

9.- Desarrollo del GUI (Graphic User Interface)

En esta sección mostraremos la interfaz gráfica diseñada en MATLAB para la ejecución de todos los programas que se implementan en este proyecto. Con esta interfaz se persigue poder controlar, desde una misma pantalla, todos los análisis y obtener las conclusiones relativas a un experimento en concreto, de manera que el usuario pueda introducir ciertos parámetros variables y el resto del proceso sea transparente a él.

La ventana está estructurada de un modo muy simple. En primer lugar se divide en EEG y señales fisiológicas, como vemos a continuación. Tecleamos en la ventana de comandos de MATLAB la palabra “Monitor”, y aparece la siguiente pantalla:



Fig. 9.1: Pantalla principal del GUI (inicio).

Como podemos ver, la interfaz se ha diseñado de manera que sólo estén resaltadas las funciones que se pueden usar en cada momento, por ejemplo, sólo podremos previsualizar una señal cuando hayamos seleccionado previamente los datos en cuestión, y sólo podremos usar el clasificador después de ejecutar los programas correspondientes al procesamiento de cada una de las señales.

9.1.- Importación de datos y previsualización

Para ilustrar su funcionamiento, abriremos los datos pertenecientes al experimento definitivo (capítulo 8).



Fig. 9.1.1: Ventana para añadir datos provenientes de BCI2000.

Al agregar el fichero '.dat', se ejecuta un programa en MATLAB que procesa dicho archivo y convierte sus datos en variables del espacio de trabajo, listas para ser tratadas. Podemos ahora previsualizar algunas de estas señales:



Fig. 9.1.2: Pantalla principal del GUI (datos cargados).

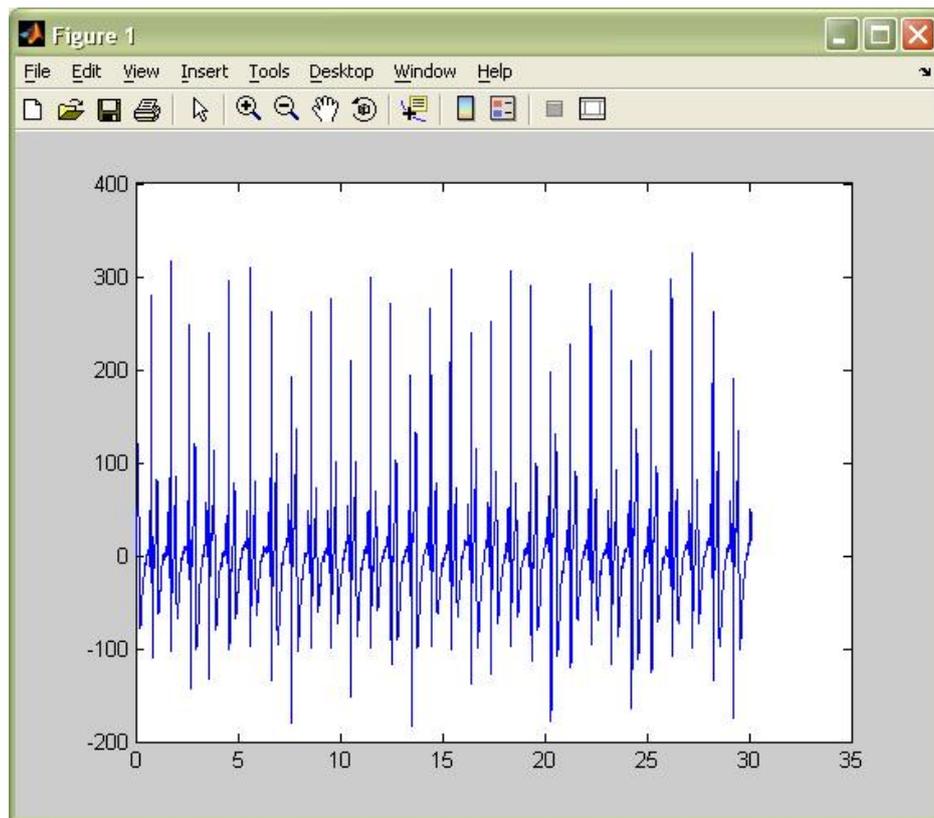


Fig. 9.1.3: Previsualización de la señal ECG.

