

CAPÍTULO 4: FASE DE PRUEBAS

En este capítulo veremos las múltiples pruebas realizadas con el algoritmo de extracción de *fingerprints*.

Primeramente mostraremos con qué hardware y con qué software se ha trabajado para las múltiples pruebas realizadas. Será importante conocerlos, sobre todo, de cara a tenerlo en cuenta a la hora de analizar el tiempo de cómputo en la extracción, pues este tiempo será mayor o menor dependiendo del procesador empleado.

En un segundo apartado del capítulo veremos las pruebas, con sus correspondientes resultados, al ir variando uno, y sólo uno, de cada una de las características que definen la técnica de extracción de *fingerprints* empleada. Sobre cada una de las características obtendremos un valor que optimiza el algoritmo manteniendo un compromiso entre las probabilidades de error implicadas en la detección y la eficiencia computacional.

En el tercer apartado se tratarán pruebas en las que se combinan variaciones en diferentes características al mismo tiempo. Este paso en la fase de pruebas se hará imprescindible, pues una vez que tengamos los valores óptimos para cada una de las características que me definen el algoritmo de extracción, habrá que verificar con qué valores realmente nos quedamos, ya que utilizar los valores óptimos obtenidos para cada uno de los parámetros nos lleva a unas probabilidades de error muy elevadas e incluso algunos no son compatibles entre sí.

En el último apartado del capítulo mostraremos los resultados de una serie de pruebas de robustez que se realizarán sobre el algoritmo implementado con los valores óptimos (calculados en el apartado 3). Una de ellas consistirá en hallar la relación señal a ruido (SNR) mínima necesaria a partir de la cual el sistema funcionará correctamente. El resto de pruebas consistirán en someter a la señal de audio a un procesado que le produzca un cierto tipo de distorsión antes de ser procesada por el sistema.

4.1. HARDWARE Y SOFTWARE

4.1.1. SISTEMA

En cuanto al hardware, para la realización del proyecto se ha utilizado un ordenador portátil INTEL CENTRINO DUO a 1.66 GHz, con 1 GB de memoria RAM DDR como refleja la ilustración 1.

Este es el sistema en el que se han realizado todas las pruebas, con sus respectivas medidas de tiempo de computación que son descritas en los apartados relacionados con el capítulo 4, "Fase de pruebas".

Es muy importante la velocidad del micro en la aplicación para la que se pretende que sirva este algoritmo, debido a su elevado coste computacional.

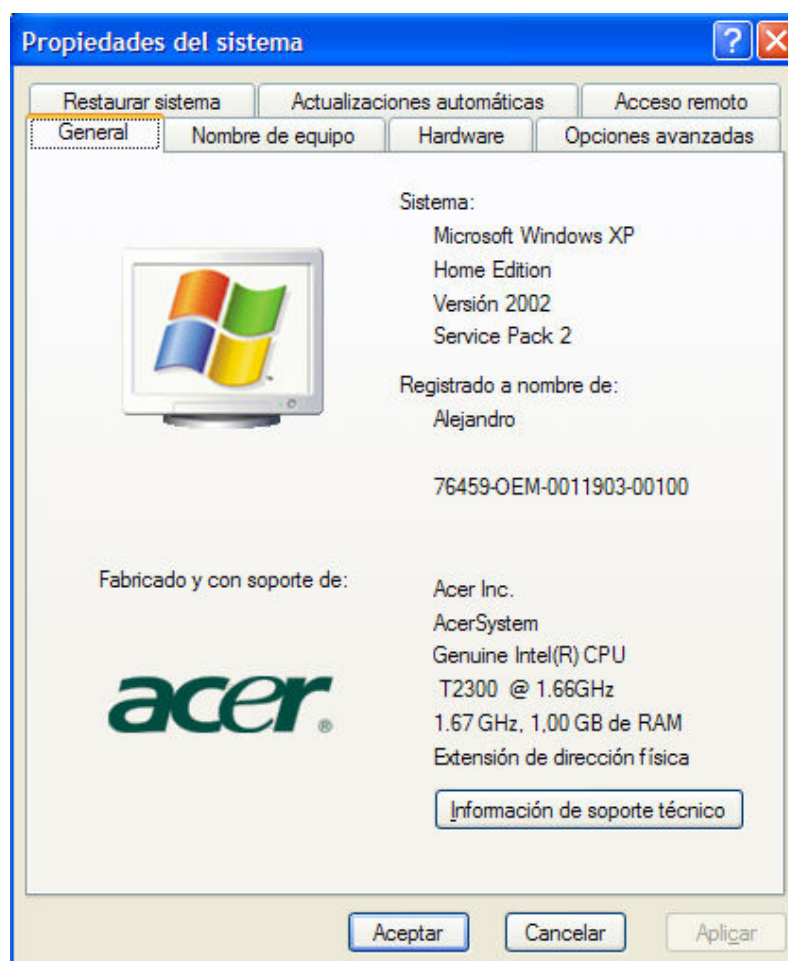


Ilustración 1. Características del sistema utilizado

4.1.2. TARJETA CAPTURADORA

Para la captura y grabación de los archivos de audio utilizados para el análisis y optimización del sistema se hará uso de una capturadora-sintonizadora de TV, TDT y FM, en concreto del modelo "AverTV Hybrid + FM PCI" de la marca **Aver Media**. Esta tarjeta nos permite capturar la señal de radio en muchos formatos; entre ellos nos ha sido útil el formato PCM mono a 8 Khz. con 8 bits por muestra.

4.1.3. SISTEMA OPERATIVO

El sistema operativo utilizado ha sido el Windows XP Profesional (Ilustración 2), sobre el cual trabajan las dos aplicaciones utilizadas para el desarrollo del proyecto: el Matlab 7.0 y el Adobe Audition 2.0.

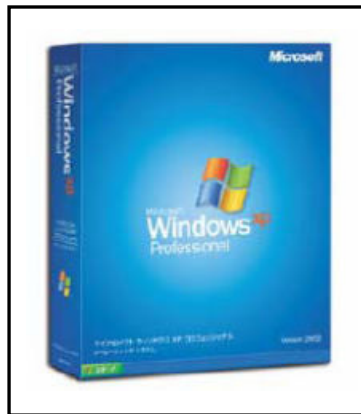


Ilustración 2. Sistema Operativo Windows XP

4.1.4. MATLAB 7.0

MatLab es un lenguaje de alto nivel para programar cálculos técnicos. Integra cálculo, visualización y programación en un entorno sencillo. Para nosotros consistirá en la herramienta principal utilizada en el desarrollo del proyecto para hacer todos los análisis y pruebas sobre el algoritmo basado en *fingerprints* implementado.

Se trata de un sistema interactivo cuyo elemento básico es una matriz, lo que nos permite resolver los problemas técnicos formulados en términos de matrices o vectores de forma muy rápida y eficiente. De hecho el nombre proviene de "matriz laboratory".

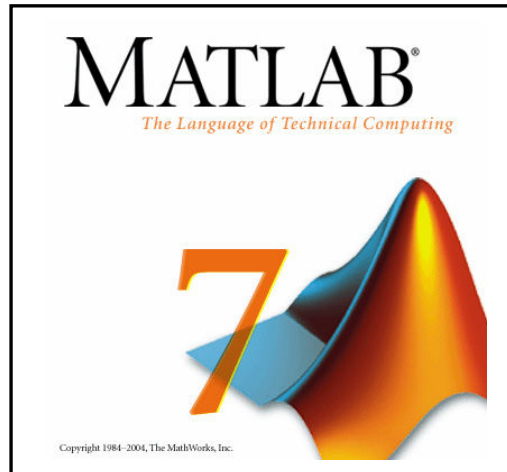


Ilustración 3. Matlab 7.0

4.1.5. ADOBE AUDITION 2.0

Este software nos permite realizar diferentes procesados en las señales de audio que conforman nuestra base de datos para llevar a cabo las diversas pruebas de robustez.

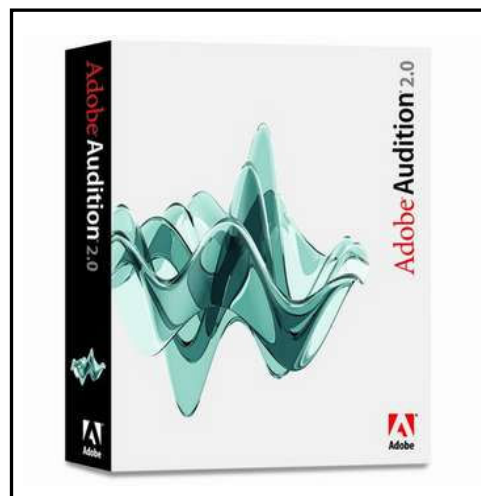


Ilustración 4. Adobe Audition 2.0

4.2. PRUEBAS CON UNA CARACTERÍSTICA

Comentar que, para realizar las diferentes pruebas, creamos la función **"fingerprint_actualizable"** (véase Anexo 2.16) que implementa el algoritmo de extracción de *fingerprints* con unas características u otras según los valores que le demos a sus parámetros de entrada. Las características que podemos variar en el algoritmo de extracción son las siguientes:

- número de muestras por trama,
- factor de overlap,
- número de bandas en que dividimos cada trama,
- frecuencia inferior para la división en bandas,
- frecuencia superior para la división en bandas,
- tiempo de detección de la señal de entrada.

De estas 6 características, las 5 primeras están relacionadas directamente con el cálculo de un *fingerprint* y, por tanto, son los parámetros de entrada de la función **fingerprint_actualizable**. Por otra parte, de esos 5 parámetros, serán objeto de estudio 3: el número de muestras por trama, el factor de overlap y el número de bandas en que dividimos cada trama. Así pues no vamos a analizar lo que ocurre al variar las frecuencias inferior y superior escogidas para hacer la división en bandas en el espectro de cada trama; esto es porque la influencia en el tiempo de cómputo de un *fingerprint* es prácticamente irrelevante y, sin embargo, sí puede influir negativamente en las probabilidades de error de nuestro sistema. Por tanto, para todas las pruebas realizadas las frecuencias inferior y superior van a ser respectivamente: 300 Hz y 2000 Hz, que es el rango espectral más relevante para el sistema de audición humano.

Veamos los resultados obtenidos con las diferentes pruebas realizadas, cada una de las cuales varían el valor de una, y sólo una, característica del algoritmo de extracción de *fingerprints*.

□ **Característica analizada: Número de muestras por trama (N)**

El objetivo de esta prueba es analizar hasta que valor máximo podemos aumentar el número de muestras por trama, es decir, el tamaño de la ventana de Hanning, consiguiendo un funcionamiento correcto del algoritmo.

El hecho de centrarnos en aumentar el tamaño de esta ventana, y no en disminuirlo, es porque al aumentar la ventana, el número de *sub-fingerprints*

por cada 3 segundos de anuncio detectado será menor y, por tanto, disminuirá el tiempo de computación.

Esta prueba está implementada en el código de Matlab "**prueba_N.m**" (véase Anexo 2.17). Comentemos en qué consiste este código de forma general.

Lo primero que tiene son las características que van a definir el algoritmo de extracción de *fingerprints* utilizado, salvo la variable que es objeto de análisis en esta prueba. Estas características se resumen en el siguiente cuadro:

<i>CARACTERÍSTICAS DE LA PRUEBA</i>	
Número de muestras por trama	<i>Variable</i>
Factor de overlap	31/32
Número de bandas	32
Frecuencia inferior	300 Hz
Frecuencia superior	2000 Hz
Tiempo de detección	3 segundos

Tabla 2. Característica de la prueba "variación del número de muestras por trama"

A continuación abrimos un fichero para almacenar los resultados; para esto implementamos la siguiente línea de código:

```
fi=fopen('p_N_2960_1000_23960','w');
```

Una vez tenemos el manejador de fichero, "fi", pasamos al núcleo del código, donde se irá modificando la característica que queremos analizar. Nos basamos en 15 anuncios diferentes cogidos de la base de anuncios de que disponemos; 4 de estos 15 anuncios son versiones diferentes del mismo anuncio, el cual es tomado como anuncio base para simular los 3 segundos de detección de la señal de radio de entrada a nuestro sistema. Por tanto, la idea es tener 4 anuncios para estudiar la probabilidad de no alarma y 11 anuncios para estudiar la probabilidad de falsa alarma.

Veamos qué se hace en esta parte central del código mediante el siguiente diagrama de flujo representado en la figura 7:

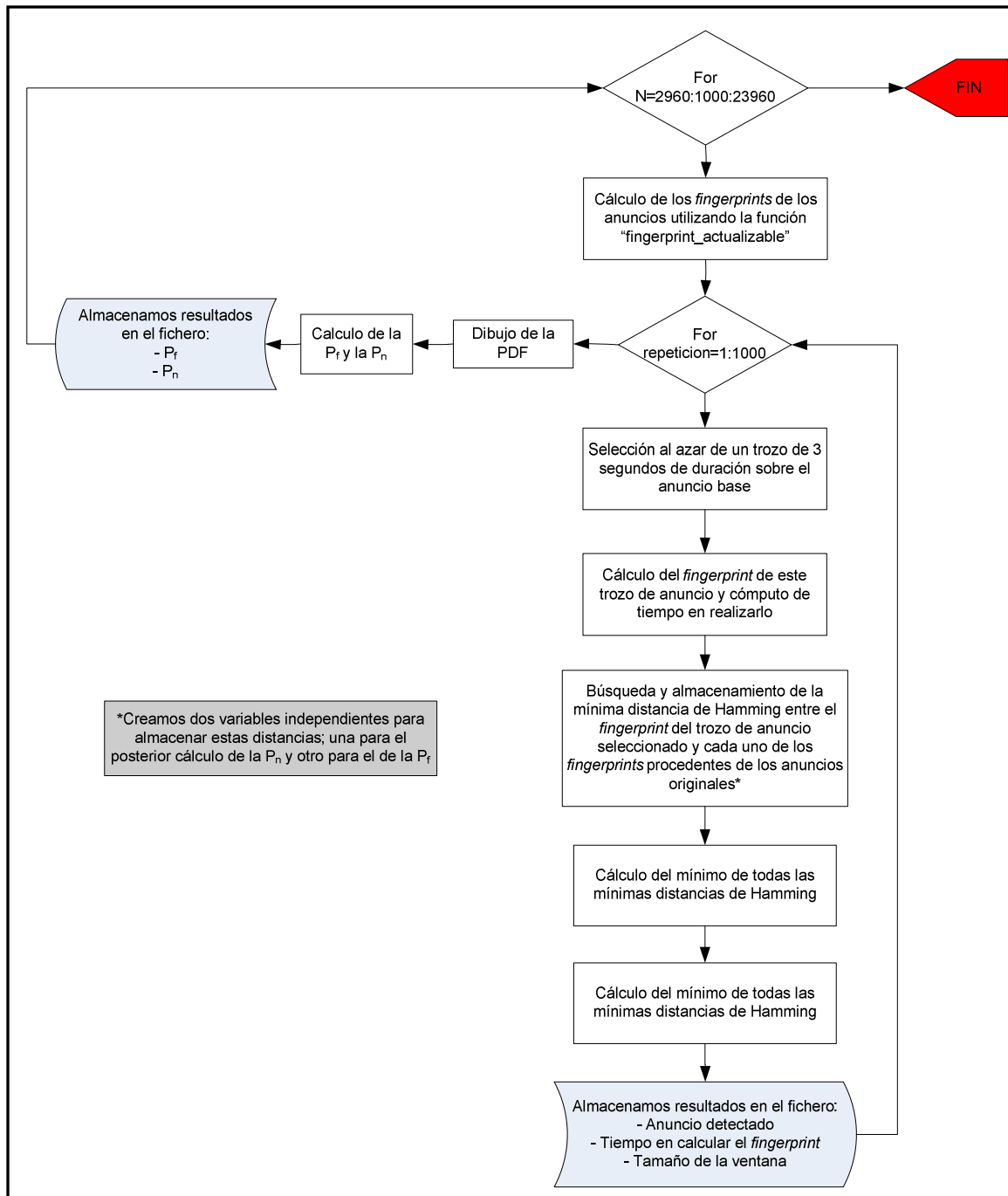
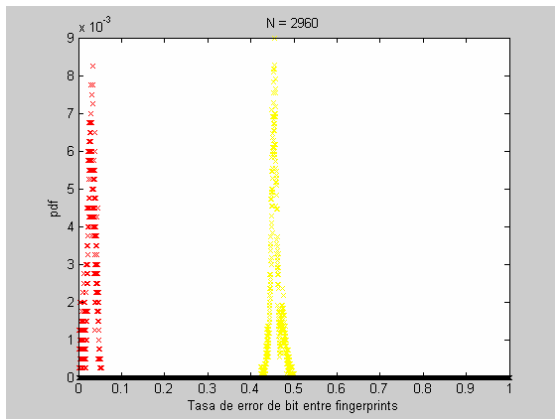


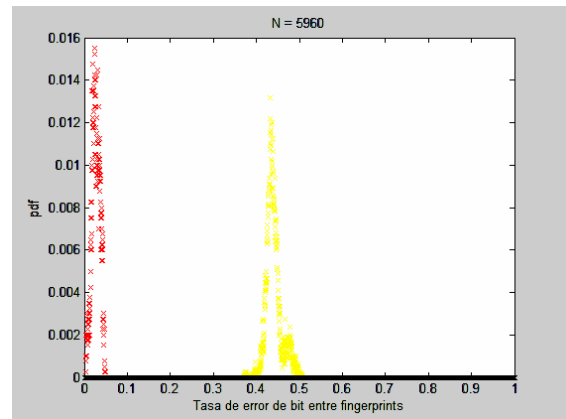
Figura 7. Diagrama de flujo del núcleo del código "prueba_N.m"

Veamos los resultados obtenidos más relevantes al ejecutar el código, en el cual variamos el parámetro objeto de análisis (el número de muestras por trama) desde 2960 a 23960 con saltos de 1000.

