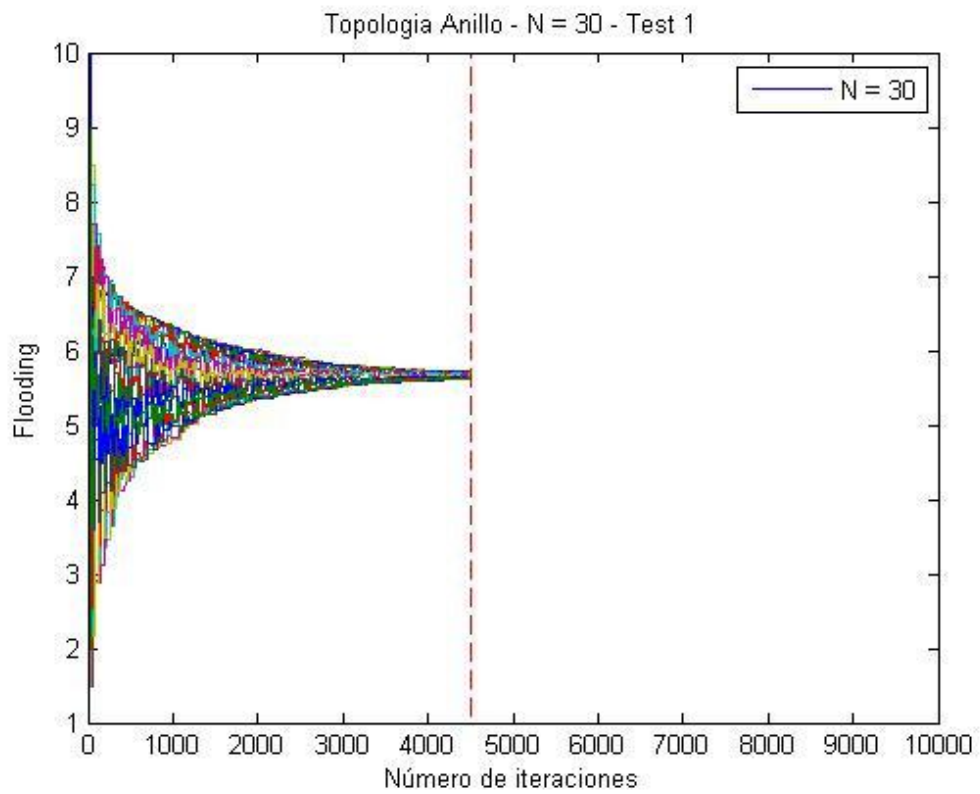
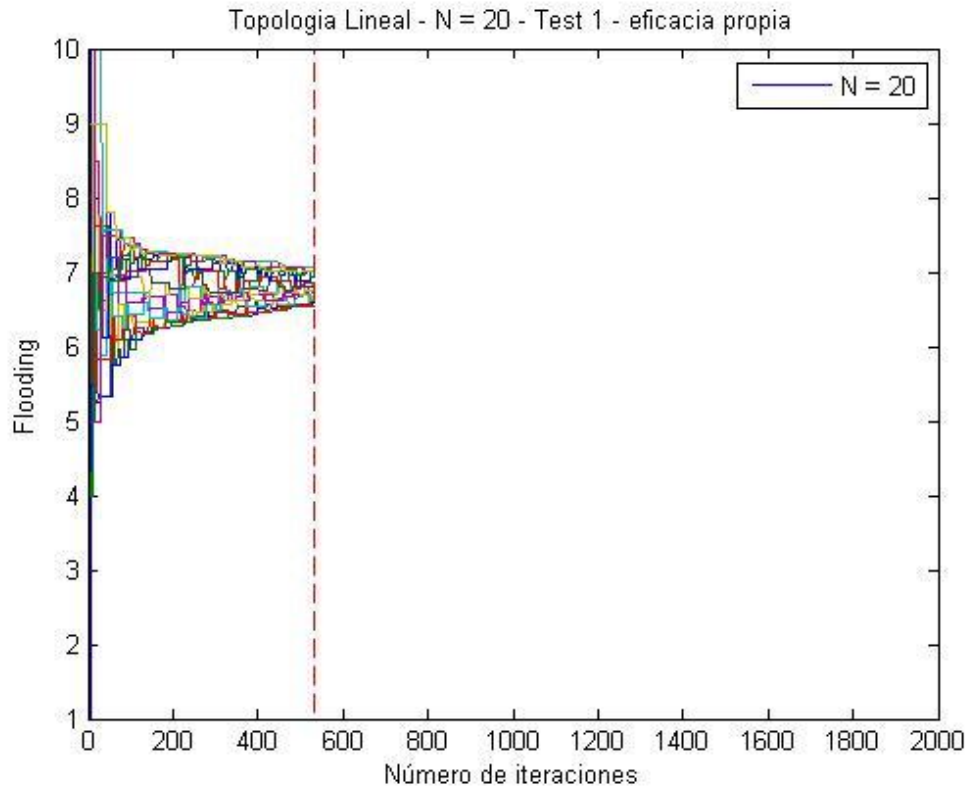


