

**Proyecto CLCrypt**  
**Cuadernos de Laboratorio de Criptografía. Entrega nº 5. Última actualización 24/10/18**  
**Autor: Dr. Jorge Ramío Aguirre (@criptored)**  
**Prácticas con el algoritmo RSA: claves parejas**

- Software genRSA v2.1: [http://www.criptored.upm.es/software/sw\\_m001d.htm](http://www.criptored.upm.es/software/sw_m001d.htm)
- Software SAMCrypt: [http://www.criptored.upm.es/software/sw\\_m001t.htm](http://www.criptored.upm.es/software/sw_m001t.htm)
- Lectura de interés:  
<http://www.criptored.upm.es/crypt4you/temas/RSA/leccion4/leccion04.html>
- Lectura de interés:  
[https://en.wikipedia.org/wiki/Safe\\_prime](https://en.wikipedia.org/wiki/Safe_prime)

**Objetivos:**

1. Observar las claves privadas parejas y las claves públicas parejas de una clave RSA.
2. Comprobar que dichas claves realizan la misma función que sus claves directas.
3. Comprobar que con el uso de primos seguros se minimiza la cantidad de estas claves parejas.
4. Observar que, además de lo anterior, en función de los valores de los primos p y q, se forman anillos de cifra que derivan en algunos casos en otros números que realizan la misma función que la clave privada.
5. Comprobar que este fenómeno no se traduce en una vulnerabilidad en RSA.

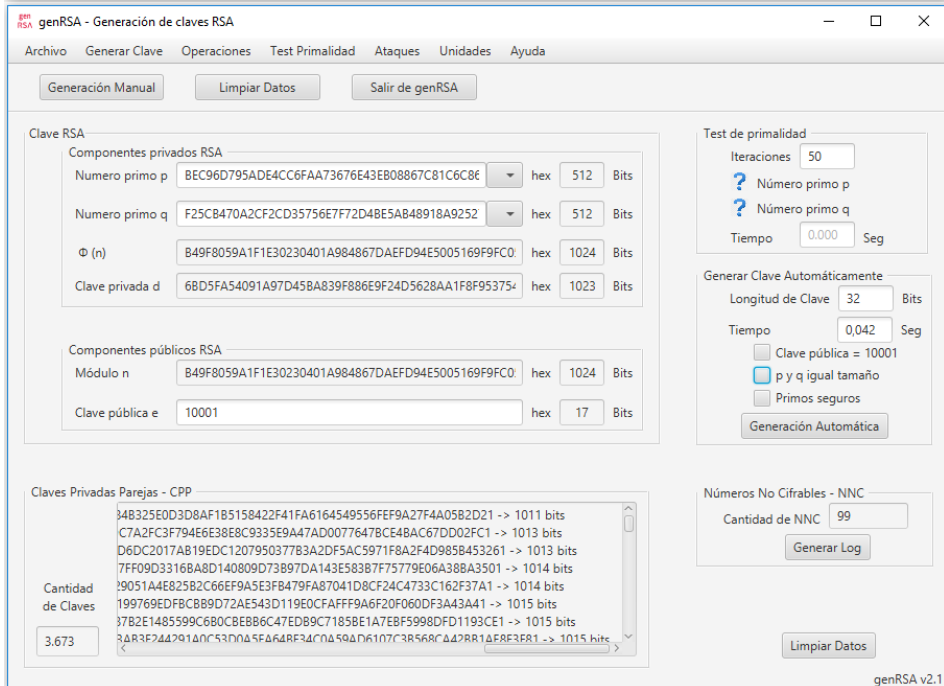
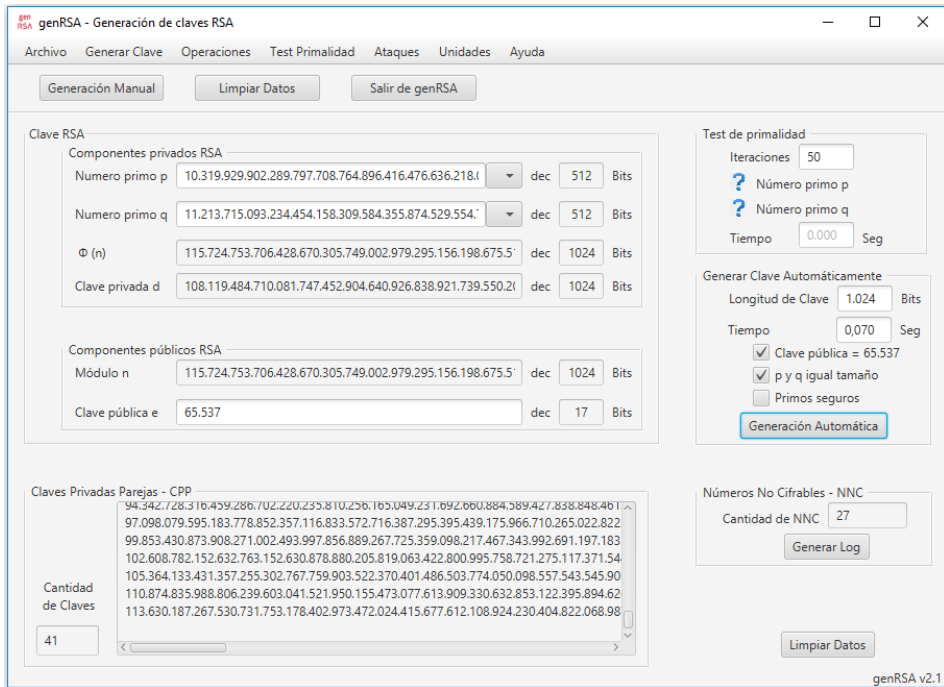
**I. Claves privadas parejas en RSA**