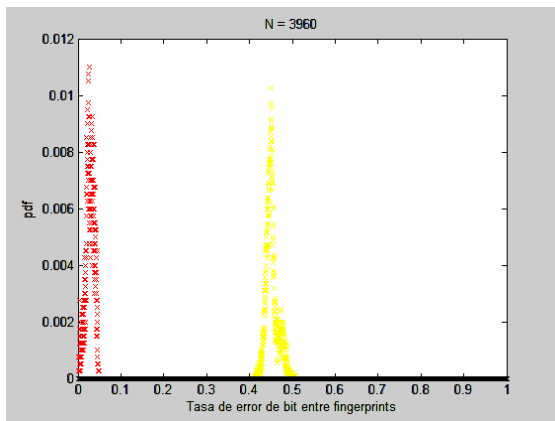
En primer lugar vamos a ver cómo evoluciona la función densidad de probabilidad experimental según el tamaño de la ventana de Hanning. Para ello mostramos las gráficas resultados de la prueba:



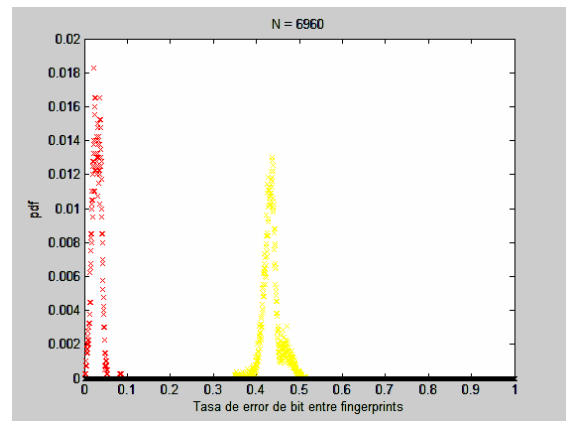
Gráfica 5. PDF Vs BER (N = 2960)



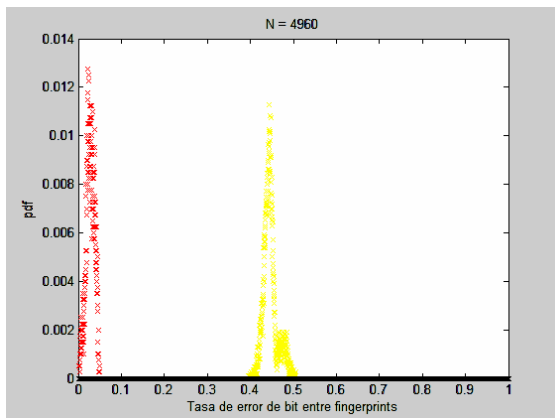
Gráfica 8. PDF Vs BER (N = 5960)



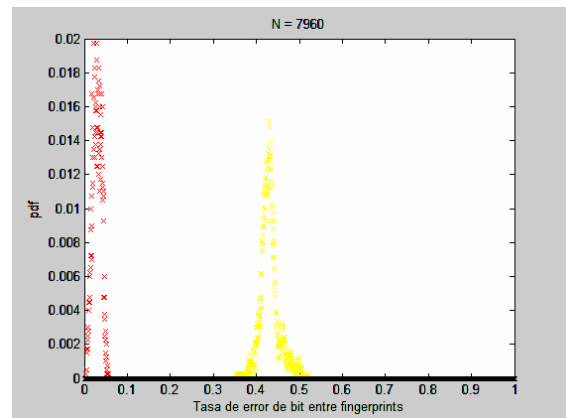
Gráfica 6. PDF Vs BER (N = 3960)



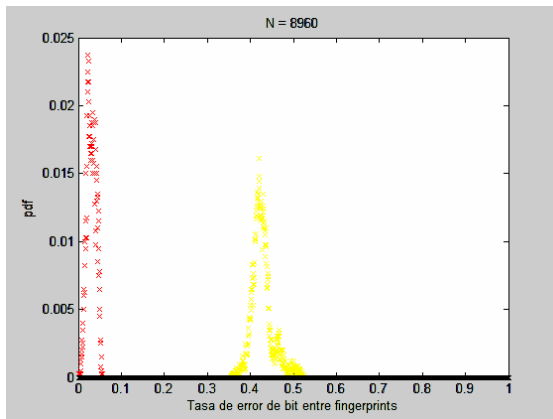
Gráfica 9. PDF Vs BER (N = 6960)



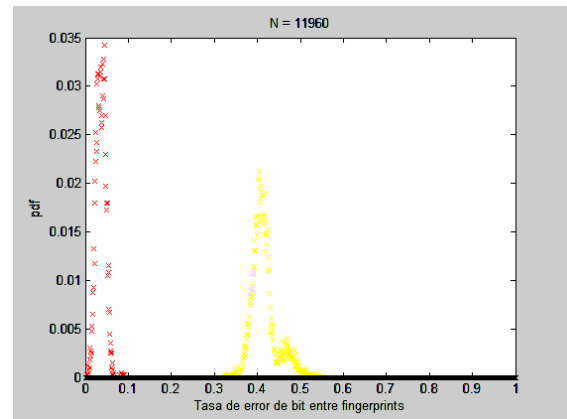
Gráfica 7. PDF Vs BER (N = 4960)



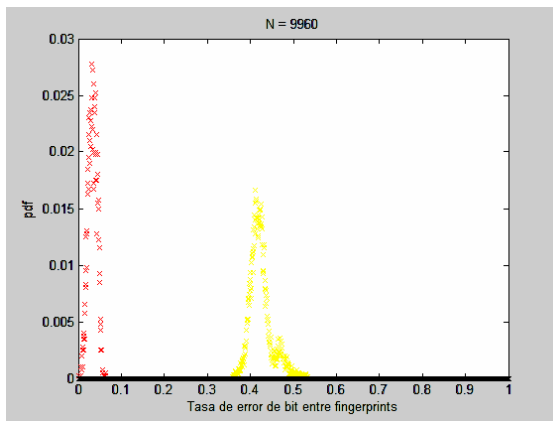
Gráfica 10. PDF Vs BER (N = 7960)



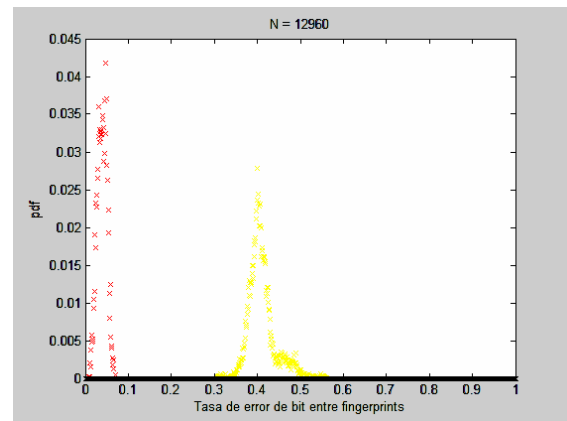
Gráfica 11. PDF Vs BER (N = 8960)



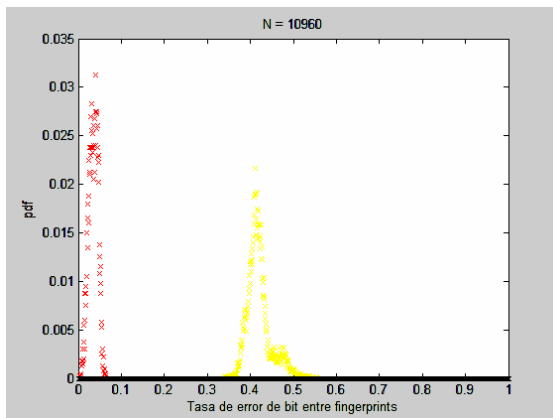
Gráfica 14. PDF Vs BER (N = 11960)



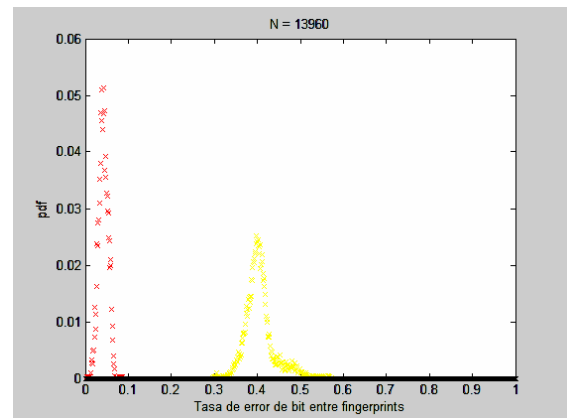
Gráfica 12. PDF Vs BER (N = 9960)



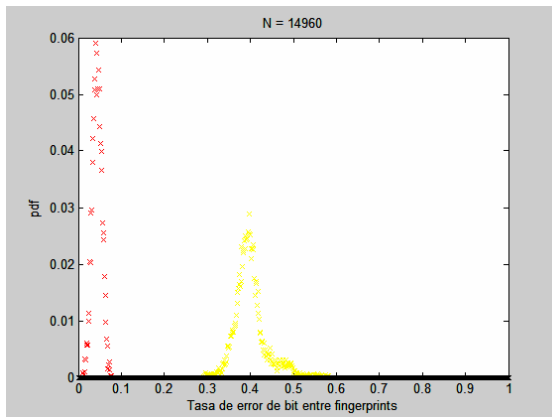
Gráfica 15. PDF Vs BER (N = 12960)



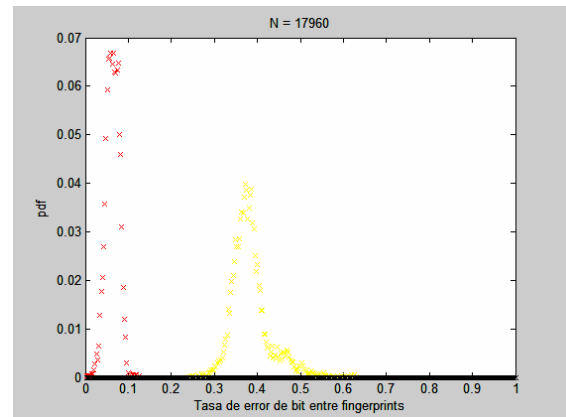
Gráfica 13. PDF Vs BER (N = 10960)



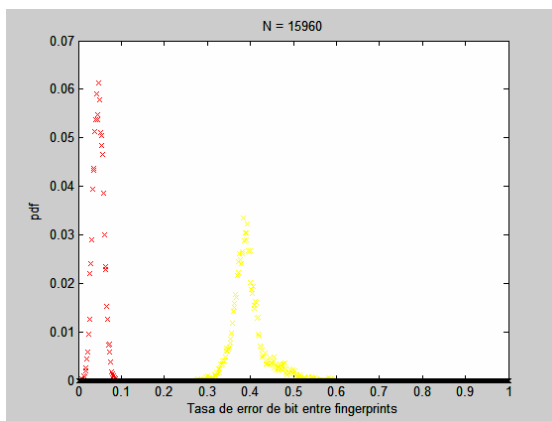
Gráfica 16. PDF Vs BER (N = 13960)



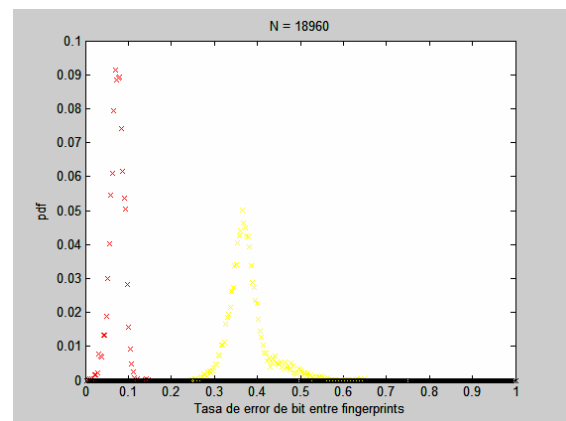
Gráfica 17. PDF Vs BER (N = 14960)



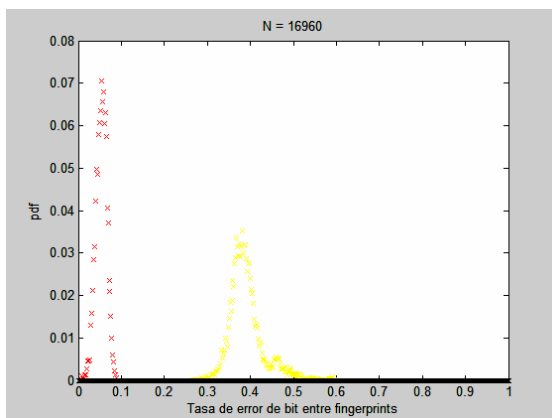
Gráfica 20. PDF Vs BER (N = 17960)



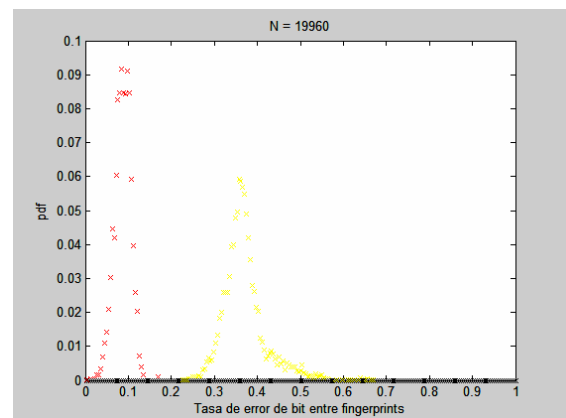
Gráfica 18. PDF Vs BER (N = 15960)



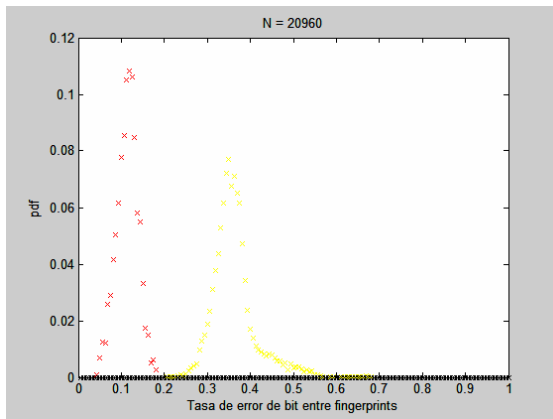
Gráfica 21. PDF Vs BER (N = 18960)



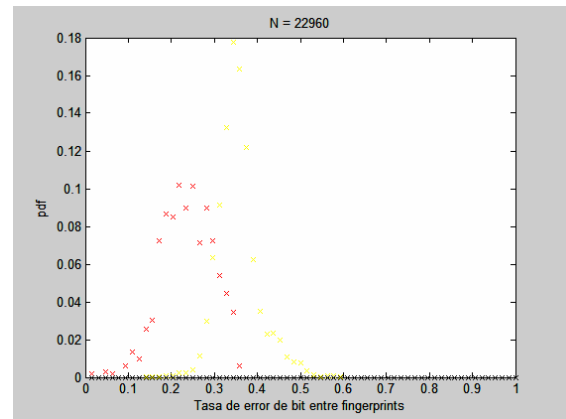
Gráfica 19. PDF Vs BER (N = 16960)



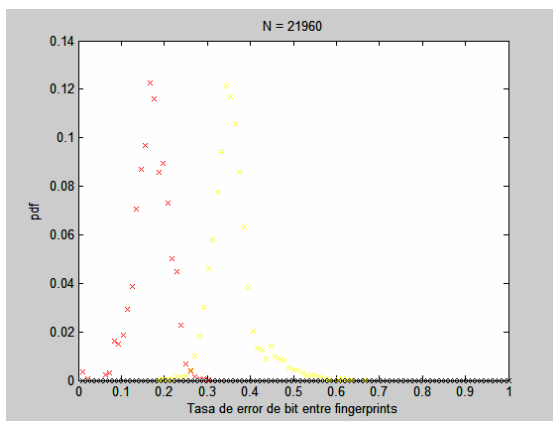
Gráfica 22. PDF Vs BER (N = 19960)



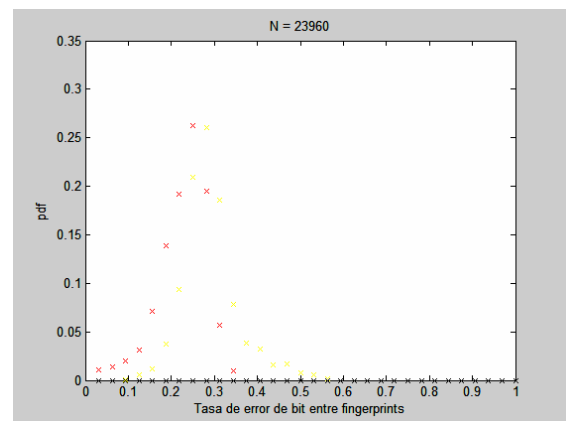
Gráfica 23. PDF Vs BER (N = 20960)



Gráfica 25. PDF Vs BER (N = 22960)



Gráfica 24. PDF Vs BER (N = 21960)



Gráfica 26. PDF Vs BER (N = 23960)

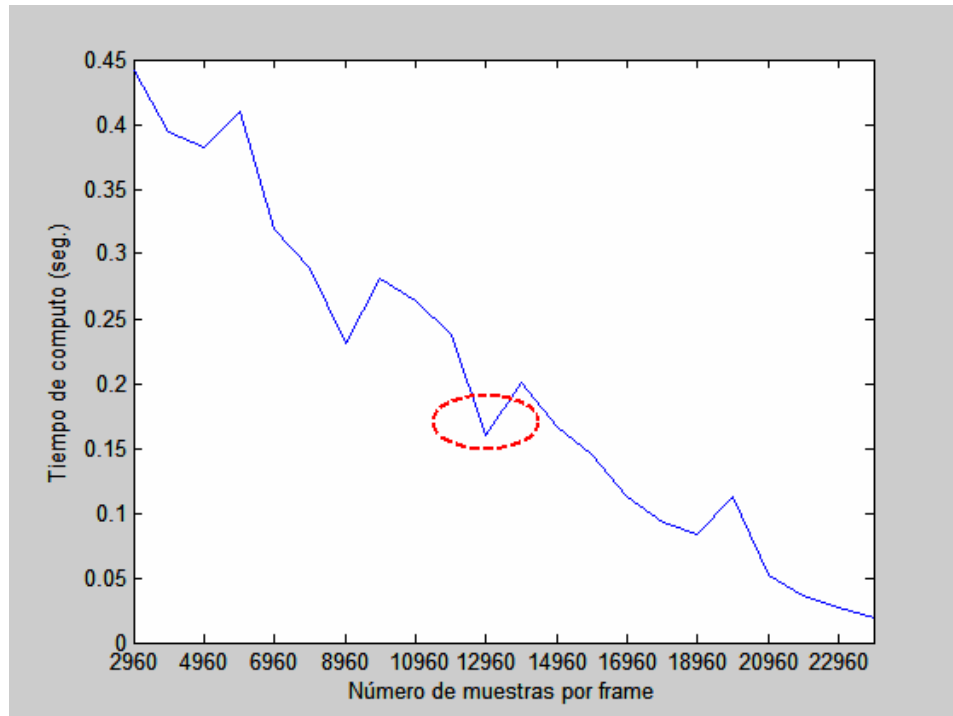
En la tabla 3 representamos la evolución de la P_f (probabilidad de falsa alarma), la P_n (probabilidad de no alarma), el tiempo de cómputo de un *fingerprint* y el número de bits por *fingerprint* en función del número de muestras por trama.