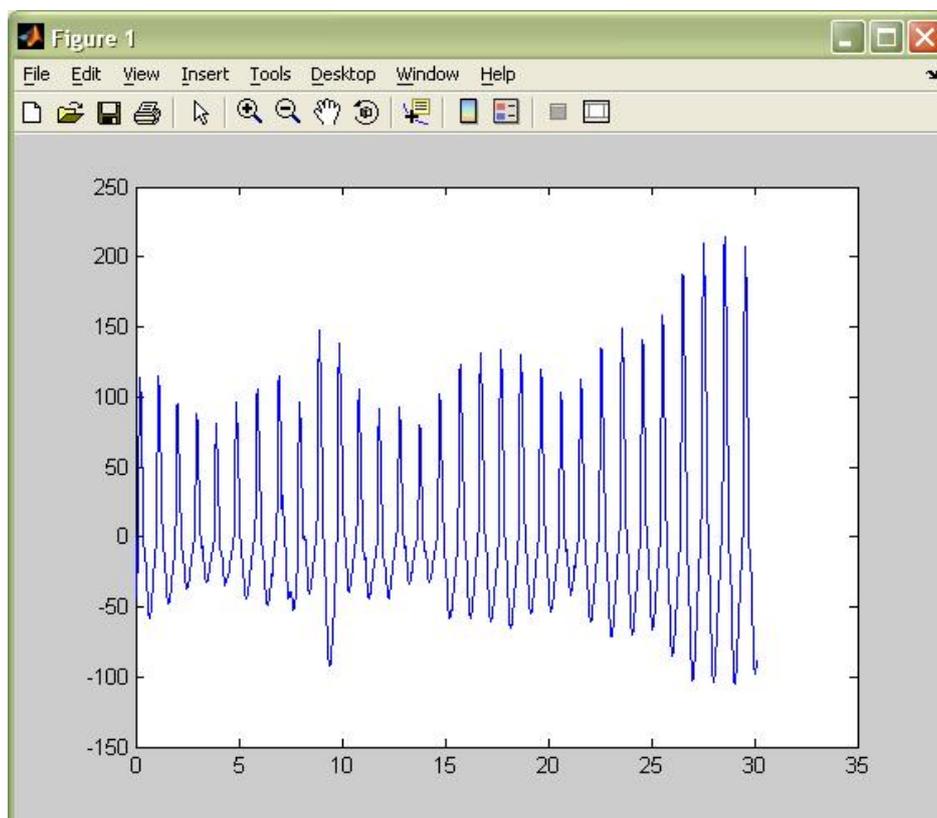


Fig. 9.1.4: Previsualización de la señal BVP.

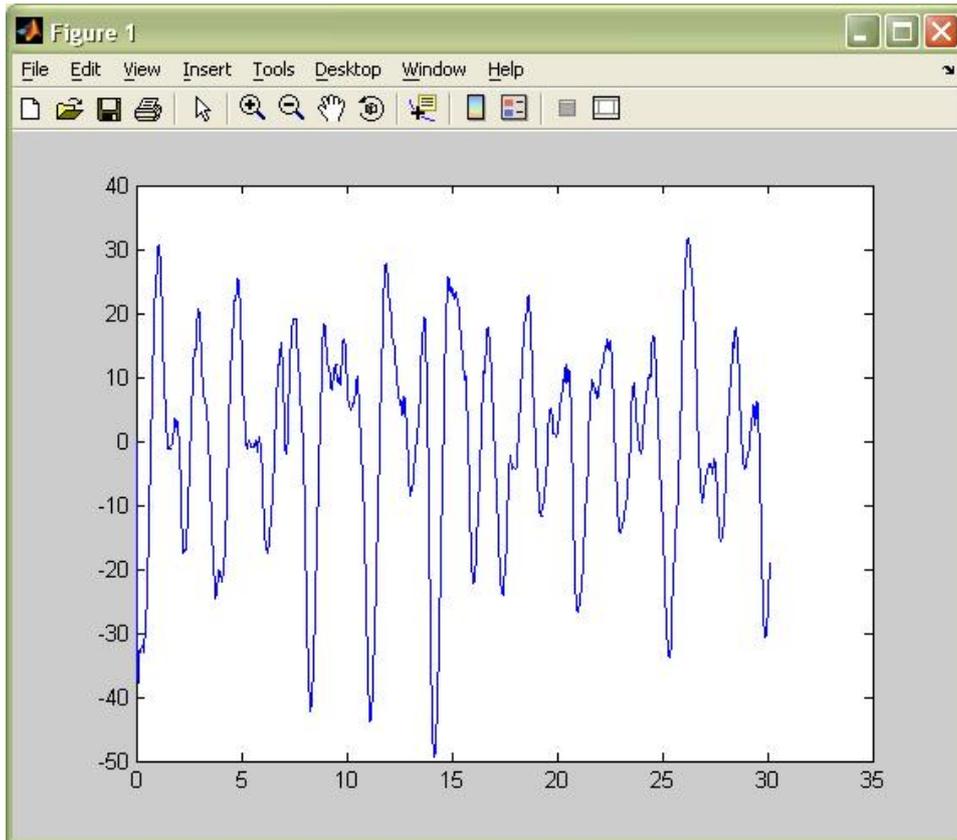


Fig. 9.1.5: Previsualización de la señal RESP.