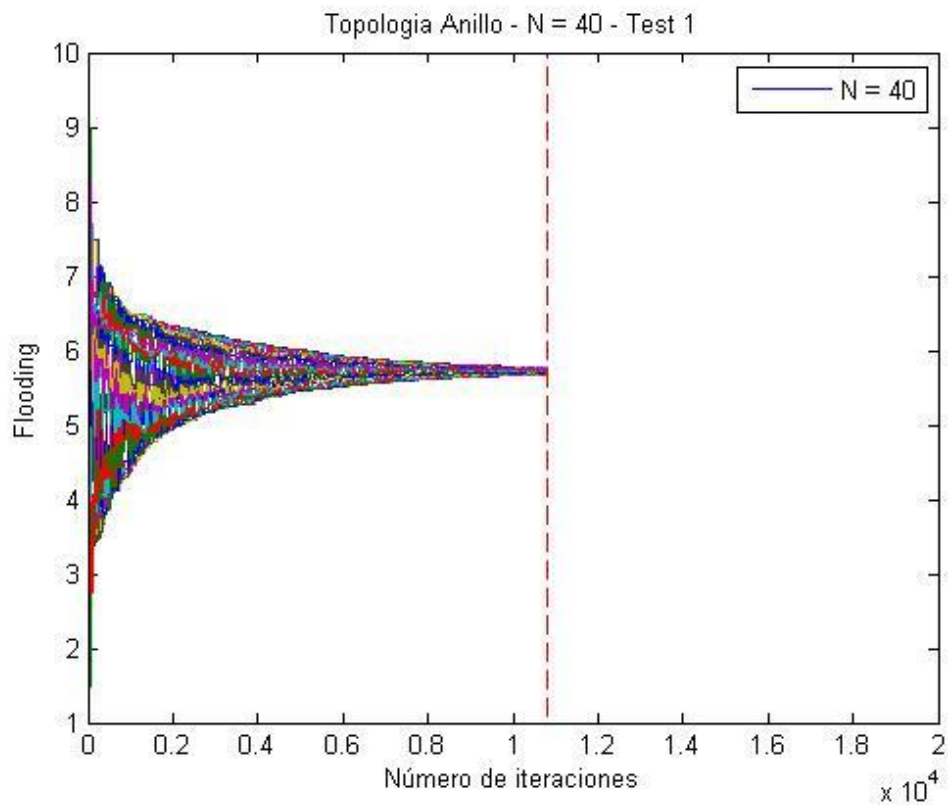
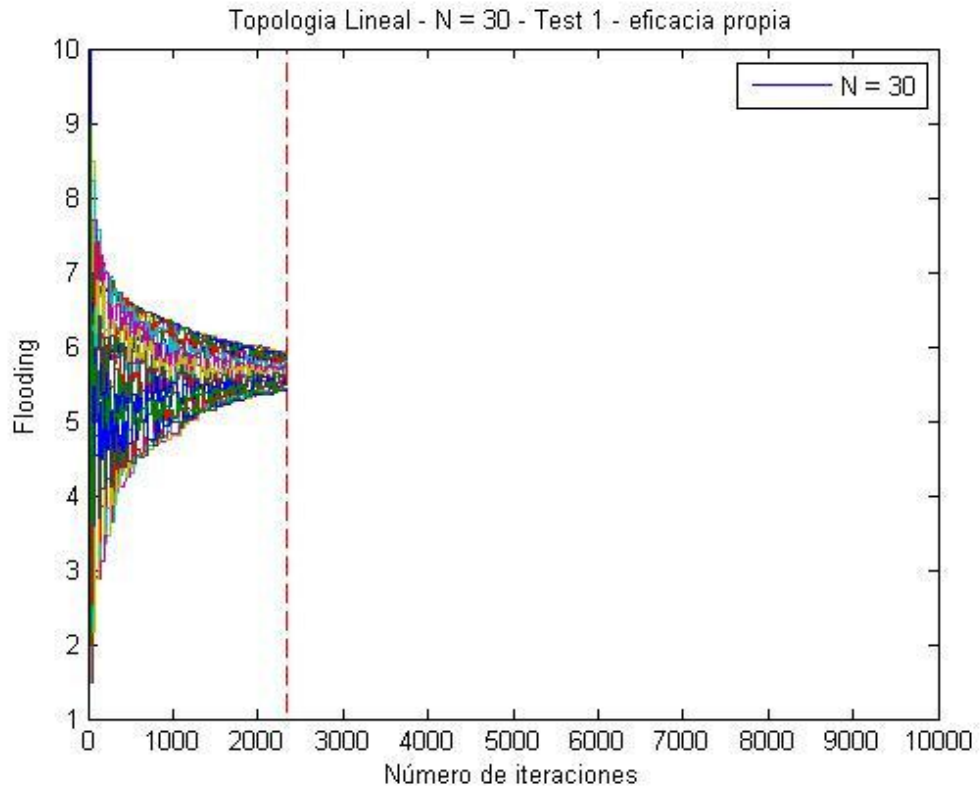