Número de muestras por trama	P_n	P_f	Tiempo de cómputo (miliseg.)	Nº bits
2960	0	3.7217e-138	441.6406	229 x 32
3960	0	1.3241e-090	394.2656	163 x 32
4960	0	3.0472e-061	382.1563	123 x 32

5960	0	4.8378e-048	409.6719	97 x 32
6960	0	9.4560e-037	319.6719	79 x 32
7960	0	4.8921e-031	289.1719	65 x 32
8960	0	1.4626e-024	231.0313	54 x 32
9960	0	3.6419e-022	280.6250	46 x 32
10960	0	5.3356e-018	263.4531	39 x 32
11960	0	1.5091e-015	237.3906	33 x 32
12960	0	1.6230e-013	159.8281	28 x 32
13960	0	1.8587e-011	200.1719	24 x 32
14960	0	9.3475e-010	167.0781	20 x 32
15960	0	7.0960e-009	145.4219	17 x 32
16960	0	7.3553e-008	112.4688	14 x 32
17960	0	2.0731e-006	92.7969	11 x 32
18960	0	1.3991e-005	83.3750	9 x 32
19960	1.2504e-012	6.2113e-005	113.1406	7 x 32
20960	7.9062e-004	1.7084e-004	51.8750	5 x 32
21960	3.9130e-001	1.7864e-004	35.8594	3 x 32
22960	8.2985e-001	1.3520e-004	26.3906	2 x 32
23960	8.0597e-001	4.5572e-002	19.5313	1 x 32

Tabla 3. Evolución de la P_f , la P_n , el tiempo de cómputo y el número de bits por *fingerprint* en función del número de muestras por trama

De los resultados obtenidos, vemos como el tiempo de cómputo de un *fingerprint* disminuye a medida que el número de muestras por trama aumenta; en contra, a medida que aumenta el número de muestras por trama, aumenta la probabilidad de no alarma y la probabilidad de falsa alarma. Vamos a tomar como solución de compromiso, un tamaño de ventana de 12960 porque mantiene una P_n igual a cero y una P_f muy baja ($P_f = 1.6230 \cdot 10^{-13}$) y, además, si observamos la gráfica 27, en $N=12960$ hay un mínimo relativo. Este mínimo relativo nos lleva a no escoger a $N=13960$ ni $N=14960$ como posibles soluciones de compromiso, pues el tiempo de cómputo es mayor que con $N=12960$ y la probabilidad de error es mayor. Podríamos pensar en escoger $N=15960$ como solución de compromiso, pero en este caso ya tenemos una P_f bastante grande ($P_f = 7.0960 \cdot 10^{-9}$).



Gráfica 27. Número de muestras por trama Vs Tiempo de cómputo de un fingerprint

Como conclusión de esta prueba, con los resultados obtenidos, tomamos como valor óptimo para el tamaño de la ventana de Hamming: **12960**, lo cual implica asumir lo siguiente:

$$P_n = 0$$

$$P_f = 1.6230 \cdot 10^{-13}$$

Tiempo de cómputo de 1 fingerprint = 0,159 segundos

Número de bits por fingerprint = $28 \times 32 = 896$ bits

□ **Característica analizada: Factor de Overlap**

El objetivo de esta prueba es analizar hasta que valor podemos aumentar el factor de *overlap*, es decir, el solapamiento entre ventanas de Hanning adyacentes, manteniendo un funcionamiento correcto del algoritmo.

El hecho de centrarnos en aumentar el factor de *overlap*, y no en disminuirlo, es porque al aumentar este parámetro, el número de *sub-fingerprints* por cada 3 segundos de anuncio detectado será menor y, por tanto, disminuirá el tiempo de computación.

Esta prueba está implementada en el código de Matlab "**prueba_Overlap.m**" (véase Anexo 2.18). Este código sigue exactamente la

misma estructura que el "prueba_N.m", salvo que ahora el parámetro variable es el factor de *overlap*.

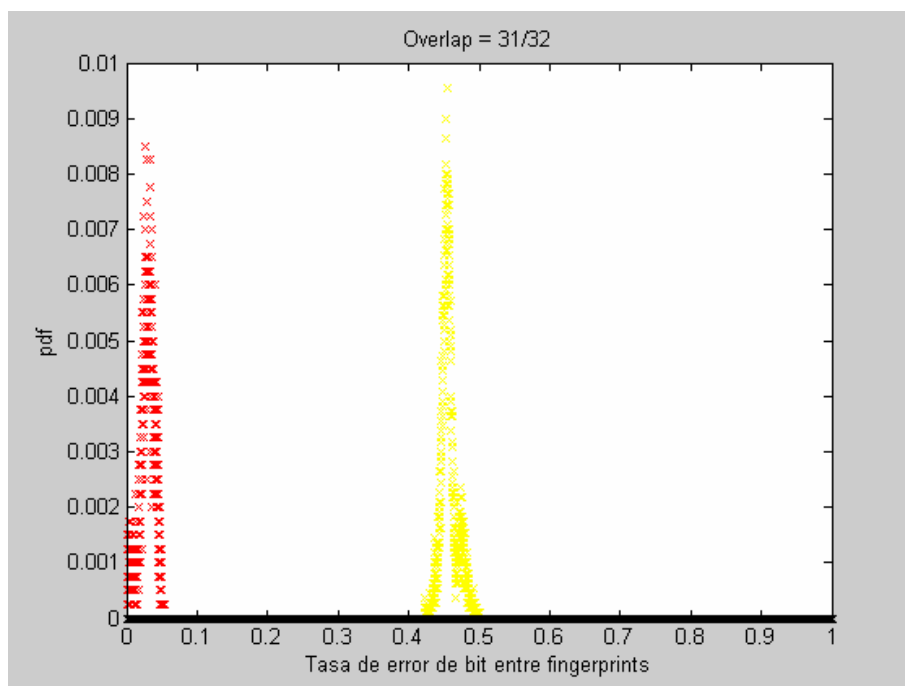
Veamos en la tabla 4 las características que se emplean en esta prueba:

<u>CARACTERÍSTICAS DE LA PRUEBA</u>	
Número de muestras por trama	2960
Factor de overlap	<i>Variable</i>
Número de bandas	32
Frecuencia inferior	300 Hz
Frecuencia superior	2000 Hz
Tiempo de detección	3 segundos

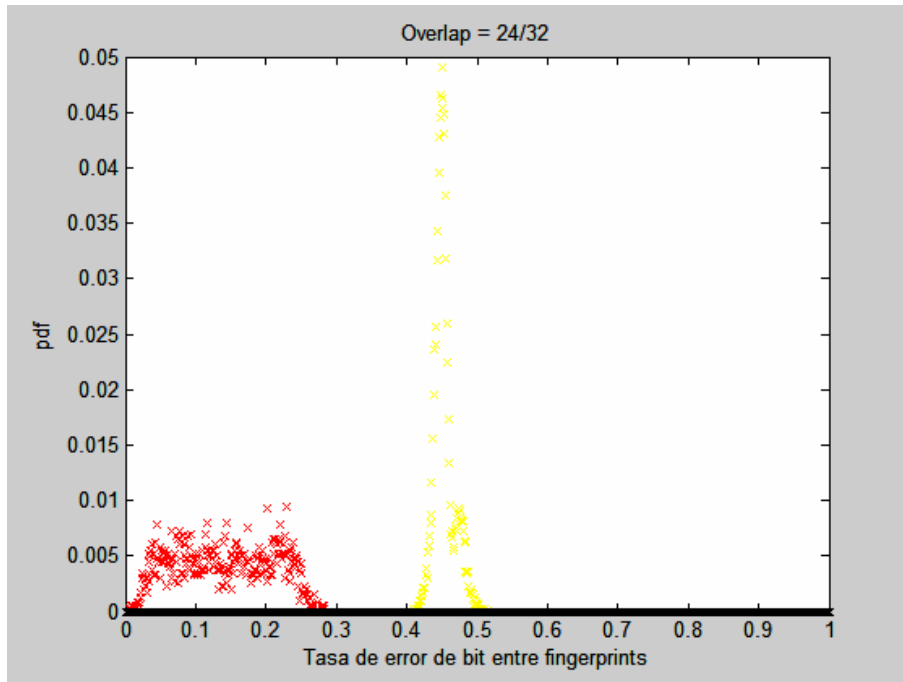
Tabla 4. Característica de la prueba "variación del *overlap*"

Veamos los resultados obtenidos más relevantes al ejecutar el código, en el cual variamos el parámetro objeto de análisis (el número factor de *overlap*) tomando los siguiente valores: 31/32, 24/32, 16/32, 12/32 y 8/32.

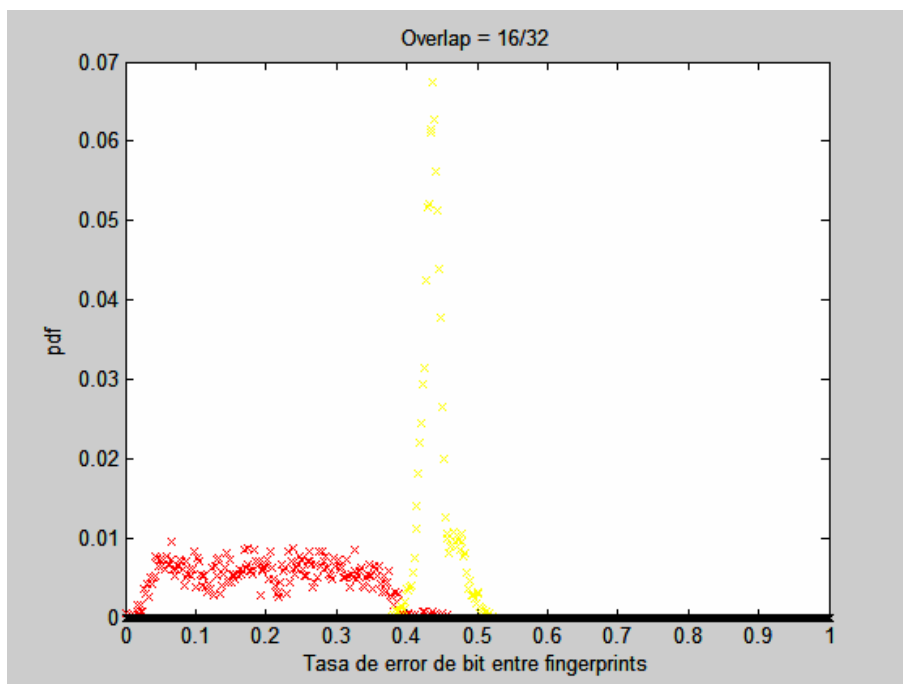
En primer lugar vamos a ver cómo evoluciona la función densidad de probabilidad experimental según el factor de *overlap*. Para ello mostramos las gráficas resultados de la prueba:



Gráfica 28. PDF Vs BER (Overlap = 31/32)



Gráfica 29. PDF Vs BER (Overlap = 24/32)



Gráfica 30. PDF Vs BER (Overlap = 16/32)

Como podemos observar, a partir de que cambiamos el factor de *overlap* a un valor diferente del de diseño, la función densidad de probabilidad relativa a la probabilidad de no alarma se ensancha muy rápidamente. Por tanto, no conseguimos ningún resultado satisfactorio al variar el solapamiento entre ventanas adyacentes porque la probabilidad de no alarma crece muy deprisa a medida que decrementamos el solapamiento.

Por lo tanto, como conclusión de esta prueba, tomamos como valor óptimo del factor de *overlap* el mismo que el de diseño, es decir, 31/32.

❑ **Característica analizada: Número de bandas**

El objetivo de esta prueba es analizar hasta que valor podemos disminuir el número de bandas en que dividimos el espectro de cada trama, manteniendo un funcionamiento correcto del algoritmo, para disminuir el número de bits por *fingerprint*. Así conseguiremos un menor tiempo de cómputo de un *fingerprint* y un tiempo de búsqueda más eficiente.

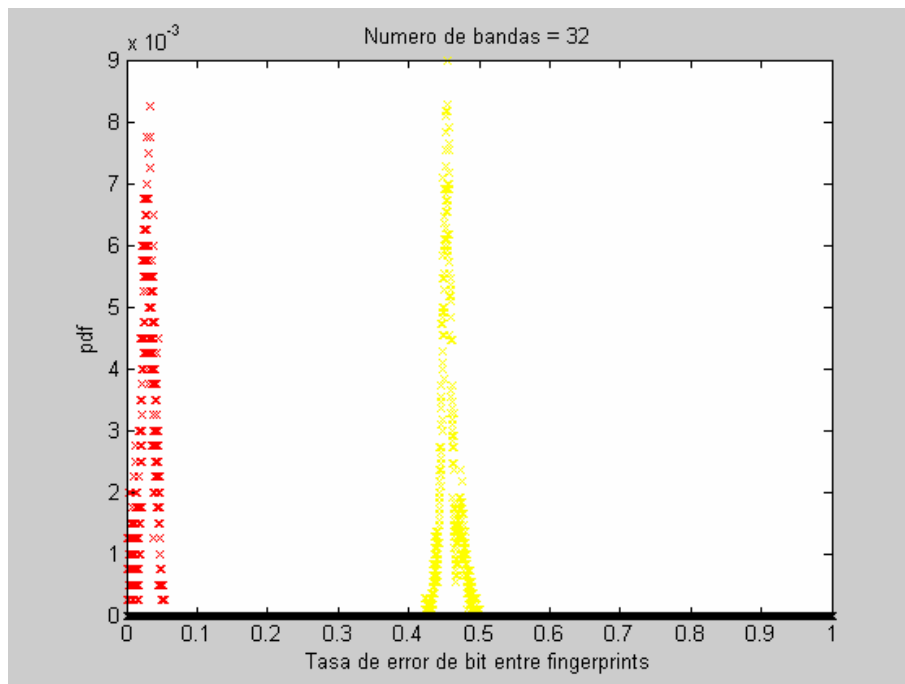
Esta prueba está implementada en el código de Matlab "**prueba_NumBand.m**" (véase Anexo 2.19). Este código sigue exactamente la misma estructura que el "**prueba_N.m**", salvo que ahora el parámetro variable es el número de bandas.

Veamos en la tabla 5 las características que se emplean en esta prueba:

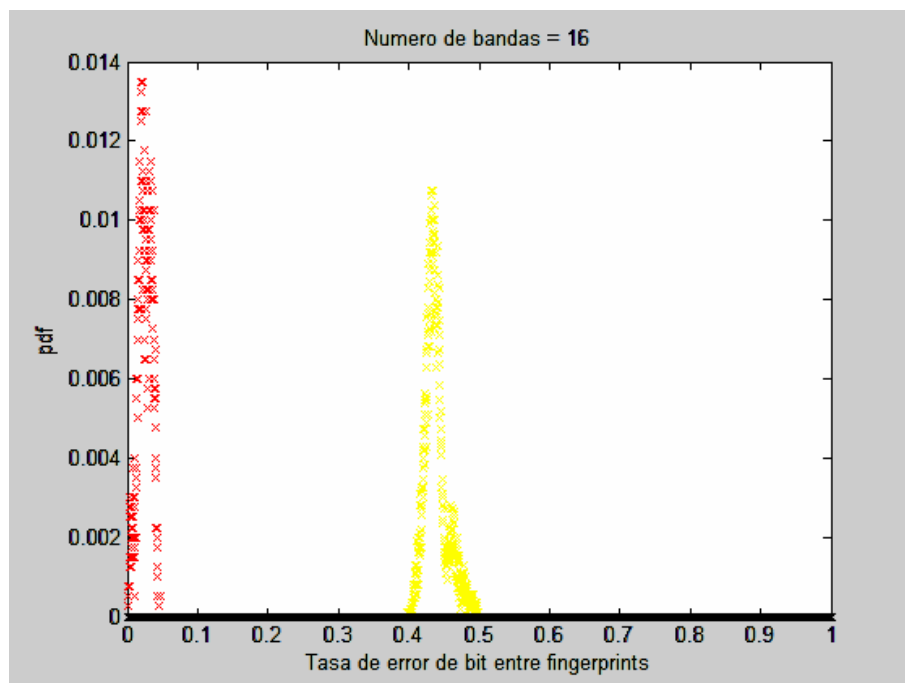
<u>CARACTERÍSTICAS DE LA PRUEBA</u>	
Número de muestras por trama	2960
Factor de overlap	31/32
Número de bandas	<i>Variable</i>
Frecuencia inferior	300 Hz
Frecuencia superior	2000 Hz
Tiempo de detección	3 segundos

Tabla 5. Característica de la prueba "variación del *número de bandas*"

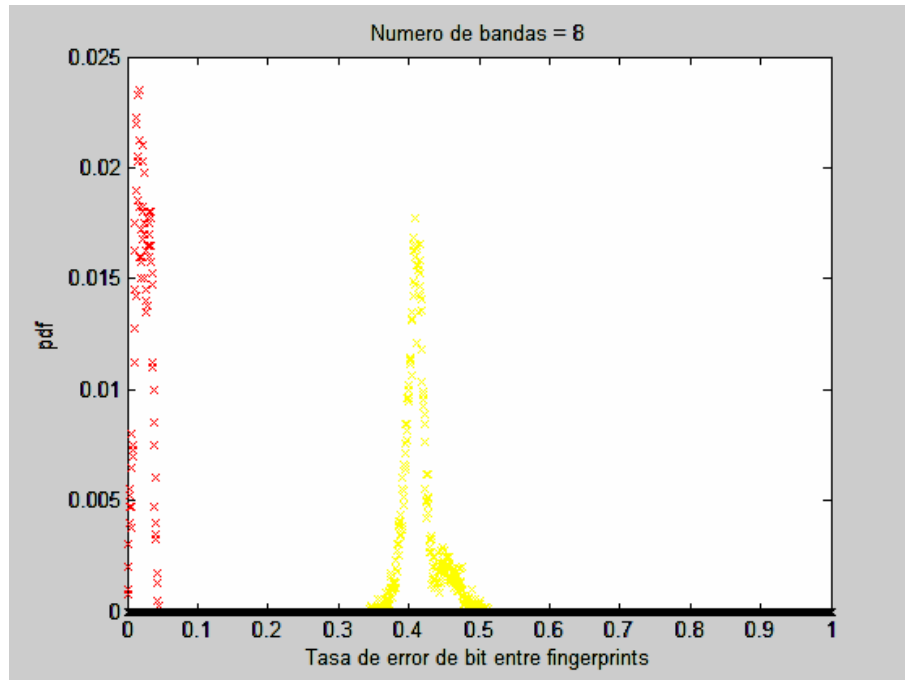
Veamos los resultados obtenidos más relevantes al ejecutar el código, en el cual variamos el parámetro objeto de análisis (el número de bandas en que dividimos el espectro de cada trama) tomando los siguiente valores: 32, 16 y 8.



Gráfica 31. PDF Vs BER (Número de bandas = 32)



Gráfica 32. PDF Vs BER (Número de bandas = 16)



Gráfica 33. PDF Vs BER (Número de bandas = 8)

En la tabla 6 representamos la evolución de la P_f (probabilidad de falsa alarma), la P_n (probabilidad de no alarma), el tiempo de cómputo de un *fingerprint* y el número de bits por *fingerprint* en función del número de bandas espectrales.

Número de bandas espectrales	P_n	P_f	Tiempo de cómputo (miliseg.)	Nº bits
32	0	3.7217e-138	443.9531	229 x 32
16	0	6.2410e-059	411.2813	229 x 16
8	0	7.7014e-026	681.4219	229 x 8

Tabla 6. Evolución de la P_f , la P_n , el tiempo de cómputo y el número de bits por *fingerprint* en función del número de bandas espectrales

Al analizar los resultados, vemos que con una división en 16 bandas espectrales conseguimos el menor tiempo de cómputo de un *fingerprint* (0,411 segundos); mientras que con 8 bandas espectrales, este tiempo aumenta casi tres décimas de segundo. Por otra parte, las probabilidades de error (P_n y P_f) se mantienen suficientemente bajas en ambos casos. Por tanto, en cuanto a tiempo de cómputo de un *fingerprint*, tomaríamos una división de 16 bandas como valor óptimo. Sin embargo, con una división en 8 bandas conseguimos reducir a la mitad el tamaño de memoria necesario para un *fingerprint*, pues

pasamos de 2 bytes por cada *sub-fingerprint* a 1 byte. Esta reducción de memoria proporciona ventajas en el algoritmo de búsqueda.