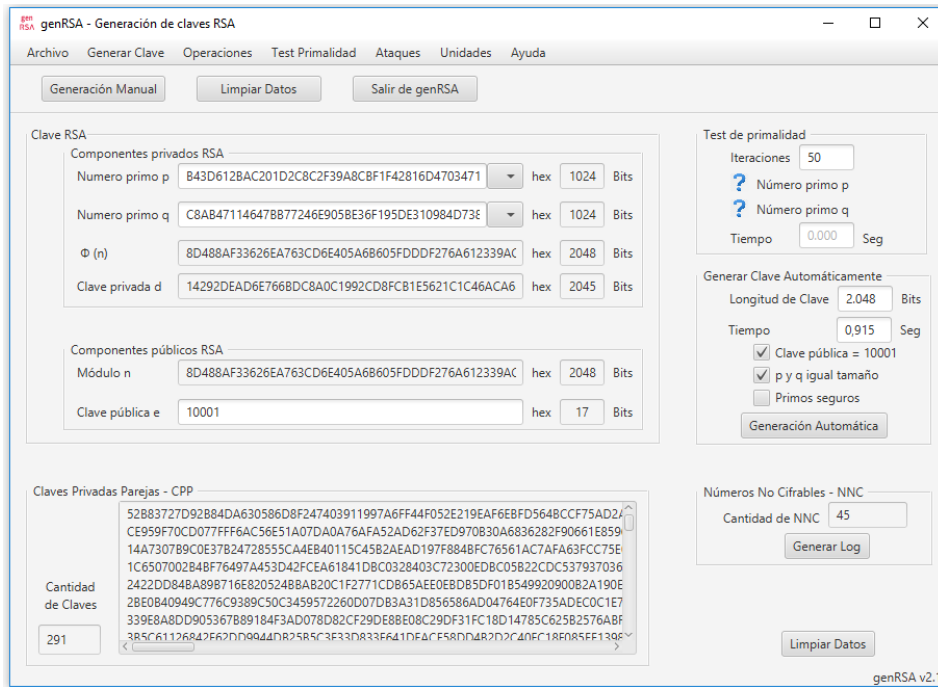
**Ejercicio 1)**

- 1.1. Con genRSA v2.1 genera de forma Manual una clave RSA con p = 31, q = 101, e = 7 en formato decimal.
- 1.2. Observa que se obtienen 10 claves privadas parejas CPP y que la separación entre ellas es el valor 300 como se aprecia: 43, 343, 643, 943, 1.243, 1.543, 1.843, 2.443, 2.743, 3.043, excepto en la zona donde se encuentra la clave privada 2.143.
- 1.3. Comprueba con genRSA v2.1 que en Operaciones al cifrar el valor 2.145 se obtiene el criptograma 2.238 y que al descifrarlo con cualquiera de esas claves privadas, se recupera el mensaje original 2.145.
- 1.4. Comprueba una de estas operaciones con la calculadora de Windows, por ejemplo:  $2.238^{43} \bmod 3.131 = 2.145$ .
- 1.5. Genera de forma Automática, con p y q de igual tamaño y Clave pública = 65.537 (no actives primos seguros) varias claves de 1.024 bits en decimal, hasta obtener alguna con más de 25 CPP.
- 1.6. Observa el tamaño de esas claves privadas parejas (ordenadas de menor a mayor), usando el *scrollbar* horizontal y vertical de esa ventana.
- 1.7. Cambia las Unidades a hexadecimal y genera de forma Manual esta clave:  
p =  
BEC96D795ADE4CC6FAA73676E43EB08867C81C6C864938729FBDE9834A64370F845D9  
8460D76718B0E43DA25FF6DA238C84D92F426B9F60D511E5A6BDCCF92E1  
q =  
F25CB470A2CF2CD35756E7F72D4BE5AB48918A92527AF42371E88CF376B3DB37584BA  
084E09F1FDAA53DBBE13BBAC772A8549A5917D6E15AD1BC3A726898FFEF  
e = 10001
- 1.8. Observa la cantidad de CPP que se obtienen y vuelve a usar el *scrollbar*. Guarda esta clave como clave1024\_muchasCPP.html.

- 1.9. Aunque la clave RSA anterior sea de 1.024 bits y hoy se usen claves de 2.048 bits, ¿podría ser ésta una clave real de un servidor web seguro de hace unos 5 o más años?
- 1.10. Abre el archivo clave1024\_muchasCPP.html y comprueba que la clave privada pareja de menor tamaño tiene 1.011 bits y la mayor 1.024 bits.
- 1.11. ¿Por qué motivo todas las CPP se encuentran tan cerca del módulo de cifra?
- 1.12. ¿El hecho de que esta clave tenga más de 3.600 CPP, la convierte en débil? ¿Por qué sí o por qué no? Justifica tu respuesta.
- 1.13. Genera de forma Automática en hexadecimal una clave RSA de 2.048 bits real, con Clave pública  $e = 10001$ ,  $p$  y  $q$  de 1.024 bits (sin primos seguros), y que tenga más de 50 CPP.
- 1.14. En un próximo cuaderno de laboratorio vamos a generar claves RSA con OpenSSL y vas a comprobar que ese programa no muestra las CPP. ¿Por qué crees que no las muestra?