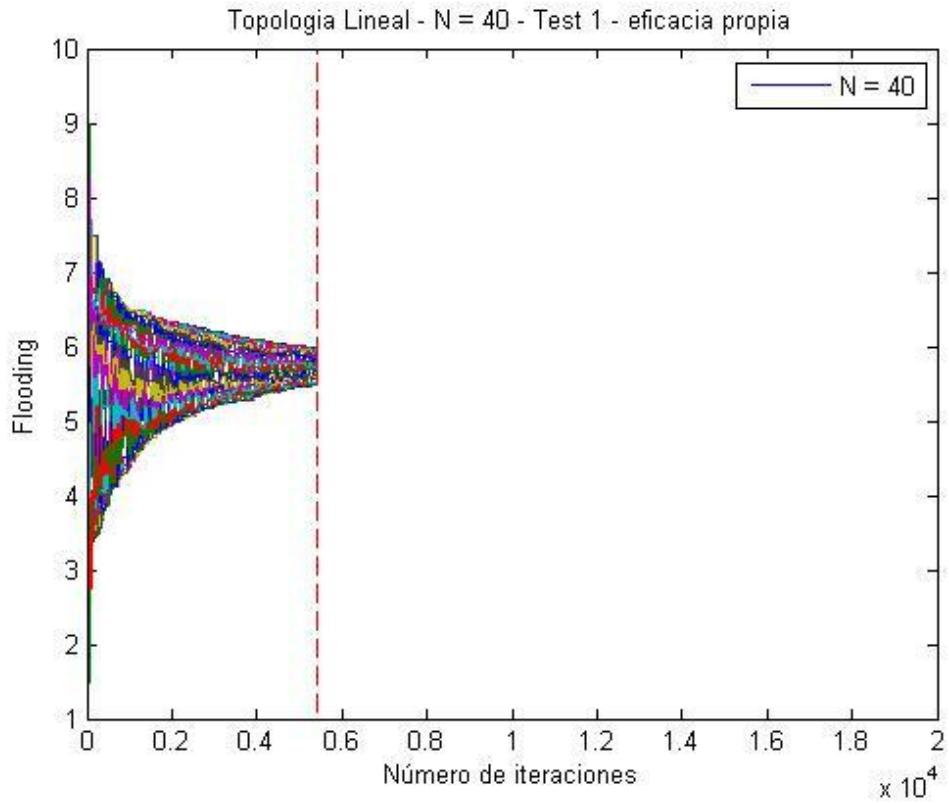












PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

**IMPLEMENTACIÓN Y EVALUACIÓN DE UNA RED INALÁMBRICA
DE SENSORES A TRAVÉS DE ALGORITMOS DE
TRANSMISIÓN BASADOS EN GOSSIPING POR RF**

ANEXO 5

PROGRAMA ARDUINO

```

// Programa para un nodo sensor en la WSN

// Autor: Martin Arturo Martínez Espinoza

Lm35DRG:

#include <Wire.h>
#include "DS3231.h"

DS3231 RTC; //Create the DS3231 object

int sensortemp = A0; // select the input mV from the LM 35
float sensorValue = 0; // variable to store the value coming from the sensor
float temp = 0; // Temp in centigredes
float temperatura;
float prom, consumo;
unsigned long time1 , time2, time3, time;
int indice = 0 ;
int i;
int flag = 0;
int randomNumber;
String inData;
int cabecera;
//char recibo[20];
//int recibo;

void setup() {
  // declare the ledPin as an OUTPUT:
  Serial.begin(9600);
  Wire.begin();
  RTC.begin();
  pinMode(sensortemp, INPUT);
  pinMode(13, OUTPUT);
}

void loop() {
  // read the value from RTC
  //RTC.convertTemperature(); //convert current temperature into registers
  //Serial.print(RTC.getTemperature()); //read registers and display the temperature
  //Serial.println("deg C");

  // read the value from the sensor:
  sensorValue = (analogRead(sensortemp))*(3300/1024);
  //Serial.println(sensorValue); //voltage in milivolts
  temp = (sensorValue)/10;
  //Serial.println("NodoA");

  Serial.print("Time1: ");
  time1 = millis();
  //prints time since program started
  Serial.println(time1);

  if (flag >= 0)

```

```

{
  Serial.print("C");