Como conclusión de esta prueba, analizando sus resultados, tomamos como valor óptimo para el número de bandas espectrales en que dividimos el espectro de cada trama: **8 bandas**; lo cual implica asumir lo siguiente:

$$P_n = 0$$

$$P_f = 7.701 \cdot 10^{-26}$$

$$\text{Tiempo de cómputo de 1 fingerprint} = 0,681 \text{ segundos}$$

$$\text{Número de bits por fingerprint} = 229 \times 8 = 1832 \text{ bits}$$

❑ **Característica analizada: Tiempo de captura**

El objetivo de esta prueba es disminuir el tiempo de captura empleado para coger un trozo de audio e intentar identificar este objeto de audio como uno de los anuncios de nuestra base de datos.

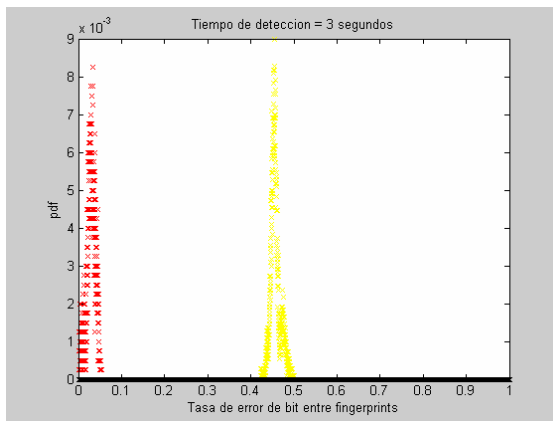
Esta prueba está implementada en el código de Matlab **prueba_TimeDetect.m** (véase Anexo 2.20). Este código sigue una estructura muy similar al código **prueba_N.m**, salvo que ahora el parámetro variable es el tiempo de detección (o de captura) de un trozo de anuncio.

Veamos en la tabla 7 las características que se emplean en esta prueba:

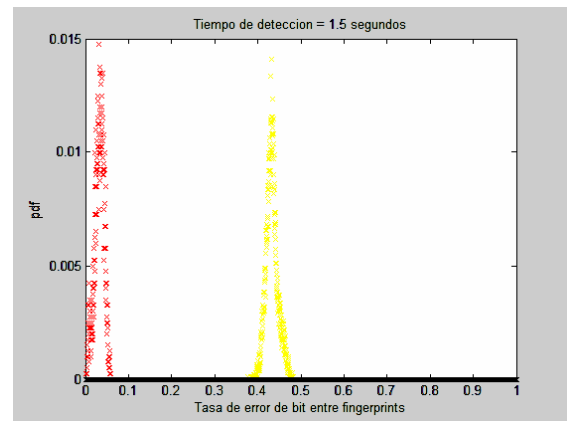
<u>CARACTERÍSTICAS DE LA PRUEBA</u>	
Número de muestras por trama	2960
Factor de overlap	31/32
Número de bandas	32
Frecuencia inferior	300 Hz
Frecuencia superior	2000 Hz
Tiempo de detección	<i>Variable</i>

Tabla 7. Característica de la prueba "variación del tiempo de detección"

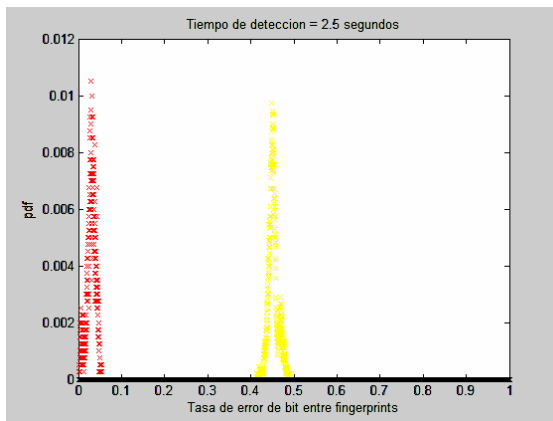
Veamos los resultados obtenidos más relevantes al ejecutar el código, en el cual variamos el parámetro objeto de análisis (el tiempo de detección de trozos de audio) tomando los siguiente valores: 3 segundos (24000 muestras), 2.5 segundos (20000 muestras), 2 segundos (16000 muestras), 1.5 segundos (12000 muestras), 1 segundo (8000 muestras) y 0.5 segundo (4000 muestras).



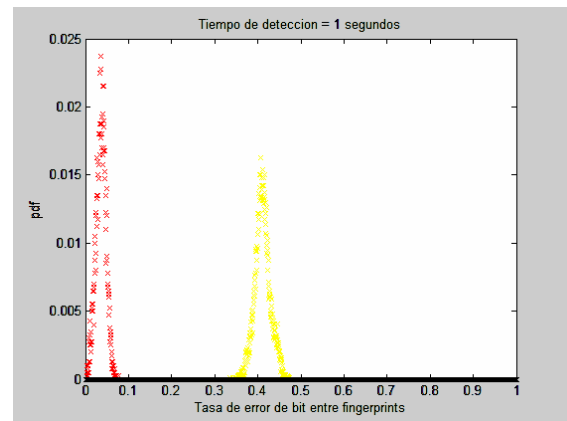
Gráfica 34. PDF Vs BER (T. Captura = 3 seg.)



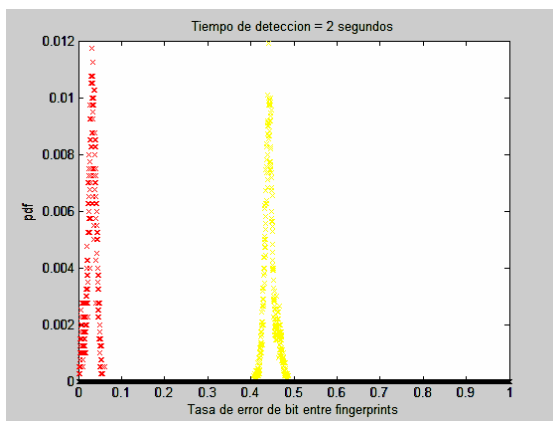
Gráfica 37. PDF Vs BER (T. Captura = 1,5 seg.)



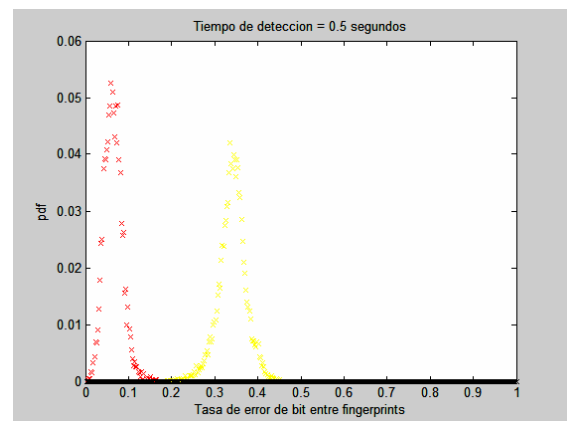
Gráfica 35. PDF Vs BER (T. Captura = 2,5 seg.)



Gráfica 38. PDF Vs BER (T. Captura = 1 seg.)



Gráfica 36. PDF Vs BER (T. Captura = 2 seg.)



Gráfica 39. PDF Vs BER (T. Captura = 0,5 seg.)

En la tabla 8 representamos la evolución de la P_f (probabilidad de falsa alarma), la P_n (probabilidad de no alarma), el tiempo de cómputo de un *fingerprint* y el número de bits por *fingerprint* en función del tiempo de detección escogido.

Tiempo de detección (segundos)	P_n	P_f	Tiempo de cómputo (miliseg.)	Nº bits
3	0	3.7217e-138	438.6875	229 x 32
2,5	0	2.4306e-129	415.5781	186 x 32
2	0	1.0473e-105	605.4844	142 x 32
1,5	0	6.8037e-075	228.6250	99 x 32
1	0	3.1933e-037	123.1563	55 x 32
0,5	1.5208e-009	1.5918e-007	24.2344	12 x 32

Tabla 8. Evolución de la P_f , la P_n , el tiempo de cómputo y el número de bits por *fingerprint* en función del número de bandas espectrales

Con los resultados obtenidos, observamos como el algoritmo sigue funcionando con unas probabilidades de error (tanto P_n como P_f) muy bajas hasta que disminuimos a medio segundo el tiempo de captura. Por su parte, el tiempo de cómputo de un *fingerprint* también disminuye proporcionalmente con la disminución del tiempo de captura.

Por lo tanto, como conclusión de esta prueba, con los resultados obtenidos de ella, tomamos como valor óptimo para el tamaño tiempo de captura de la señal de audio de entrada a nuestro sistema: **1 segundo**; lo cual implica asumir lo siguiente:

$$P_n = 0$$

$$P_f = 3.1933 \cdot 10^{-37}$$

$$\text{Tiempo de cómputo de 1 fingerprint} = 0,123 \text{ segundos}$$

$$\text{Número de bits por fingerprint} = 55 \times 32 = 1760 \text{ bits}$$

4.3. PRUEBAS CON VARIAS CARACTERÍSTICA

En las tres pruebas que se describen en este apartado se parte de los siguientes valores óptimos obtenidos para cada uno de los parámetros a variar (véase apartado 4.2):

- Número de muestras por trama: 12960 muestras
- Factor de overlap: 31/32
- Número de bandas espectrales: 8
- Tiempo de captura: 1 segundo

En principio podríamos pensar en que la solución óptima para nuestro sistema sería coger los valores óptimos obtenidos para cada una de las características que influyen en el algoritmo de extracción; sin embargo, esto no es posible porque, a parte de las altas probabilidades de error que tendríamos, estos valores no son compatibles entre sí. Esta incompatibilidad se debe a que el tamaño óptimo de la ventana de Hanning ha salido 12960 muestras; mientras que el tiempo de captura óptimo ha salido 1 segundo, que son 8000 muestras. Como es obvio, no podemos tener un tamaño de ventana superior al número de muestras correspondientes a una captura y, por lo tanto, no podemos tomar estos valores para nuestra optimización del algoritmo de extracción.

Por consiguiente, el objetivo de estas prueba es, a partir de estos valores óptimos, centrarse en una de las características tomando su valor óptimo y variar los valores del resto de características desde sus óptimos hacia valores menos óptimos que permitan tener un algoritmo de extracción de *fingerprints* que funcione correctamente.

Aunque hayan sido 4 las características sobre las que hemos hecho las pruebas en el apartado anterior, dado que para una de ellas (el factor de *overlap*) no hay otra posibilidad que la de su valor de origen, en las pruebas siguientes sólo nos centramos en las otras tres características. Así pues, el valor tomado para el factor de *overlap* permanece invariante en 31/32 para las siguientes pruebas.

Las tres pruebas que se exponen a continuación siguen una estructura muy similar a las anteriores (véase figura 7), salvo que, ahora en vez de modificar sólo una característica, se modifican dos por cada prueba. Esto, en el código de Matlab, lo único que implica es hacer un bucle "for" dentro de otro bucle "for"; de tal forma que cada bucle va modificando cada una de las dos características cuyos valores se quieren analizar.

Veamos las tres pruebas realizadas y sus correspondientes resultados.

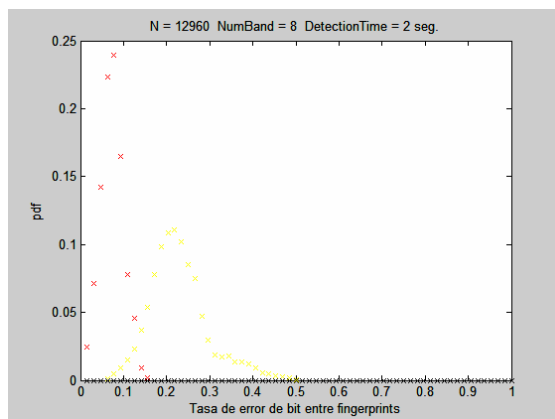
❑ Característica óptima: Número de muestras por trama (N)

Esta prueba está implementada en el código de Matlab **"prueba_global_N.m"** (véase Anexo 2.21). En ella permanece invariante el parámetro "número de muestras por trama", que se fija en 12960 muestras, y se varían los parámetros "número de bandas espectrales para la división de una trama" y "tiempo de captura" tomando los siguientes valores:

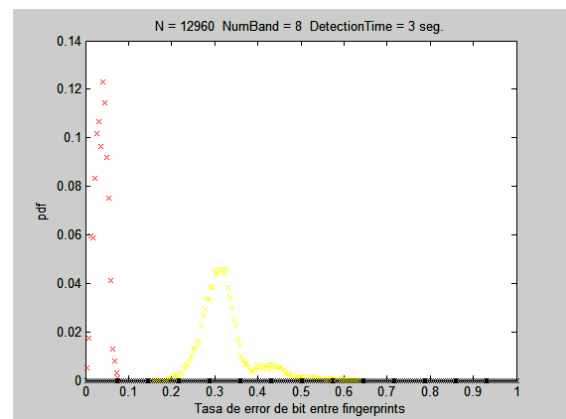
- para el número de bandas espectrales: 8, 16 y 32
- para el tiempo de captura: 2 segundos, 2.5 segundos y 3 segundos

Como ya comentamos anteriormente, tomando el valor óptimo del número de muestras por trama, no podemos tomar el valor óptimo correspondiente al tiempo de captura (1 segundo), pues 1 segundo equivale a 8000 muestras, valor más pequeño que el tamaño de enventanado.

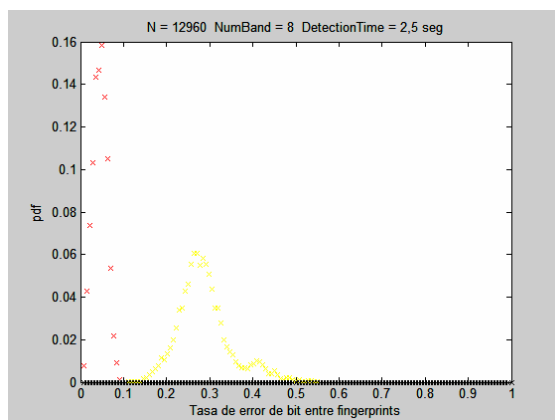
Veamos los resultados obtenidos al ejecutar esta prueba. En primer lugar se muestran las funciones densidad de probabilidad obtenidas:



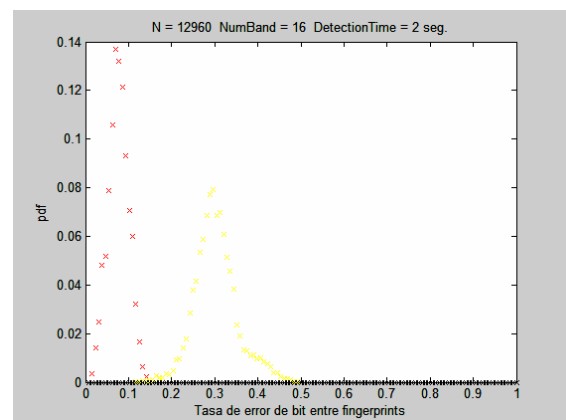
Gráfica 40. PDF Vs BER (N = 12960 N° Bandas = 8 T. Captura = 2 seg.)



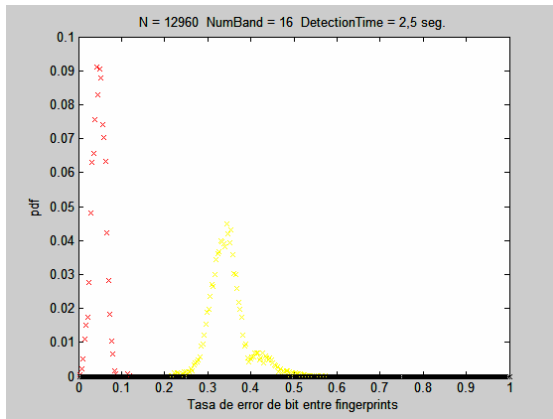
Gráfica 42. PDF Vs BER (N = 12960 N° Bandas = 8 T. Captura = 3 seg.)



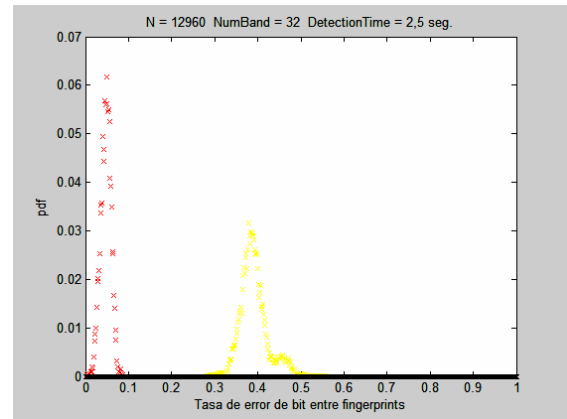
Gráfica 41. PDF Vs BER (N = 12960 N° Bandas = 8 T. Captura = 2,5 seg.)



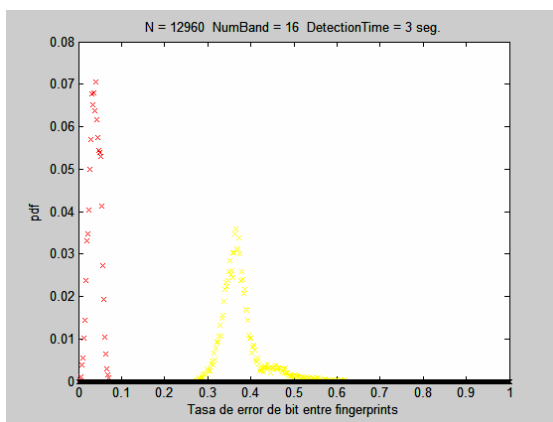
Gráfica 43. PDF Vs BER (N = 12960 N° Bandas = 16 T. Captura = 2 seg.)



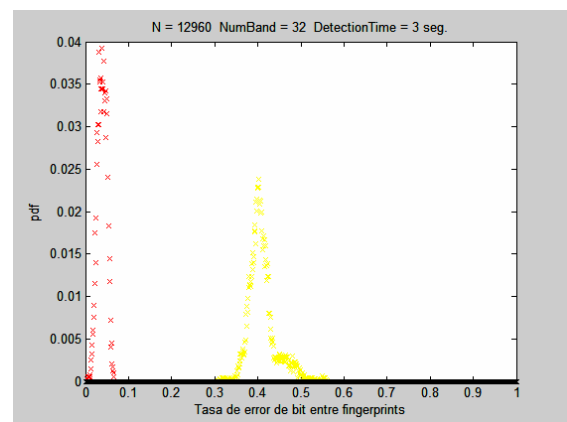
Gráfica 44. PDF Vs BER (N = 12960 N° Bandas = 16 T. Captura = 2,5 seg.)