### Comprueba tu trabajo:





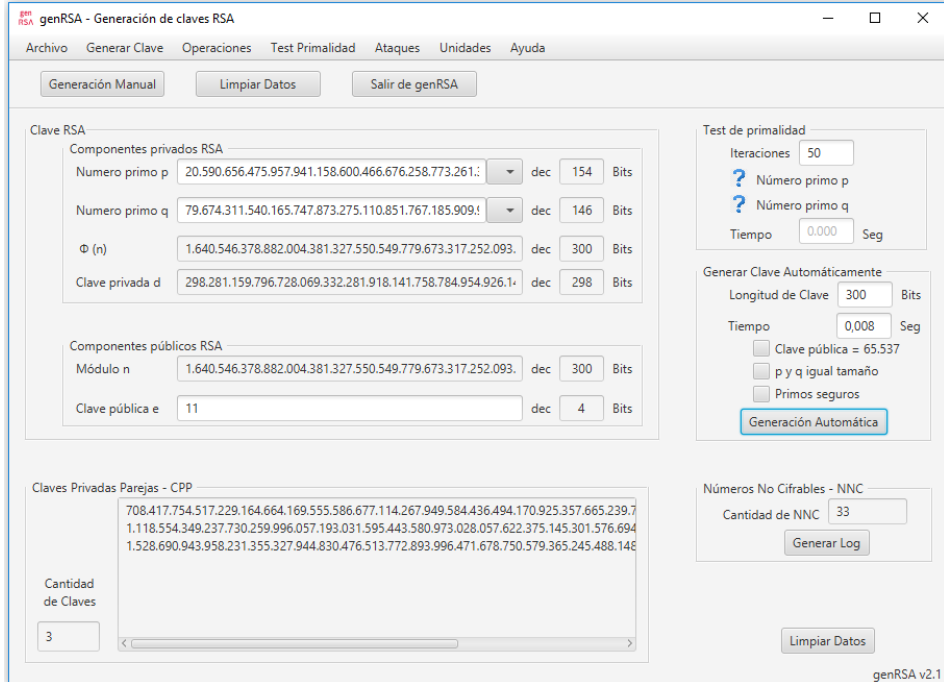
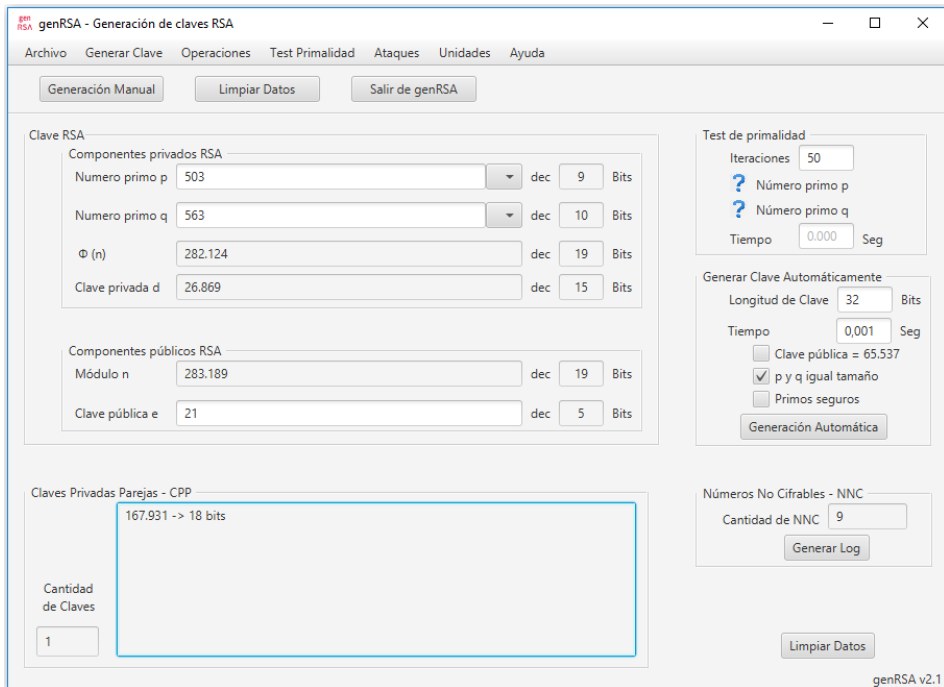