  Serial.println(temperatura);    // temp in centigredes
  flag = flag - 1;
} //Serial.println("deg C");
else
{
  // delay (50);
  Serial.print("C");
  Serial.println(temp);
}

randomNumber = random(500,1000);
delay (randomNumber);

Serial.print("Time2: ");
time2 = millis();
//prints time since program started
Serial.println(time2);
time = time2-time1;
//Serial.println(time);

//delay(500);

//recibo = Serial.read();

//if (recibo == 97) {
//  Serial.println(recibo);//}

if (Serial.available() > 0) {
  int h=Serial.available();
  // if you are getting escape -characters try h--; here

for (i=0;i<h;i++){
  inData += (char)Serial.read();
  if (inData[i] == 99)
  {
    break;}
  }
  // if you are getting escape -characters try Serial.read(); here
}
//print it out
indice = i;

cabecera = inData[i];
//  Serial.println(cabecera);

if (cabecera == 99)
{
  flag = 1;
  temperatura = (inData[i-5] - 48)*10 + (inData[i-4] - 48) + (inData[i-2] - 48)*0.1
+(inData[i-1] - 48)*0.01 ;
//  Serial.println(cabecera);
//  Serial.print("Temp recibida:");

```

```
// delay(300);
// Serial.print("C");
// Serial.println(temperatura);
// Serial.println(inData);
// prom = (temp + temperatura)*0.5;
// Serial.print("Temp prom:");
// Serial.println(prom);
}
// else
// {Serial.print("");}

inData = "";

randomNumber = random(3500,4000) ;
delay(randomNumber);
Serial.print("Time3: ");
time3 = millis();
//prints time since program started
Serial.println(time3);
time = time3-time2;
//Serial.println(time);
time = ((time2-time1)/(time3-time1));
consumo = (time*53.4+(1-time)*12.4)*3.3;
Serial.print("D");
Serial.print(consumo);
Serial.println("mW");
}
```