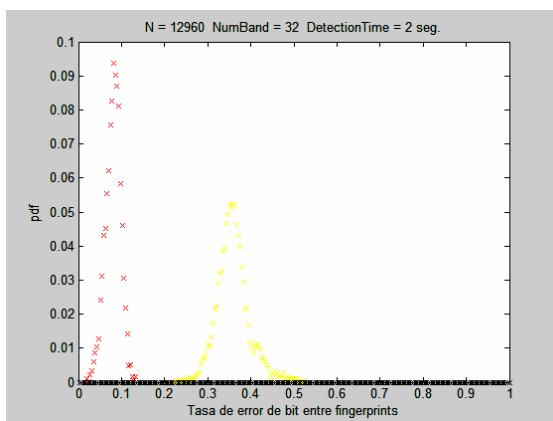
Gráfica 47. PDF Vs BER (N = 12960 N° Bandas = 32 T. Captura = 2,5 seg.)



Gráfica 45. PDF Vs BER (N = 12960 N° Bandas = 16 T. Captura = 3 seg.)



Gráfica 48. PDF Vs BER (N = 12960 N° Bandas = 32 T. Captura = 3 seg.)



Gráfica 46. PDF Vs BER (N = 12960 N° Bandas = 32 T. Captura = 2 seg.)

A continuación, en la tabla 9, se muestra un cuadro resumen con las probabilidades de falsa alarma y de no alarma, el tiempo de cómputo de un *fingerprint* y el número de bits por *fingerprint* en función de los valores escogidos para el número de bandas espectrales y el tiempo de captura.

N = 12960		Factor de Overlap = 31/32	
Nº Bandas T. Captura	8	16	32
2 segundos	P _f = 2.3906e-001 P _n = 4.6491e-004 T. Cómputo = 0.0268 Nº bits = 8 x 8	P _f = 7.2717e-003 P _n = 2.6744e-007 T. Cómputo = 0.0430 Nº bits = 8 x 16	P _f = 1.7733e-006 P _n = 0 T. Cómputo = 0.0464 Nº bits = 8 x 32
2,5 segundos	P _f = 4.1701e-002 P _n = 0 T. Cómputo = 0.0831 Nº bits = 18 x 8	P _f = 7.4569e-005 P _n = 0 T. Cómputo = 0.0911 Nº bits = 18 x 16	P _f = 4.1304e-011 P _n = 0 T. Cómputo = 0.1026 Nº bits = 18 x 32
3 segundos	P _f = 9.2181e-003 P _n = 0 T. Cómputo = 0.1351 Nº bits = 28 x 8	P _f = 1.0161e-005 P _n = 0 T. Cómputo = 0.1487 Nº bits = 28 x 16	P _f = 2.1838e-013 P _n = 0 T. Cómputo = 0.1566 Nº bits = 28 x 32

Tabla 9. Evolución de la P_f , la P_n , el tiempo de cómputo y el número de bits por *fingerprint* en función del número de bandas espectrales y el tiempo de captura

De la tabla 9, fijándonos en la dependencia de las probabilidades P_n y P_f con los parámetros, deducimos que sólo dos posibilidades de las 9 que se plantean son viables para un funcionamiento correcto del sistema: o bien tomamos como solución 12960 muestras de tamaño de ventana de Hanning, 32 bandas espectrales y 2,5 segundos de tiempo de captura; o bien tomamos como solución 12960 muestras de tamaño de ventana de Hanning, 32 bandas espectrales y 3 segundos de tiempo de captura. De estas dos posibles soluciones, nos quedamos con la segunda por tener una probabilidad de falsa alarma menor.

Por lo tanto, como conclusión de esta prueba, aceptamos como valores que optimizan el algoritmo de extracción de *fingerprints* los siguientes:

- Número de muestras por trama: **12960 muestras**
- Factor de overlap: **31/32**
- Número de bandas espectrales: **32 bandas**
- Frecuencia inferior para la división en bandas: **300 Hz**
- Frecuencia superior para la división en bandas: **2000 Hz**
- Tiempo de captura de la señal de entrada: **3 segundos**

lo cual implica asumir lo siguiente:

$$P_n = 0$$

$$P_f = 2.1838 \cdot 10^{-13}$$

$$\text{Tiempo de cómputo de 1 fingerprint} = 0,1566 \text{ segundos}$$

$$\text{Número de bits por fingerprint} = 28 \times 32 = 896 \text{ bits}$$

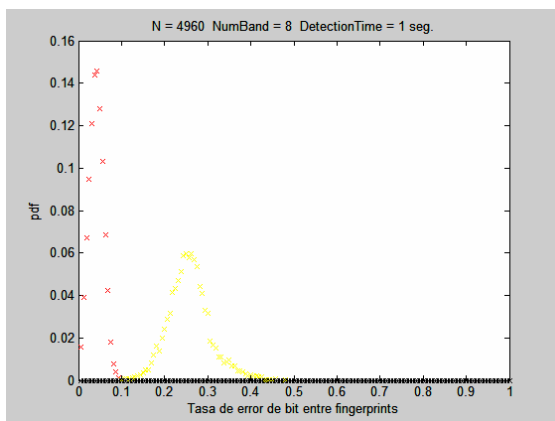
❑ Característica óptima: Número de bandas espectrales

Esta prueba está implementada en el código de Matlab "**prueba_global_NumBand.m**" (véase Anexo 2.22). En este caso permanece invariante el parámetro "número de bandas en el que se divide el espectro de cada trama", que se fija en 8 bandas, y se varían los parámetros "número de muestras por trama" y "tiempo de captura" tomando los siguientes valores:

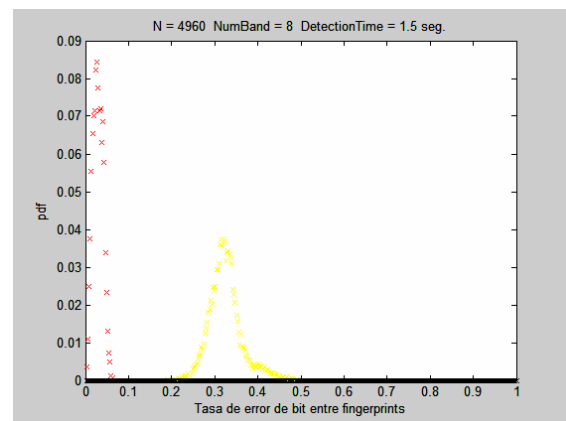
- para el número de muestras por trama: desde 11960 a 2960 muestras en intervalos de 1000
- para el tiempo de captura: 1, 1.5, 2, 2.5 y 3 segundos

Dado la incompatibilidad entre algunos valores, se añade en el código una sentencia del tipo "if" de tal forma que, para los casos en que el tamaño de la ventana de Hanning (N) sea superior al número de muestras correspondientes al tiempo de captura considerado ($time_detect$), se salta la prueba para esa combinación de valores. Es el caso, por ejemplo de tener $N=11960$ y $time_detect=8000$ (equivalente a 1 segundo de tiempo de captura).

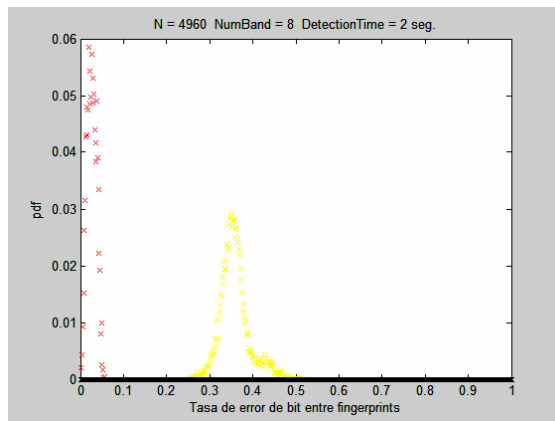
Veamos los resultados obtenidos al ejecutar esta prueba. En primer lugar se muestran en las sucesivas gráficas las funciones densidad de probabilidad para los casos más próximos a la solución óptima. En segundo lugar, en la tabla 10, se resumen todos los resultados obtenidos de la prueba, mostrándose la probabilidad de falsa alarma (P_f), la probabilidad de no alarma (P_n), el tiempo de computo de un *fingerprint* (T. C.) y el número de bits resultantes por cada *fingerprint* para cada posible combinación de los valores tomados por los parámetros.



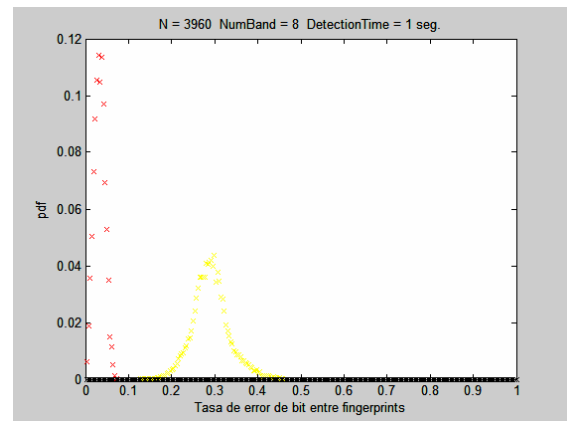
Gráfica 49. PDF Vs BER (N = 4960 N° Bandas = 8 T. Captura = 1 seg.)



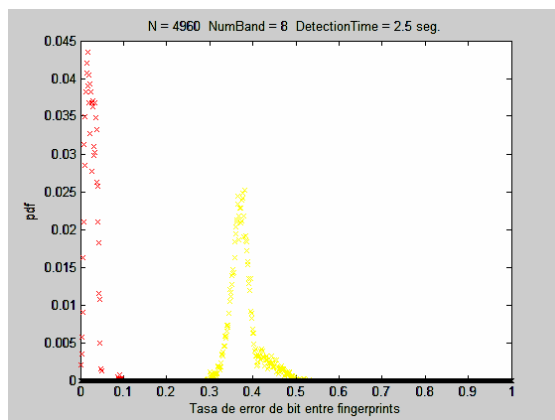
Gráfica 50. PDF Vs BER (N = 4960 N° Bandas = 8 T. Captura = 1,5 seg.)



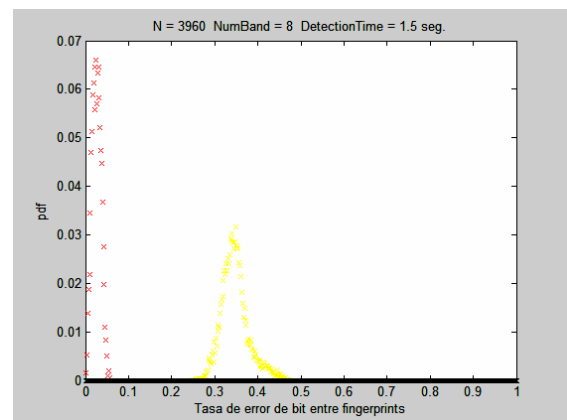
Gráfica 51. PDF Vs BER (N = 4960 N° Bandas = 8 T. Captura = 2 seg.)



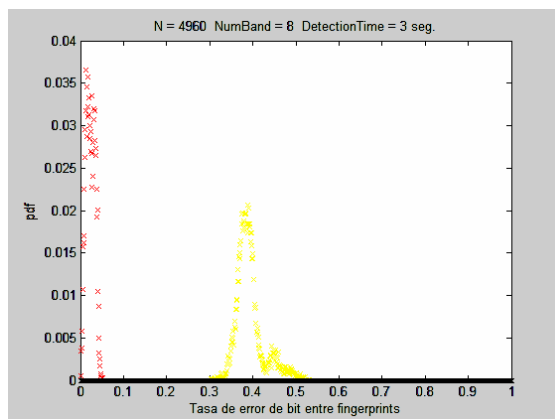
Gráfica 54. PDF Vs BER (N = 3960 N° Bandas = 8 T. Captura = 1 seg.)



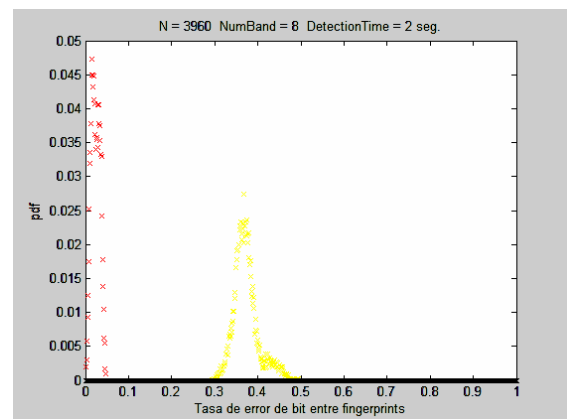
Gráfica 52. PDF Vs BER (N = 4960 N° Bandas = 8 T. Captura = 2,5 seg.)



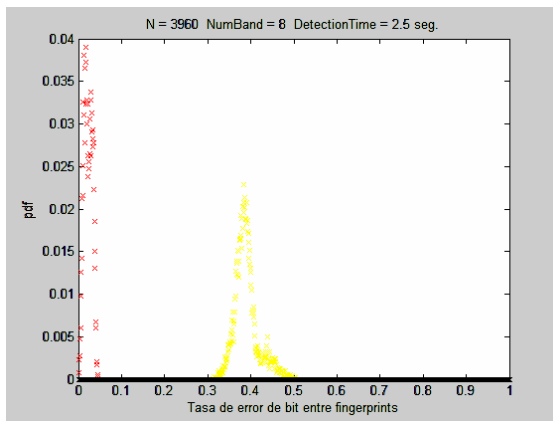
Gráfica 55. PDF Vs BER (N = 3960 N° Bandas = 8 T. Captura = 1,5 seg.)



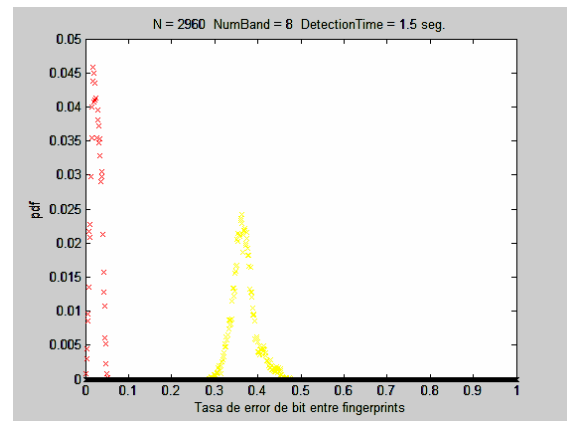
Gráfica 53. PDF Vs BER (N = 4960 N° Bandas = 8 T. Captura = 3 seg.)



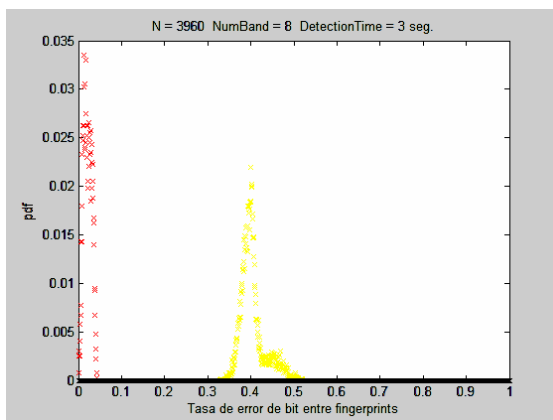
Gráfica 56. PDF Vs BER (N = 3960 N° Bandas = 8 T. Captura = 2 seg.)



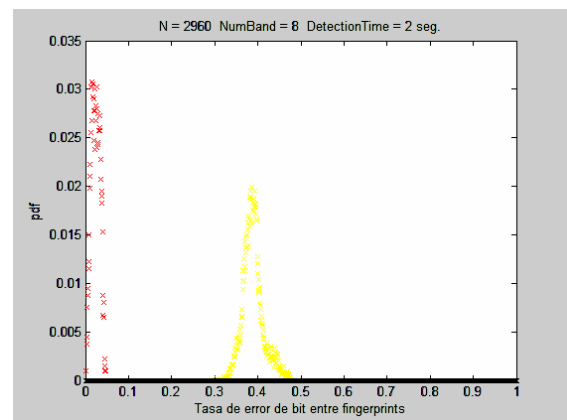
Gráfica 57. PDF Vs BER (N = 3960 N° Bandas = 8 T. Captura = 2,5 seg.)



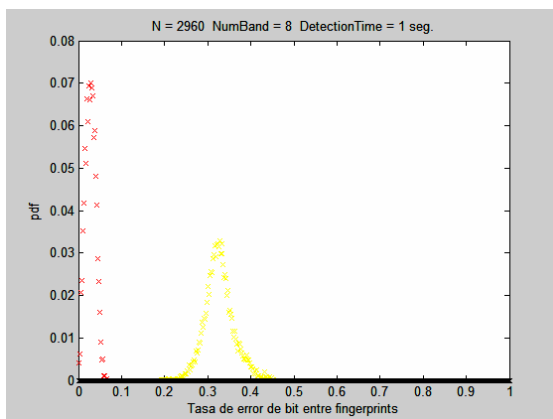
Gráfica 60. PDF Vs BER (N = 2960 N° Bandas = 8 T. Captura = 1,5 seg.)



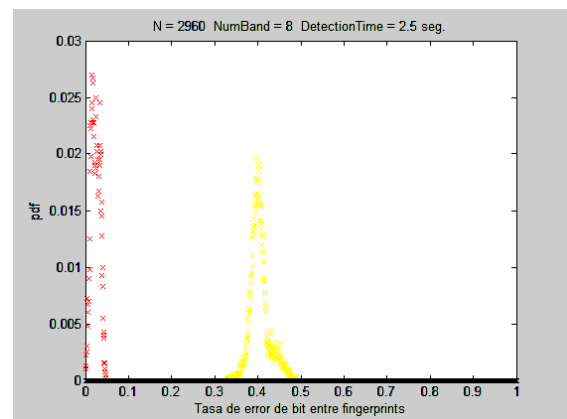
Gráfica 58. PDF Vs BER (N = 3960 N° Bandas = 8 T. Captura = 3 seg.)



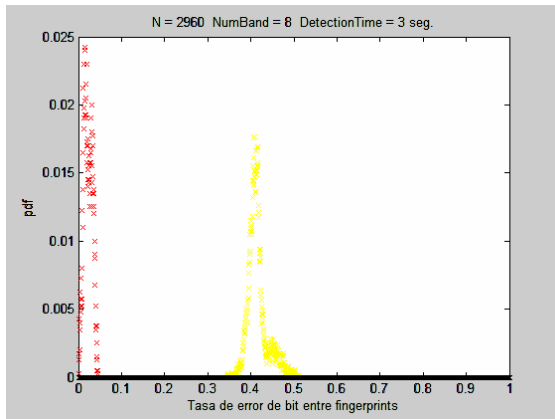
Gráfica 61. PDF Vs BER (N = 2960 N° Bandas = 8 T. Captura = 2 seg.)



Gráfica 59. PDF Vs BER (N = 2960 N° Bandas = 8 T. Captura = 1 seg.)