Observación de las Claves Privadas Parejas en RSA.

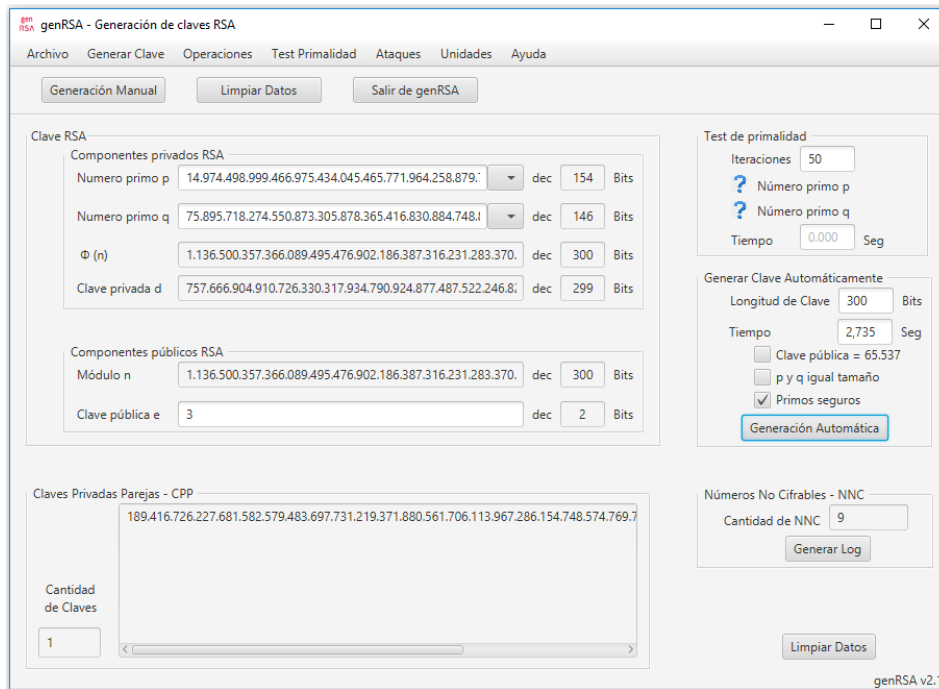
## II. Minimizando las claves privadas parejas en RSA

### Ejercicio 2)

- 2.1. Con genRSA v2.1 genera de forma Manual una clave RSA decimal eligiendo desde la misma ventana los valores de p y q como primos de 3 dígitos y pon como clave pública un número impar de 2 dígitos, e.g. 11, 37, 49, 63, etc.
- 2.2. Como los valores que propone genRSA v2.1 son primos seguros, comprueba que la cantidad de CPP siempre será 1, la menor posible.
- 2.3. Genera de forma Automática varias claves de 300 bits, desactivando todas las opciones de la Generación Automática.
- 2.4. Comprueba que en este caso el valor de Clave pública e será siempre el número impar más bajo posible y que el número de CPP cambia de una a otra clave.
- 2.5. ¿Por qué siempre la clave pública, o la clave privada, es un número impar? ¿Por qué no puede ser un número par?
- 2.6. Repite el apartado 2.3 activando ahora solamente la opción Primos seguros.

### Comprueba tu trabajo:



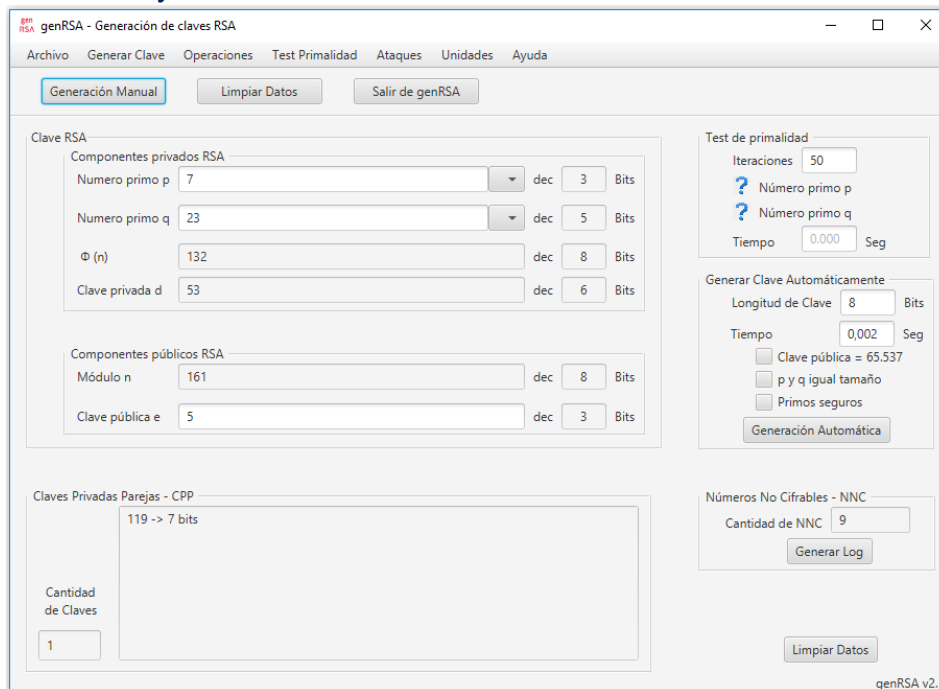


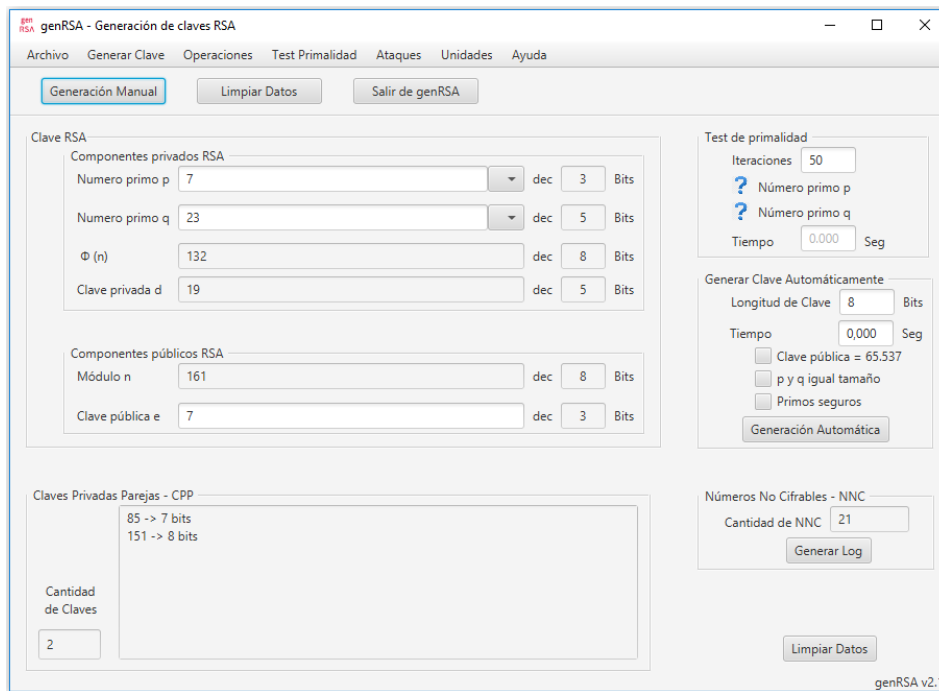
Claves Privadas Parejas mínimas en RSA con primos seguros.