Gráfica 62. PDF Vs BER (N = 2960 N° Bandas = 8 T. Captura = 2,5 seg.)



Gráfica 63. PDF Vs BER (N = 2960 N° Bandas = 8 T. Captura = 3 seg.)

Número de bandas espectrales = 8		Factor de Overlap = 31/32			
T. Captura N	1 seg.	1,5 seg.	2 seg.	2,5 seg.	3 seg.
11960	---	$P_f = 8.7900e-001$ $P_n = 3.2936e-002$ T. C. = 0.0080 Nº bits = 1 x 8	$P_f = 1.6137e-001$ $P_n = 7.2846e-009$ T. C. = 0.0673 Nº bits = 11 x 8	$P_f = 2.2473e-002$ $P_n = 0$ T. C. = 0.1397 Nº bits = 22 x 8	$P_f = 4.7727e-003$ $P_n = 0$ T. C. = 0.2028 Nº bits = 33 x 8
10960	---	$P_f = 4.1963e-001$ $P_n = 1.1151e-001$ T. C. = 0.0254 Nº bits = 4 x 8	$P_f = 8.9639e-002$ $P_n = 6.4393e-015$ T. C. = 0.0885 Nº bits = 15 x 8	$P_f = 1.1777e-002$ $P_n = 0$ T. C. = 0.1567 Nº bits = 27 x 8	$P_f = 2.0604e-003$ $P_n = 0$ T. C. = 0.2274 Nº bits = 39 x 8
9960	---	$P_f = 3.4966e-001$ $P_n = 4.3603e-003$ T. C. = 0.0388 Nº bits = 7 x 8	$P_f = 3.6722e-002$ $P_n = 0$ T. C. = 0.1089 Nº bits = 20 x 8	$P_f = 3.7548e-003$ $P_n = 0$ T. C. = 0.1794 Nº bits = 33 x 8	$P_f = 4.7026e-004$ $P_n = 0$ T. C. = 0.2506 Nº bits = 46 x 8
8960	---	$P_f = 1.9081e-001$ $P_n = 1.6332e-006$ T. C. = 0.0387 Nº bits = 11 x 8	$P_f = 1.1583e-002$ $P_n = 0$ T. C. = 0.087 Nº bits = 26 x 8	$P_f = 8.7313e-004$ $P_n = 0$ T. C. = 0.1327 Nº bits = 40 x 8	$P_f = 1.1963e-004$ $P_n = 0$ T. C. = 0.1818 Nº bits = 54 x 8
7960	$P_f = 9.6414e-001$ $P_n = 1.2730e-002$ T. C. = 0.0052 Nº bits = 1 x 8	$P_f = 7.0661e-002$ $P_n = 1.0214e-014$ T. C. = 0.0625 Nº bits = 17 x 8	$P_f = 3.4128e-003$ $P_n = 0$ T. C. = 0.1188 Nº bits = 33 x 8	$P_f = 1.6160e-004$ $P_n = 0$ T. C. = 0.1758 Nº bits = 49 x 8	$P_f = 2.3829e-005$ $P_n = 0$ T. C. = 0.2389 Nº bits = 65 x 8
6960	$P_f = 5.1033e-001$ $P_n = 5.6705e-002$ T. C. = 0.0200 Nº bits = 5 x 8	$P_f = 2.2117e-002$ $P_n = 0$ T. C. = 0.0750 Nº bits = 24 x 8	$P_f = 6.6052e-004$ $P_n = 0$ T. C. = 0.1295 Nº bits = 42 x 8	$P_f = 1.2110e-005$ $P_n = 0$ T. C. = 0.2282 Nº bits = 61 x 8	$P_f = 1.9406e-006$ $P_n = 0$ T. C. = 0.6127 Nº bits = 79 x 8
5960	$P_f = 2.7415e-001$ $P_n = 6.4584e-006$ T. C. = 0.0943 Nº bits = 11 x 8	$P_f = 3.2766e-003$ $P_n = 0$ T. C. = 0.2776 Nº bits = 33 x 8	$P_f = 3.3779e-005$ $P_n = 0$ T. C. = 0.4534 Nº bits = 54 x 8	$P_f = 3.1090e-007$ $P_n = 0$ T. C. = 0.6317 Nº bits = 76 x 8	$P_f = 2.5691e-008$ $P_n = 0$ T. C. = 0.8064 Nº bits = 97 x 8

4960	$P_f = 5.8186e-002$ $P_n = 9.8255e-014$ T. C. = 0.1118 Nº bits = 20 x 8	$P_f = 8.6046e-005$ $P_n = 0$ T. C. = 0.2538 Nº bits = 46 x 8	$P_f = 5.4159e-008$ $P_n = 0$ T. C. = 0.3956 Nº bits = 72 x 8	$P_f = 2.6936e-010$ $P_n = 0$ T. C. = 0.5389 Nº bits = 98 x 8	$P_f = 2.1272e-011$ $P_n = 0$ T. C. = 0.6806 Nº bits = 123 x 8
3960	$P_f = 4.4306e-003$ $P_n = 0$ T. C. = 0.1370 Nº bits = 33 x 8	$P_f = 9.8939e-008$ $P_n = 0$ T. C. = 0.2692 Nº bits = 66 x 8	$P_f = 1.0911e-011$ $P_n = 0$ T. C. = 0.3969 Nº bits = 98 x 8	$P_f = 2.7393e-014$ $P_n = 0$ T. C. = 0.2285 Nº bits = 131 x 8	$P_f = 4.5286e-016$ $P_n = 0$ T. C. = 0.2713 Nº bits = 163 x 8
2960	$P_f = 6.9891e-006$ $P_n = 0$ T. C. = 0.0801 Nº bits = 55 x 8	$P_f = 2.4342e-012$ $P_n = 0$ T. C. = 0.1429 Nº bits = 99 x 8	$P_f = 2.3712e-018$ $P_n = 0$ T. C. = 0.2066 Nº bits = 142 x 8	$P_f = 1.5892e-024$ $P_n = 0$ T. C. = 0.2713 Nº bits = 186 x 8	$P_f = 5.6919e-026$ $P_n = 0$ T. C. = 0.3307 Nº bits = 229 x 8

Tabla 10. Evolución de la P_f , la P_n , el tiempo de cómputo y el número de bits por *fingerprint* en función del número de muestras por trama y el tiempo de captura

Observando la evolución del tiempo de cómputo y el número de bits por *fingerprint* y teniendo en cuenta que consideramos una probabilidad de error baja cuando es superior o del orden de 10^{-12} , tomamos como solución óptimo la celda sombreada en amarillo

Por lo tanto, como conclusión de esta prueba, aceptamos como valores que optimizan el algoritmo de extracción de *fingerprints* los siguientes:

- Número de muestras por trama: **2960 muestras**
- Factor de overlap: **31/32**
- Número de bandas espectrales: **8 bandas**
- Frecuencia inferior para la división en bandas: **300 Hz**
- Frecuencia superior para la división en bandas: **2000 Hz**
- Tiempo de captura de la señal de entrada: **1,5 segundos**

lo cual implica asumir lo siguiente:

$$P_n = 0$$

$$P_f = 2.4342 \cdot 10^{-12}$$

$$\text{Tiempo de cómputo de 1 fingerprint} = 0,1429 \text{ segundos}$$

$$\text{Número de bits por fingerprint} = 99 \times 8 = 792 \text{ bits}$$

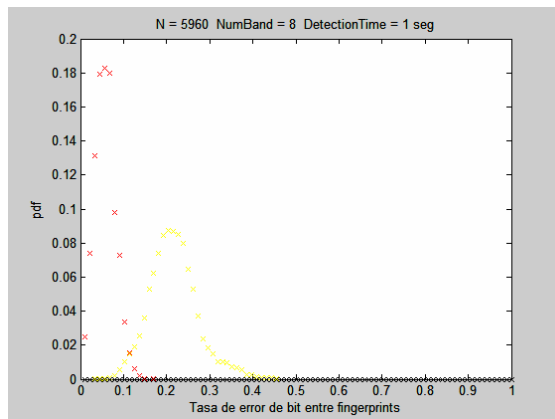
❑ Característica óptima: Tiempo de captura

Esta prueba está implementada en el código de Matlab "**prueba_global_TimeDetect.m**" (véase Anexo 2.23). En este caso permanece invariante el parámetro "tiempo de captura", que se fija en 1 segundo, y se varían los parámetros "número de muestras por trama" y "número de bandas espectrales" tomando los siguientes valores:

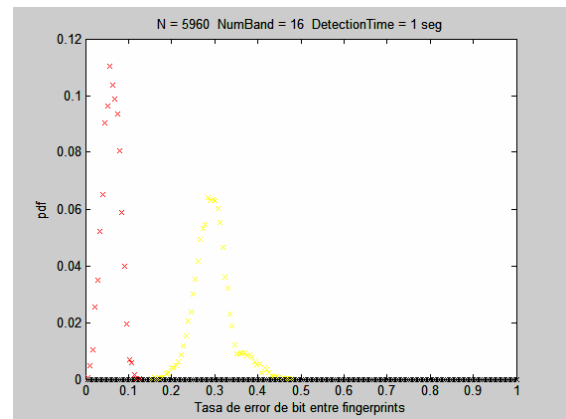
- para el número de muestras por trama: desde 7960 a 2960 muestras en intervalos de 1000
- para el número de bandas espectrales: 8, 16 y 32 bandas

En esta prueba el valor más alto posible para el tamaño de la ventana de Hanning es 7960, pues es el mayor de los valores que es inferior a 8000, que es el número de muestras correspondientes a 1 segundo de captura de la señal de audio de entrada a nuestro sistema.

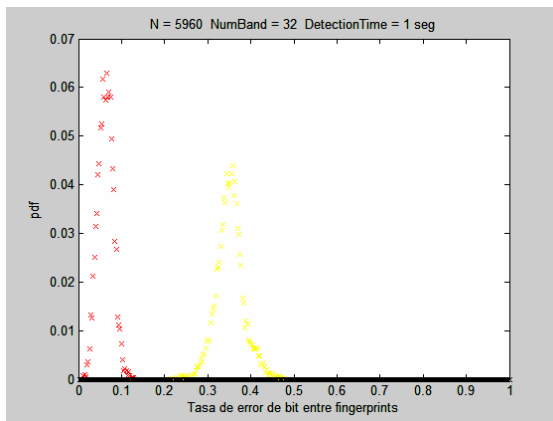
Veamos los resultados obtenidos al ejecutar esta prueba. En primer lugar se muestran en las sucesivas gráficas las funciones densidad de probabilidad para los casos más próximos a la solución considerada como óptima. En segundo lugar, en la tabla 11, se resumen todos los resultados obtenidos de la prueba, mostrándose la probabilidad de falsa alarma (P_f), la probabilidad de no alarma (P_n), el tiempo de computo de un *fingerprint* (T. C.) y el número de bits resultantes por cada *fingerprint* para cada posible combinación de los valores tomados por los parámetros.



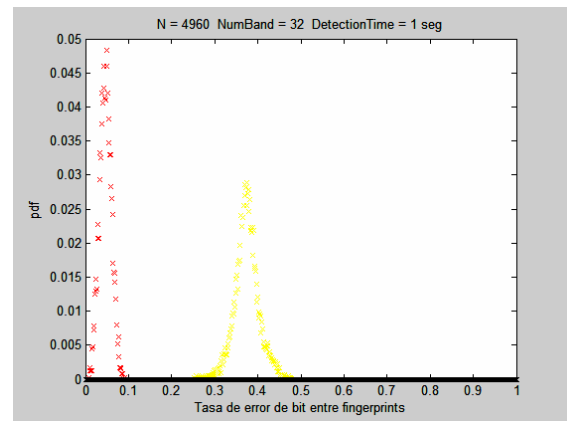
Gráfica 64. PDF Vs BER (N = 5960 N° Bandas = 8 T. Captura = 1 seg.)



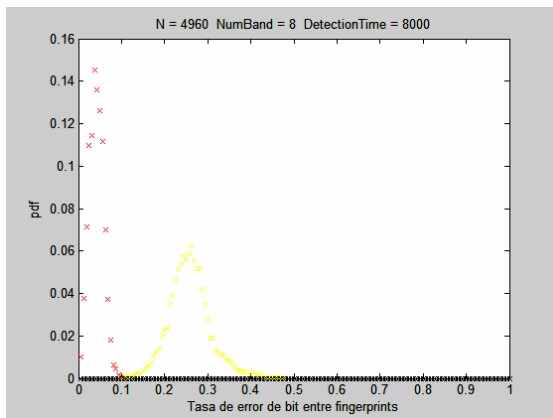
Gráfica 65. PDF Vs BER (N = 5960 N° Bandas = 16 T. Captura = 1 seg.)



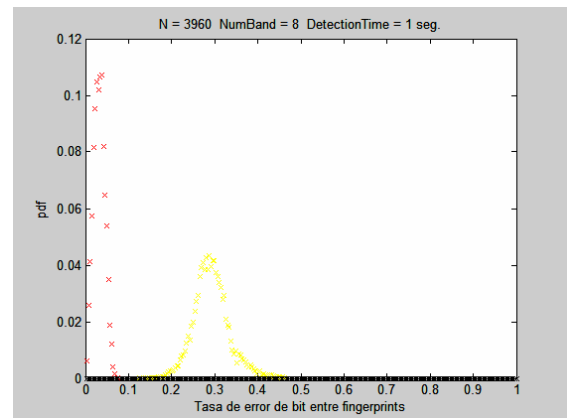
Gráfica 66. PDF Vs BER (N = 5960 N° Bandas = 32 T. Captura = 1 seg.)



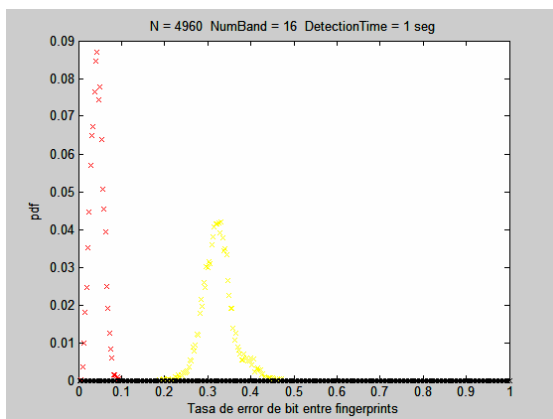
Gráfica 69. PDF Vs BER (N = 4960 N° Bandas = 32 T. Captura = 1 seg.)



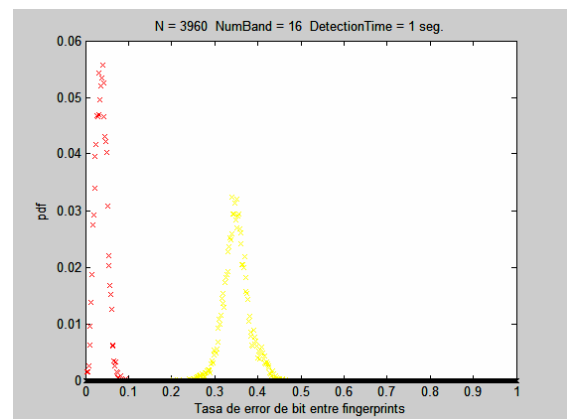
Gráfica 67. PDF Vs BER (N = 4960 N° Bandas = 8 T. Captura = 1 seg.)



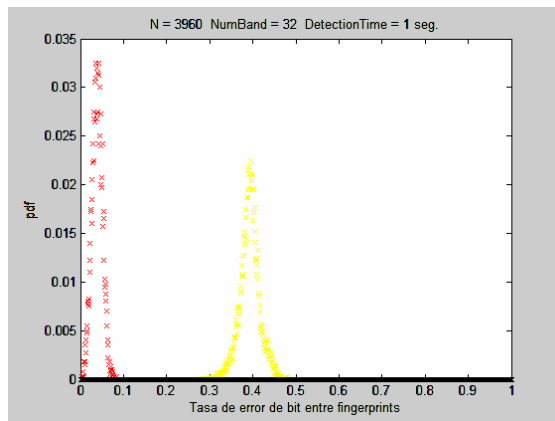
Gráfica 70. PDF Vs BER (N = 3960 N° Bandas = 8 T. Captura = 1 seg.)



Gráfica 68. PDF Vs BER (N = 4960 N° Bandas = 16 T. Captura = 1 seg.)



Gráfica 71. PDF Vs BER (N = 3960 N° Bandas = 16 T. Captura = 1 seg.)



Gráfica 72. PDF Vs BER (N = 3960 N° Bandas = 32 T. Captura = 1 seg.)

Tiempo de captura = 1 segundo		Factor de Overlap = 31/32		
N	N° Bandas	8	16	32
7960		$P_f = 9.6710e-001$ $P_n = 1.2730e-002$ T. Cómputo = 0.0052 N° bits = 1 x 8	$P_f = 6.2354e-001$ $P_n = 2.1959e-001$ T. Cómputo = 0.0056 N° bits = 1 x 16	$P_f = 9.2146e-002$ $P_n = 5.0372e-001$ T. Cómputo = 0.0061 N° bits = 1 x 32
6960		$P_f = 5.0259e-001$ $P_n = 4.4438e-002$ T. Cómputo = 0.0171 N° bits = 5 x 8	$P_f = 4.0436e-002$ $P_n = 1.5079e-002$ T. Cómputo = 0.0185 N° bits = 5 x 16	$P_f = 3.7443e-005$ $P_n = 1.1666e-002$ T. Cómputo = 0.0216 N° bits = 5 x 32
5960		$P_f = 2.6165e-001$ $P_n = 6.6133e-006$ T. Cómputo = 0.0380 N° bits = 11 x 8	$P_f = 3.2029e-003$ $P_n = 1.1191e-013$ T. Cómputo = 0.0409 N° bits = 11 x 16	$P_f = 5.5073e-008$ $P_n = 0$ T. Cómputo = 0.0469 N° bits = 11 x 32
4960		$P_f = 5.4895e-002$ $P_n = 9.1704e-014$ T. Cómputo = 0.0468 N° bits = 20 x 8	$P_f = 2.2186e-005$ $P_n = 0$ T. Cómputo = 0.0547 N° bits = 20 x 16	$P_f = 4.4363e-012$ $P_n = 0$ T. Cómputo = 0.0645 N° bits = 20 x 32
3960		$P_f = 3.3985e-003$ $P_n = 0$ T. Cómputo = 0.0377 N° bits = 33 x 8	$P_f = 5.7351e-009$ $P_n = 0$ T. Cómputo = 0.0639 N° bits = 33 x 16	$P_f = 4.1132e-020$ $P_n = 0$ T. Cómputo = 0.0843 N° bits = 33 x 32
2960		$P_f = 8.6658e-006$ $P_n = 0$ T. Cómputo = 0.0798 N° bits = 55 x 8	$P_f = 7.5339e-016$ $P_n = 0$ T. Cómputo = 0.0957 N° bits = 55 x 16	$P_f = 7.5067e-037$ $P_n = 0$ T. Cómputo = 0.1258 N° bits = 55 x 32

Tabla 11. Evolución de la P_f , la P_n , el tiempo de cómputo y el número de bits por *fingerprint* en función del número de muestras por trama y el número de bandas espectrales.