### III. Los primos seguros no garantizan CPP = 1 Ejercicio 3)

- 3.1. Genera de forma Manual una clave RSA decimal de 8 bits con  $p = 7$ ,  $q = 23$ , ambos primos seguros ( $7 = 2 \times 3 + 1$  y  $23 = 2 \times 11 + 1$ ) y  $e = 5$ , el menor posible. Observa que el número de claves privadas parejas es 1.
- 3.2. Usa ahora  $e = 7, 19, 23, 35, 37, 53$ . Observa que a pesar de usar primos seguros, no obtenemos el número mínimo de CPP, aunque sí un valor muy bajo, 2.

#### Comprueba tu trabajo:





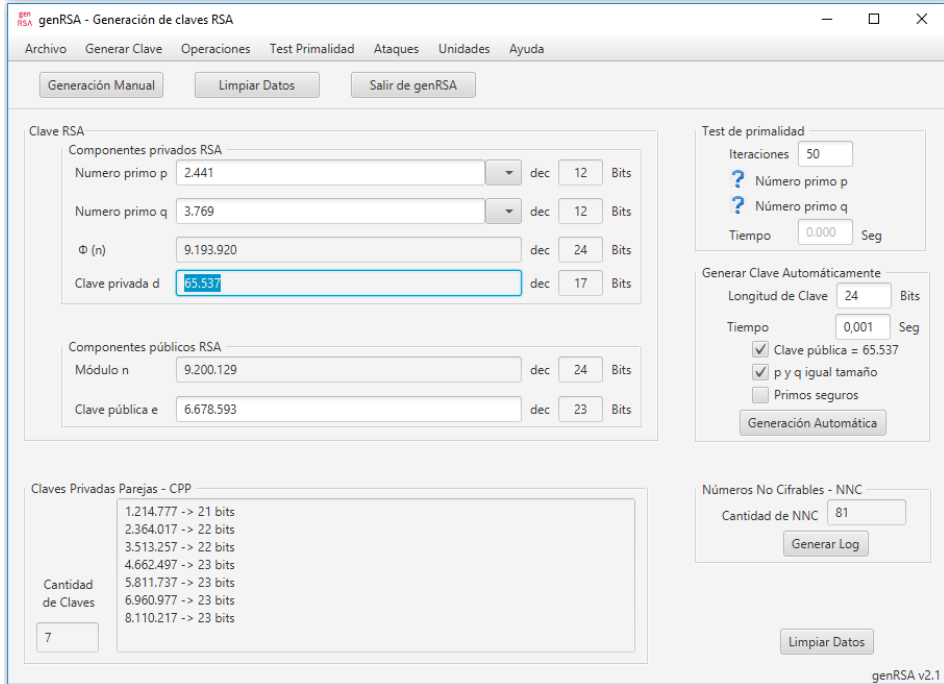
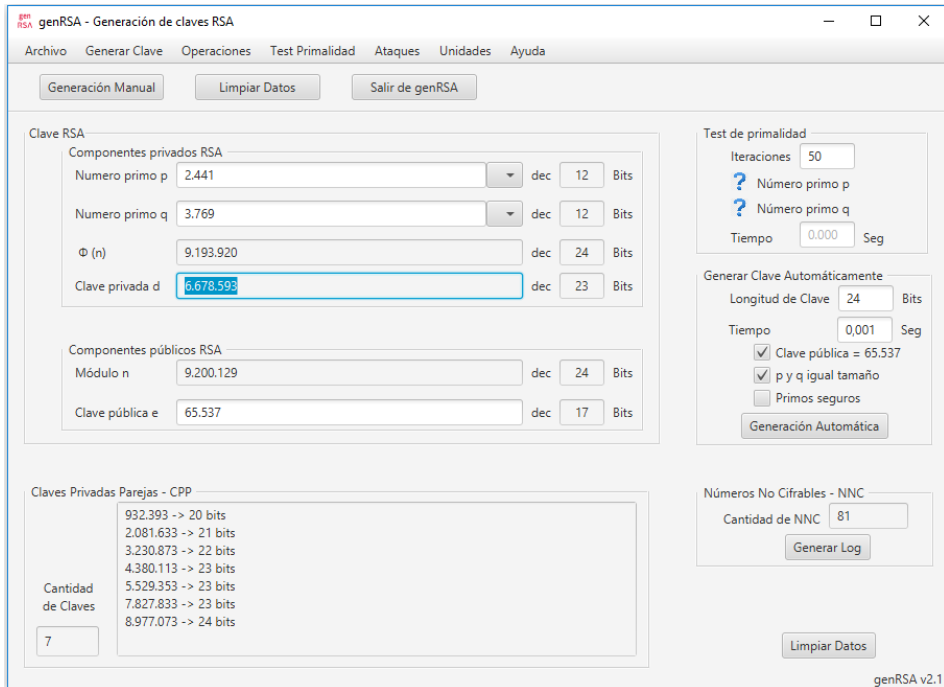
Primos seguros no significa necesariamente CPP = 1.

#### IV. Claves públicas parejas en RSA

##### Ejercicio 4)

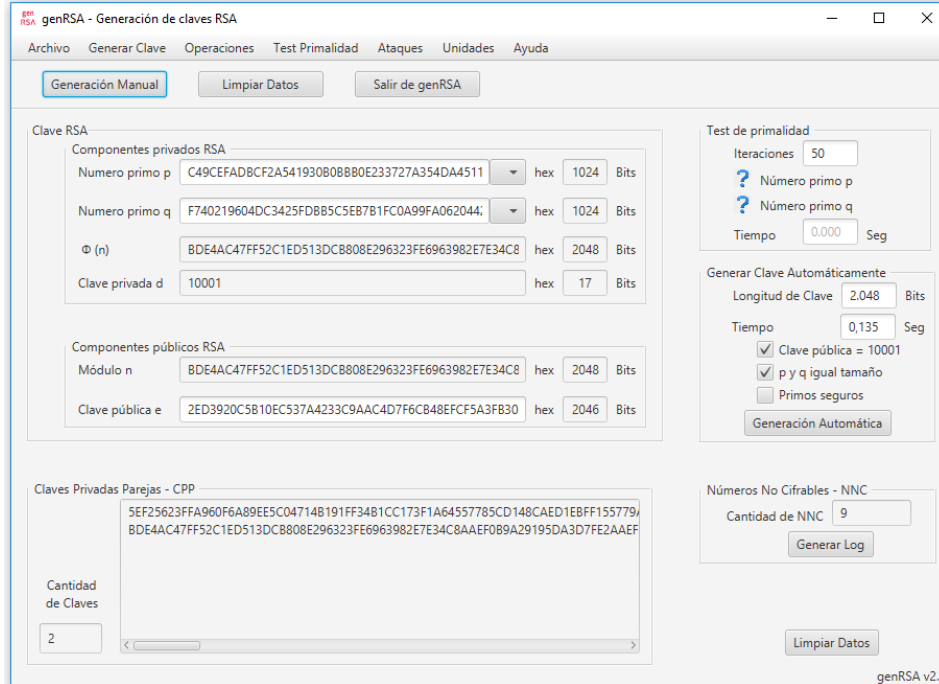
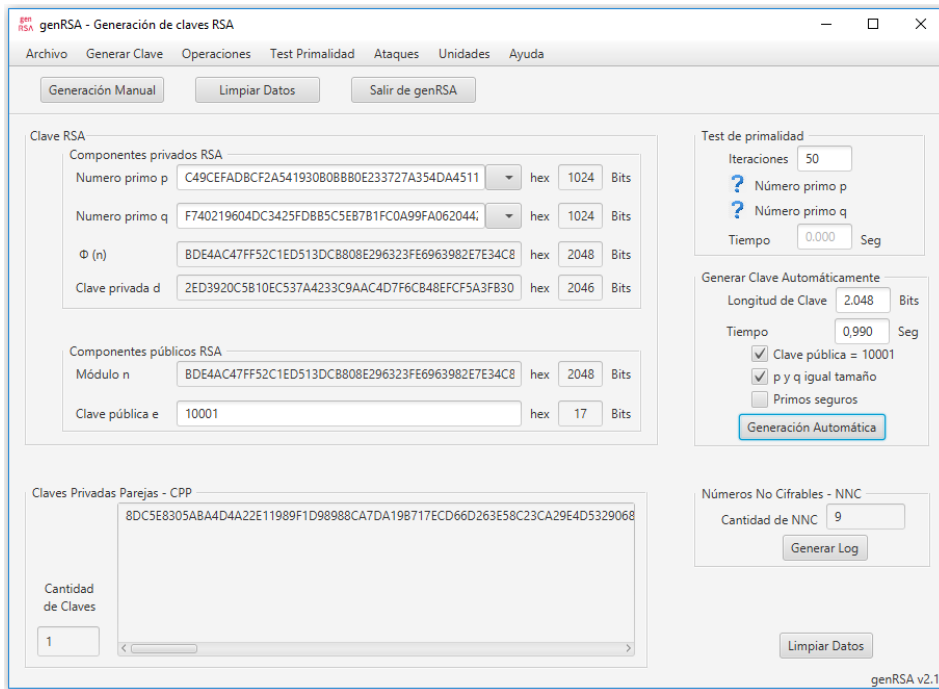
- 4.1. Genera de forma Manual una clave RSA decimal con  $p = 2.441$ ,  $q = 3.769$ ,  $e = 65.537$ .
- 4.2. Copia al portapapeles el valor de la clave privada 6.678.593 y pega ese valor en la casilla de la Clave pública e. Hecho esto, haz clic en Generación Manual.
- 4.3. Observa que las claves pública y privada han cambiado de lugar. Los números que se observan ahora como “Claves Privadas Parejas”, corresponderán lógicamente a las Claves Públicas Parejas.
- 4.4. Repite el ejercicio, ahora generando de forma Automática varias claves reales en formato hexadecimal de 2.048 bits, con primos  $p$  y  $q$  de igual tamaño y Clave pública 10001, hasta que tengas una clave con solo 1 Clave Privada Pareja.
- 4.5. Selecciona el valor de la Clave privada  $d$  y cópiala en el portapapeles. Pega ese número en la casilla de la Clave pública e.
- 4.6. Hecho esto, vuelve generar la clave pero ahora Manualmente y observa lo que sucede con las claves pública y privada.
- 4.7. Observa que en muchos casos, la cantidad de Claves Públicas Parejas aumenta en una unidad con respecto a las Claves Privadas Parejas.