Entre las diferentes combinaciones que aparecen en la tabla 11, la celda sombreada en amarillo muestra aquella que minimiza en mayor grado tanto el tiempo de cómputo como el número de bits resultantes por *fingerprint* al mismo tiempo que presenta una probabilidad de falsa alarma suficientemente baja.

Por lo tanto, como conclusión de esta prueba, aceptamos como valores que optimizan el algoritmo de extracción de *fingerprints* los siguientes:

- Número de muestras por trama: **4960 muestras**
- Factor de overlap: **31/32**
- Número de bandas espectrales: **32 bandas**
- Frecuencia inferior para la división en bandas: **300 Hz**
- Frecuencia superior para la división en bandas: **2000 Hz**
- Tiempo de captura de la señal de entrada: **1 segundo**

lo cual implica asumir lo siguiente:

$$P_n = 0$$

$$P_f = 4.4363 \cdot 10^{-12}$$

$$\text{Tiempo de cómputo de 1 fingerprint} = 0,0645 \text{ segundos}$$

$$\text{Número de bits por fingerprint} = 20 \times 32 = 640 \text{ bits}$$

CONCLUSIÓN DE LAS TRES PRUEBAS:

Una vez realizadas las tres pruebas partiendo de cada uno de los valores óptimos para cada una de las características, tenemos que tomar de las tres soluciones posibles la más óptima, o bien una solución intermedia entre las tres que llegue a ser aún más óptima.

Para establecer una solución final, comparamos las tres soluciones obtenidas atendiendo a los siguientes criterios:

- menor tiempo de cómputo de un *fingerprint*
- y menor número de bits resultantes por *fingerprint*.

y bajo las siguientes condiciones:

- probabilidad de falsa alarma (P_f) del orden de 10^{-12} o inferior,
- probabilidad de no alarma (P_n) del orden de 10^{-12} o inferior,

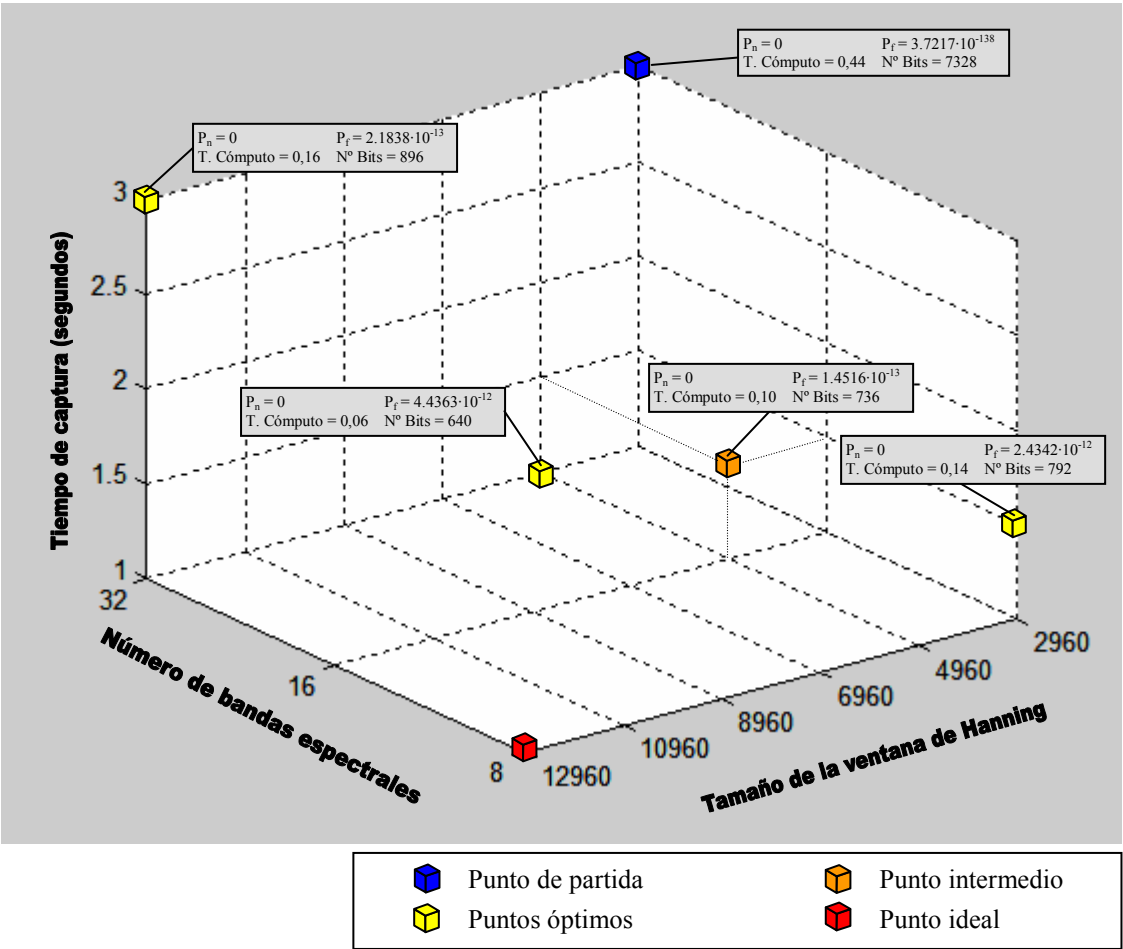
Veamos en la gráfica 73 un resumen de las pruebas realizadas en este apartado. Se muestra el punto de partida, es decir los valores de partida del diseño original (*punto azul*); los puntos óptimos obtenidos con las pruebas anteriores en las que se fija una característica y se varían las otras dos (*puntos amarillos*); una solución intermedia que, en primera instancia, se podría pensar que es la solución más óptima (*punto naranja*); y la solución ideal, es decir aquella que está formada por los tres valores óptimos obtenidos para cada característica en las pruebas previas (*punto rojo*). Como ya hemos comentado con anterioridad, este punto ideal no es alcanzable porque, primero, los valores que implica la solución no son compatibles entre sí y, segundo, porque las probabilidades de no alarma y de falsa alarma no serían tolerables para un buen funcionamiento del sistema.

Por lo tanto, el objetivo es encontrar el punto que más se acerque a esa solución ideal manteniendo unas probabilidades de no alarma y de falsa alarma que se puedan considerar aceptables.

De las tres soluciones obtenidas con las pruebas, es decir de los tres puntos amarillos mostrados en la gráfica 73, el que obtiene mejores resultados es el relativo a la tercera prueba, aquella en la que se parte de 1 segundo para el tiempo de captura de la señal de entrada al sistema y se varían tanto el tamaño de la ventana de Hanning como el número de bandas espectrales para la división de cada trama. En este punto tenemos un tiempo de cómputo para un *fingerprint* de 0,06 segundos con un tamaño de bits por *fingerprint* de 640 bits; en las otras dos posibles soluciones (los otros dos puntos amarillos) se obtienen peores valores para el tiempo de cómputo y para el número de bits por *fingerprint*.

A partir de este punto nos desplazamos hacia el punto ideal tratando de buscar soluciones que optimicen aún más el algoritmo de extracción de

fingerprint. Sin embargo, de entre todas las posibilidades que se puedan plantear alrededor de este punto, ninguna cumple con los requisitos de probabilidades de error, excepto la solución correspondiente al punto naranja. No obstante, este punto naranja no consigue mejores prestaciones que las obtenidas con el punto amarillo al que nos referíamos antes.



Gráfica 73. Comparativa entre diferentes soluciones

En esta gráfica no se muestran los resultados obtenidos para puntos alrededor del punto amarillo considerado como solución más óptima y que no cumplen con los requisitos de probabilidades de error. Para estos puntos intermedios entre las tres soluciones que se obtuvieron con los apartados previos, se han hecho pruebas particularizadas cuyos resultados se pueden ver en la tabla 12.

Características	P_n	P_f	T. Cómputo (miliseg.)	Nº Bits
N = 6960 Nº Bandas = 16 T. Captura = 1	$1,5079 \cdot 10^{-02}$	$4,0436 \cdot 10^{-02}$	19	80
N = 5960 Nº Bandas = 16 T. Captura = 1	$1,1191 \cdot 10^{-13}$	$3,2029 \cdot 10^{-03}$	41	176
N = 6960 Nº Bandas = 16 T. Captura = 1,5	0	$9,3094 \cdot 10^{-07}$	64	384
N = 5960 Nº Bandas = 16 T. Captura = 1,5	0	$2,1970 \cdot 10^{-09}$	96	528
N = 4960 Nº Bandas = 16 T. Captura = 1,5	0	$1,4516 \cdot 10^{-13}$	102	736

Tabla 12. Resultados para valores intermedios entre las tres soluciones óptimos

Una vez analizadas todas las posibilidades que se pudieran considerar como óptimas para el algoritmo de extracción de *fingerprints*, concluimos que los valores adoptados para cada una de las características que optimizan este algoritmo son los siguientes:

- Número de muestras por trama: **4960 muestras**
- Factor de overlap: **31/32**
- Número de bandas espectrales: **32 bandas**
- Frecuencia inferior para la división en bandas: **300 Hz**
- Frecuencia superior para la división en bandas: **2000 Hz**
- Tiempo de captura de la señal de entrada: **1 segundo**

lo cual implica asumir lo siguiente:

$$P_n = 0$$

$$P_f = 4.4363 \cdot 10^{-12}$$

Tiempo de cómputo de 1 fingerprint = 0,0645 segundos

Número de bits por fingerprint = $20 \times 32 = 640$ bits

4.4. PRUEBAS DE ROBUSTEZ

En este apartado se muestran los experimentos de robustez, con sus correspondientes resultados, realizados sobre el algoritmo de extracción de *fingerprint* propuesto en la conclusión del apartado anterior. Con estas pruebas pretendemos resolver la cuestión de si la tasa de error de bit (BER) entre el *fingerprint* procedente de un anuncio contenido en nuestra base de datos y el *fingerprint* correspondiente a este anuncio extraído de una señal de audio de entrada al sistema que se encuentra degradada se mantiene por debajo del umbral de detección que hemos establecido ($\alpha = 0.18$). Esta degradación de la señal puede ser debida a ruido que haya en el canal o por cualquier tipo de procesado que sufra la señal.

❑ Degradación por ruido blanco gaussiano (AWGN)

En esta prueba se estudia por encima de que relación señal a ruido (SNR) nuestro sistema funciona correctamente. Para ello, en el código de Matlab **p_robustez_SNR.m** (véase Anexo 2.24) se simula una señal de audio de entrada al sistema degradada por un ruido blanco gaussiano cuya energía dependerá de los diferentes valores de SNR estudiados. Para los diferentes valores de SNR (desde 5 dB hasta 60 dB) se calcula la tasa de error de bit media y su desviación estándar entre el *fingerprint* extraído y los *fingerprints* correspondientes a las cuatro grabaciones del mismo anuncio. Para cada valor de SNR se repite el proceso 1000 veces, para tener así resultados estadísticos fiables. También se estudia la evolución de la probabilidad de no alarma (P_n) y la probabilidad de falsa alarma (P_f) en función de la SNR.

Como cabía esperar, y como verifican los resultados obtenidos de esta prueba de robustez, la probabilidad de falsa alarma no se ve alterada por el ruido del canal simulado, pues este ruido únicamente contribuye a aleatorizar los bits que conforman un *fingerprint*. Dado que la probabilidad de falsa alarma se calcula con los *fingerprints* correspondientes a los anuncios que no deberían ser detectados con el *fingerprint* extraído, la tasa de error de bit entre el extraído y los correspondientes a los anuncios que no guardan ninguna relación con este no se ve alterada.

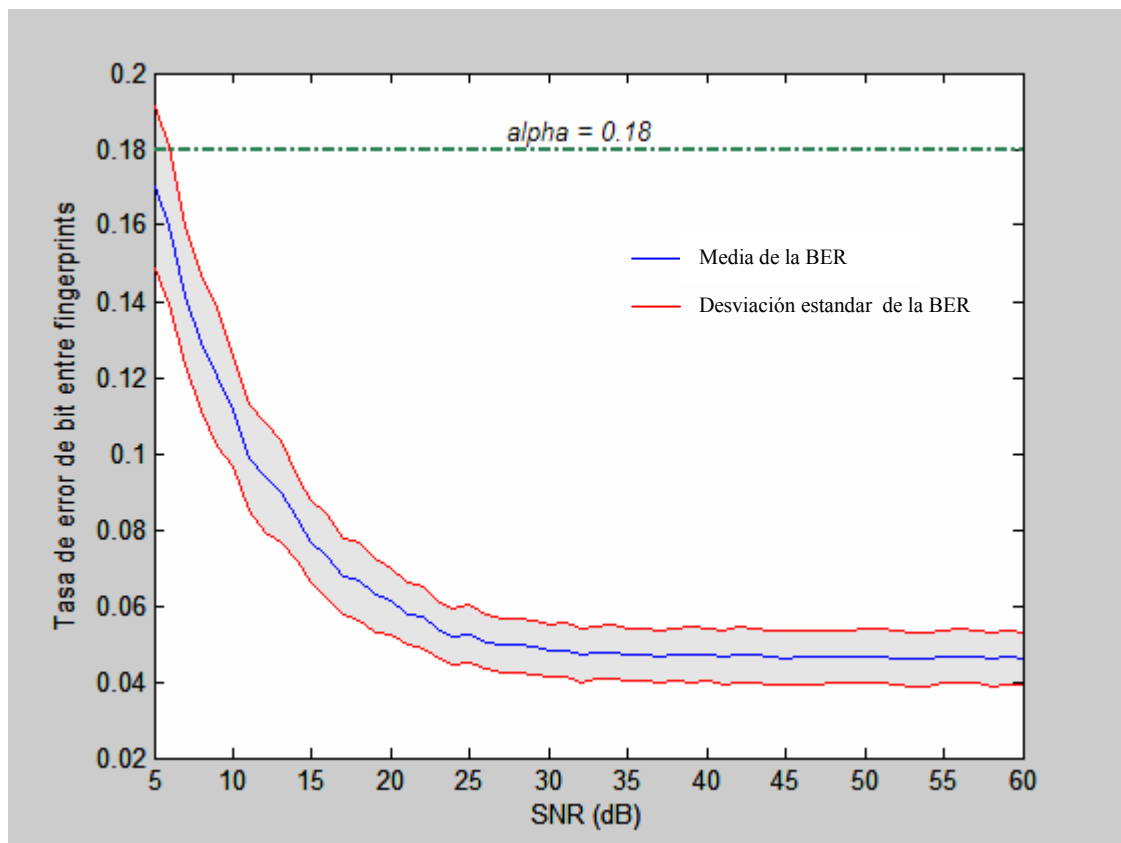
En cuanto a la probabilidad de no alarma, se ve muy afectada por el ruido introducido, pues este ruido, si tiene una energía tal que implica una SNR superior a un valor que luego veremos, aleatoriza los bits del *fingerprint* extraído en tal medida que disminuye la similitud (la distancia de Hamming) entre este *fingerprint* extraído y los *fingerprints* del anuncio al que se corresponde.

En la gráfica 74 se muestra la evolución de la tasa de error de bits entre el *fingerprint* extraído de la señal degradada y los *fingerprints* correspondientes

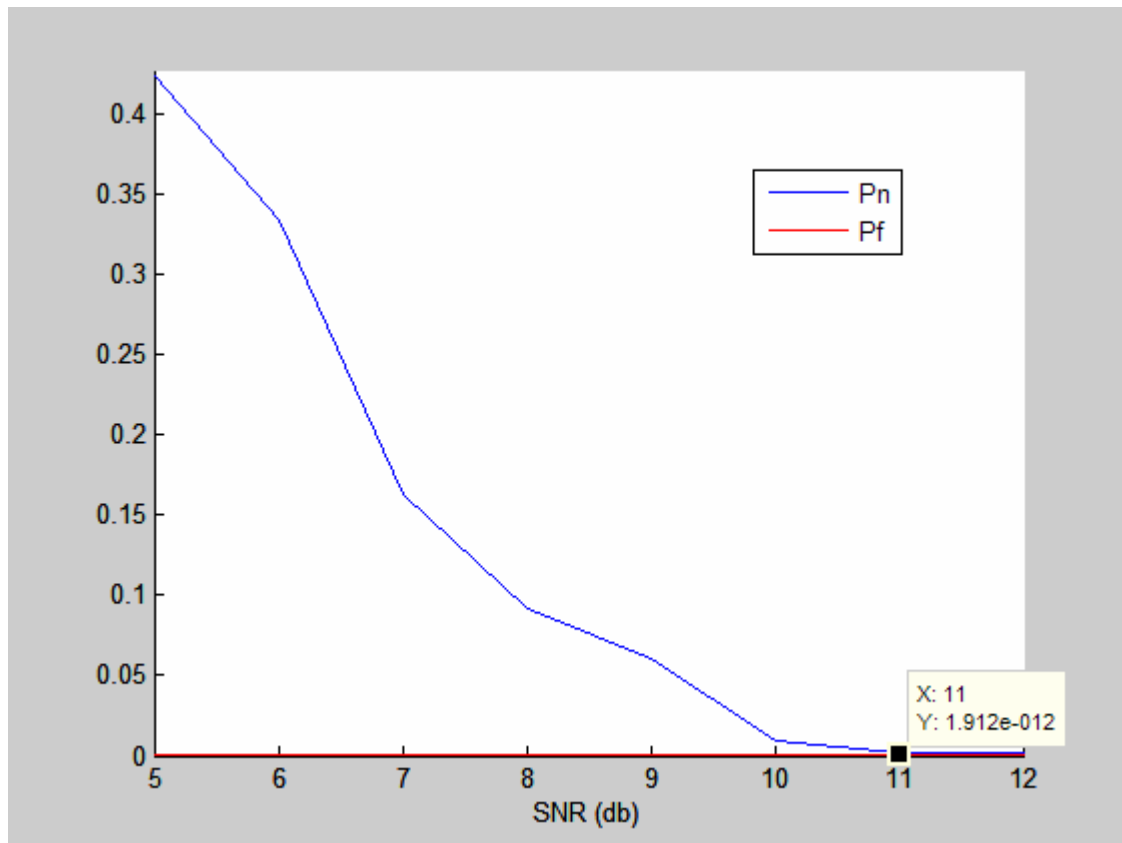
a las 4 versiones diferentes que poseemos del anuncio que se toma como referencia. En azul se representa la media de la BER, mientras que en azul se representa la desviación estándar de esta BER. Como podemos observar, a partir de una SNR de unos 7 dB la BER obtenida es inferior a 0.18, que es el umbral que hemos adoptado para considerar detección.

Por otra parte, en la gráfica 75, donde se muestra la evolución de la probabilidad de no alarma (línea azul) en función de la SNR, vemos que a partir de valores de SNR superiores a 11 dB la P_n es del orden de 10^{-12} o inferior (requisito que hemos tomado para considerar que la probabilidad de error es tolerable).