- 4.8. ¿Podría ser un problema el hecho de que se pueda comprobar una firma digital RSA con un valor de clave pública diferente al estándar, el número 4 de Fermat?
- 4.9. Visto lo anterior, ¿puedes justificar por qué no se permite que el usuario elija un valor aleatorio de clave pública?
- 4.10. Si se permitiese que el usuario al generar su clave RSA pudiera elegir un valor aleatorio para su Clave pública, ¿qué limitaciones le pondrías?

##### Comprueba tu trabajo:



Arriba clave RSA, abajo clave RSA cambiando de e por d.  
Claves Públicas Parejas de una clave RSA de 24 bits.





Arriba clave RSA, abajo clave RSA cambiando de e por d.  
Claves Públicas Parejas de una clave RSA de 2.048 bits.

## V. Generación de anillos en la cifra RSA y nuevas claves válidas Ejercicio 5)

- 5.1. Con gen RSA v2.1 genera de forma Manual esta clave RSA decimal con  $p = *$ ,  $q = *$ ,  $e = *$ .
- 5.2. Con SAMCrypt cifra el mensaje 13 **BUSCAR TRABAJO HECHO EN CASA**
- 5.3. Descifra el criptograma obtenido con todos los valores posibles de la clave d, incluso con los valores pares aunque se sepa que ni la clave pública ni la clave privada en RSA pueden ser números pares.
- 5.4. Etc.

**Comprueba tu trabajo:**