También en la gráfica 75 se muestra la dependencia de la probabilidad de falsa alarma (línea roja) con la relación señal a ruido. En cuanto a esta probabilidad, como ya dijimos anteriormente, no se ve afectada por el ruido introducido, manteniéndose en el orden de magnitud que ya habíamos obtenido por diseño (una P_f del orden de 10^{-12}).



Gráfica 74. BER Vs SNR



Gráfica 75. P_n Vs SNR y P_f Vs SNR

En conclusión de este análisis de robustez, obtenemos que nuestro sistema es robusto a partir de una relación señal a ruido superior a 11 dB.

❑ Degradación por diferentes procesados de la señal

Para este bloque de pruebas vamos a trabajar con cuatro anuncios de radio (Mono, 8000 Hz., 8 bits/muestra): "*Bacardi Limón*", "*Viajes Marsans*", "*Carrefour*" y "*La ONCE*". Sobre cada uno de ellos se realizarán diferentes procesados y se compararán 1000 *fingerprints* extraídos por cada anuncio procesado con los respectivos *fingerprints* de los anuncios sin procesar. Con estas 1000 comparaciones por anuncio obtendremos la media y la desviación estándar de la BER entre dos *fingerprints*. Así verificaremos si nuestro sistema es robusto o no ante los diferentes procesados propuestos. Consideraremos que el sistema funciona correctamente si la media de la BER es inferior al umbral $\alpha = 0.18$.

Para realizar las diferentes pruebas, previamente se simulará el procesado de la señal con el programa "*Adobe Audition 2.0*", cuya apariencia general se muestra en la figura 8, y luego se ejecutará el código de matlab **p_robustez_PROCESADO.m** (véase Anexo 2.25) realizando pequeños cambios según el procesado que se quiera estudiar. Básicamente estos cambios serán modificar los archivos de audio que recogen el procesado que se desea estudiar.

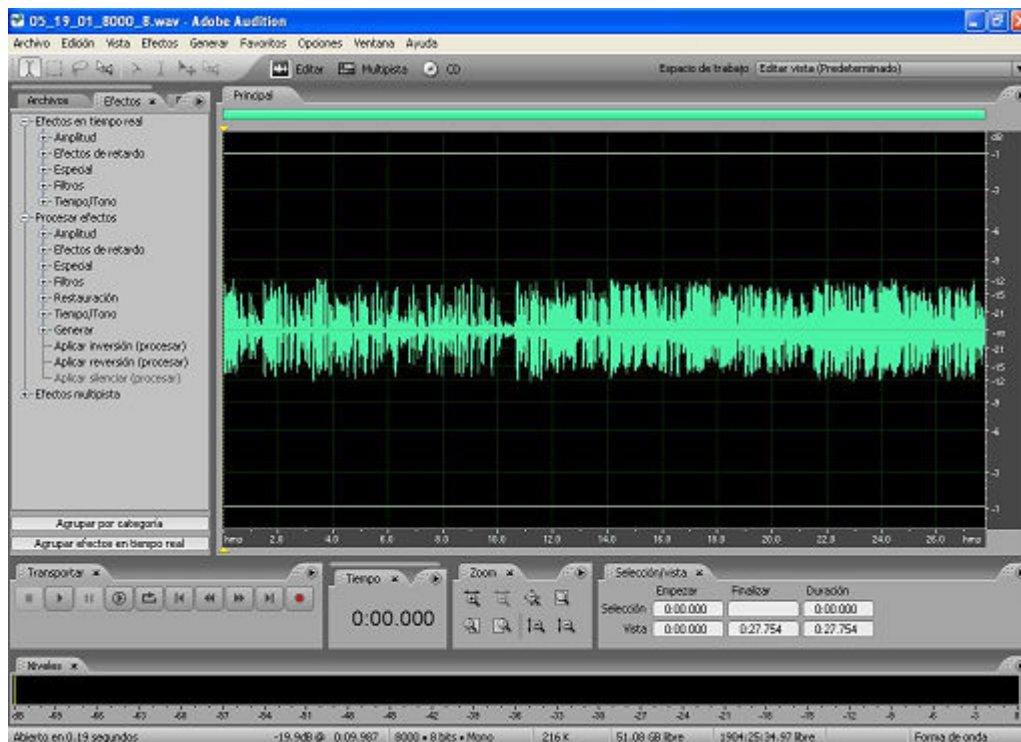


Figura 8. Apariencia general de Adobe Audition 2.0

Veamos los diferentes procesos a los que han sido sometidos los anuncios:

- **Codificación/decodificación MP3** con codificación de velocidad de bits variable, de modo que se utiliza velocidades de bits superiores para

el material más complejo e inferiores para el material más sencillo. Se han utilizado los niveles de calidad 10 (calidad baja, 40-50 Kbps), 50 (calidad media, 105-140 Kbps) y 100 (calidad superior, 150-240 Kbps).

- **Codificación/decodificación Windows Media Audio 9.1** con codificación de velocidad de bits constante a 5 Kbps y 6 Kbps. El resto de parámetros son los mismos para ambas pruebas: 8 KHz., 16 bits, mono.

- **Filtro pasa todo** con función de transferencia
$$H(z) = \frac{0.81z^2 - 1.64z + 1}{z^2 - 1.64z + 0.81}$$

Para realizar este procesamiento añadiremos al código de Matlab, haciendo uso de la función "filter" proporcionada por este programa, la siguiente línea que filtra la señal de audio según la función de transferencia expresada anteriormente:

```
anuncio_detectado1=filter([0.81,-1.64,1],[-1.64,0.81],anuncio1);
```

- **Compresión de Amplitud** con las siguientes tasas de compresión: 8.94:1 para $|A| \geq -28.6$ dB; 1.73:1 para -46.4 dB $< |A| < -28.6$ dB; 1:1.61 para $|A| \leq -46.4$ dB.

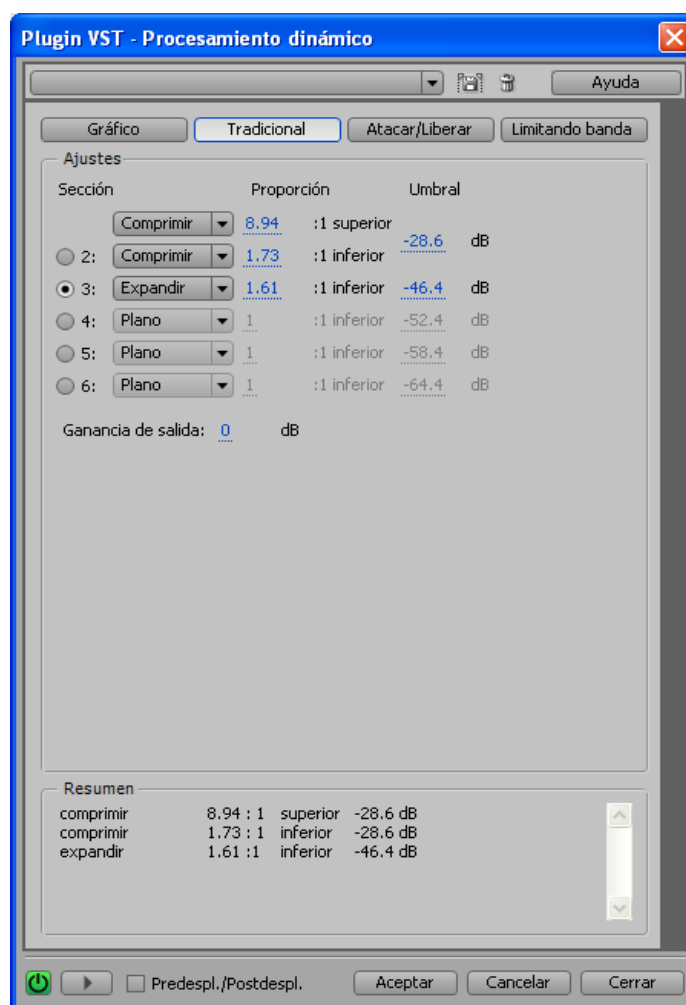


Figura 9. Compresión de Amplitud en Adobe Audition

- **Ecualización:** Un ecualizador típico de 10 bandas con los siguientes parámetros:

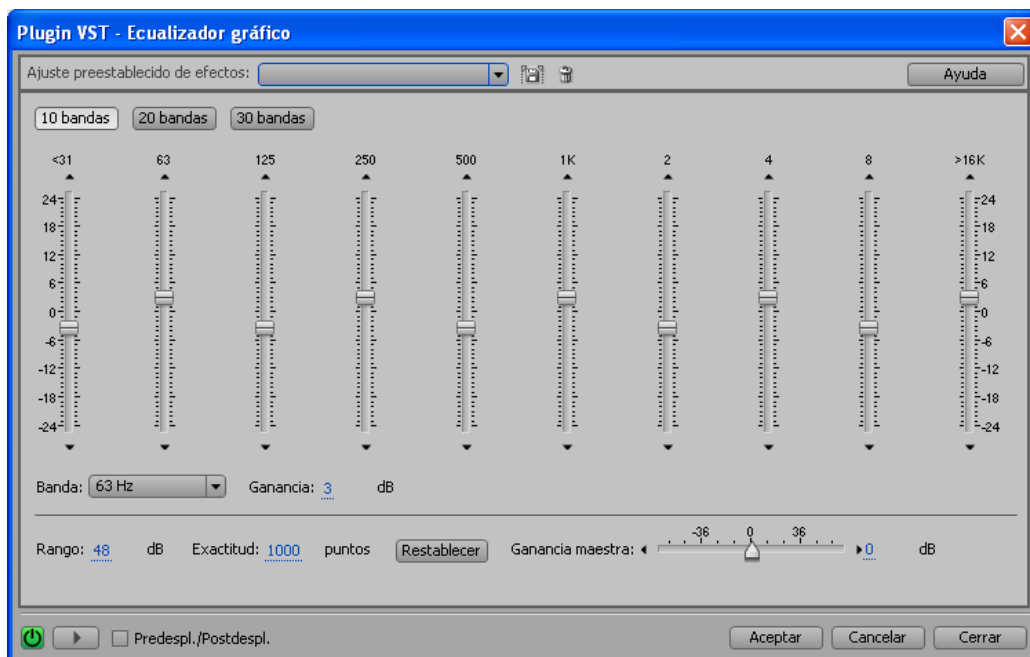


Figura 10. Ecualizador de 10 bandas

- **Filtrado paso de banda** usando un filtro Butterworth de segundo orden con frecuencias de corte de 100 Hz y 6000 Hz.

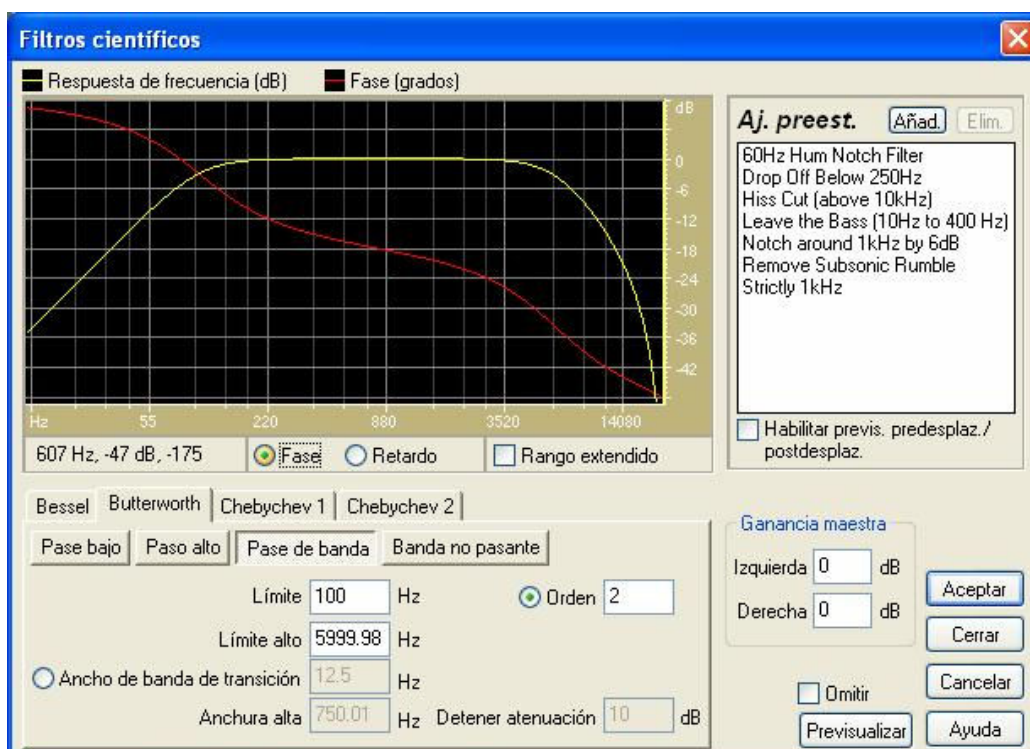


Figura 11. Filtro Butterworth Paso de Banda

- **Modificación de la escala temporal** en un -2%, +2%, -4% y +4%. Solo cambia el tiempo, el tono permanece inalterado.

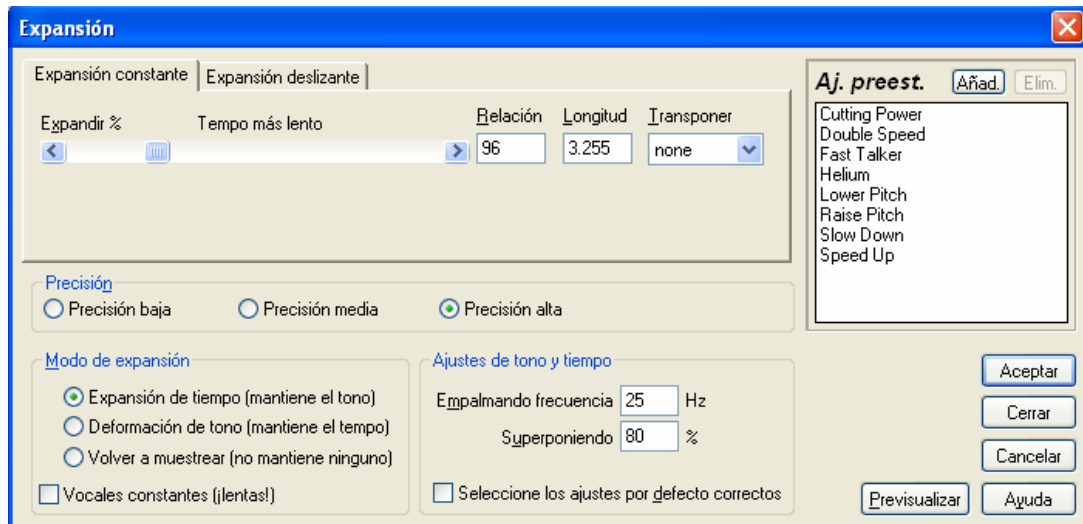


Figura 12. Expansión temporal de un -4% en Adobe Audition

- **Cambio lineal de la velocidad** en un -1%, +1%, -4% y +4%. Cambian tanto el tono como el tiempo. En el *Adobe Audition* se realiza con el mismo menú que para la expansión temporal, señalando en modo de expansión la opción "Volver a muestrear".

Después de realizar dichas transformaciones (por separado, a ningún anuncio se le hacen dos transformaciones distintas al mismo tiempo), se mide la BER entre *fingerprints* procedentes del fragmento original y los extraídos del fragmento degradado para cada una de los cuatro anuncios escogidos. En la tabla 13 se muestran la media y la desviación estándar obtenidos a partir de las 1000 comparaciones que se realizan por cada anuncio en cada prueba:

Procesado	Bacardi Limón		Viajes Marsans		Carrefour		La ONCE	
	Media	Desv.	Media	Desv.	Media	Desv.	Media	Desv.
MP3 con VBR nivel 10	0.0419	0.0130	0.0420	0.0125	0.0403	0.0120	0.0408	0.0127
MP3 con VBR nivel 50	0.0407	0.0135	0.0401	0.0123	0.0405	0.0120	0.0409	0.0128
MP3 con VBR nivel 100	0.0411	0.0134	0.0398	0.0125	0.0405	0.0119	0.0399	0.0129
WMA con CBR 5 Kbps	0.0416	0.0137	0.0411	0.0124	0.0403	0.0118	0.0413	0.0128
WMA con CBR 6 Kbps	0.0402	0.0123	0.0420	0.0125	0.0403	0.0120	0.0408	0.0127
Filtrado pasa todo	0.0708	0.0184	0.0633	0.0146	0.0635	0.0157	0.0615	0.0153
Comp. Amplitud	0.1166	0.0377	0.1115	0.0285	0.1049	0.0278	0.1343	0.0373
Ecualización	0.0630	0.0179	0.0582	0.0134	0.0577	0.0131	0.0570	0.0147
Filtrado paso banda	0.0536	0.0299	0.0491	0.0213	0.0496	0.0222	0.0566	0.0295
Esc. Tiempo -2%	0.0625	0.0209	0.0609	0.0174	0.0627	0.0229	0.0412	0.0133
Esc. Tiempo +2%	0.0675	0.0242	0.0752	0.0193	0.0651	0.0247	0.0657	0.0215
Esc. Tiempo -4%	0.0744	0.0221	0.0938	0.0226	0.0776	0.0234	0.0799	0.0233

Esc. Tiempo +4%	0.0796	0.0256	0.0886	0.0247	0.0788	0.0266	0.0814	0.0308
Veloc. Lineal -1%	0.1421	0.0313	0.1447	0.0328	0.1404	0.0319	0.1595	0.0401
Veloc. Lineal +1%	0.1388	0.0347	0.1415	0.0347	0.1218	0.0238	0.1507	0.0351
Veloc. Lineal -4%	0.3466	0.0243	0.3618	0.0278	0.3429	0.0311	0.3556	0.0256
Veloc. Lineal +4%	0.3430	0.0329	0.3608	0.0224	0.3386	0.0318	0.3492	0.0287

Tabla 13. Resultados de las pruebas de robustez con procesado

Vemos que sólo en los casos señalados en rojo se supera el umbral de error de 0.18. Corresponden con el caso de los cambios lineales de velocidad del 4%, tanto en un sentido como en el contrario. Esto es debido a la desviación del tramado (desalineamiento temporal) y al escalado espectral (desalineamiento frecuencial).

Por otra parte, los casos señalados en azul podemos considerarlos como críticos, pues, aún sin superar el umbral en valor medio, si observamos la media de la BER y su desviación estándar correspondiente deducimos que pudieran aparecer algunos casos en los que se superara el umbral de 0.18.

Tras analizar este conjunto de pruebas sobre robustez, podemos concluir que el algoritmo de extracción optimizado presenta un comportamiento bastante robusto ante posibles degradaciones que pudiera sufrir la señal de audio de entrada al sistema. Quizás se podría pensar en ser un poco menos ambicioso en los valores óptimos de los parámetros de configuración del algoritmo para obtener mejores resultados en cuanto a la robustez. Dependerá en gran medida de la aplicación para la que se quiera utilizar el sistema de detección de audio y del canal por el que nos llegue la señal de audio el hecho de que tomemos valores menos óptimos de los parámetros con el objetivo de obtener unas probabilidades de error de detección más pequeñas.