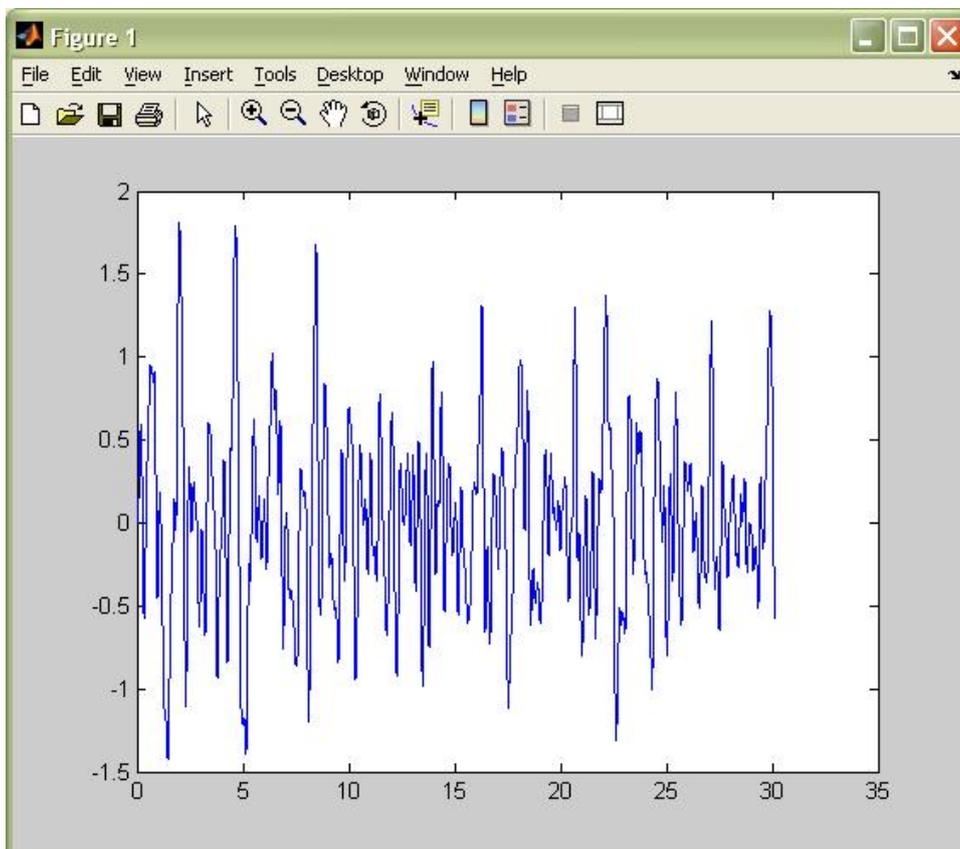


Fig. 9.1.6: Previsualización de la señal SC.

9.2.- Biosensores: Procesado

A continuación lanzamos el programa que procesará la señal ECG. Como resultado nos mostrará unas gráficas con el HR medio y su varianza en cada intervalo, así como el HRV. Además, colocará los datos resultantes del tratamiento de la señal en una matriz cuyas filas se irán rellenando conforme vayamos lanzando el resto de biosensores. Nótese que, por este motivo, el botón del clasificador no estará disponible hasta que se hayan ejecutado todos ellos, y la matriz esté completa.

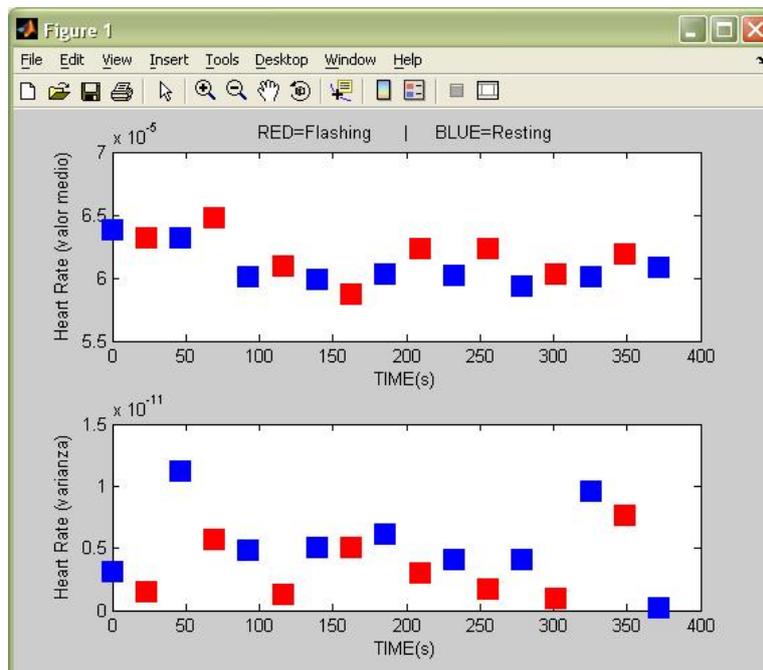


Fig. 9.2.1: Valor medio y varianza del HR en cada intervalo.

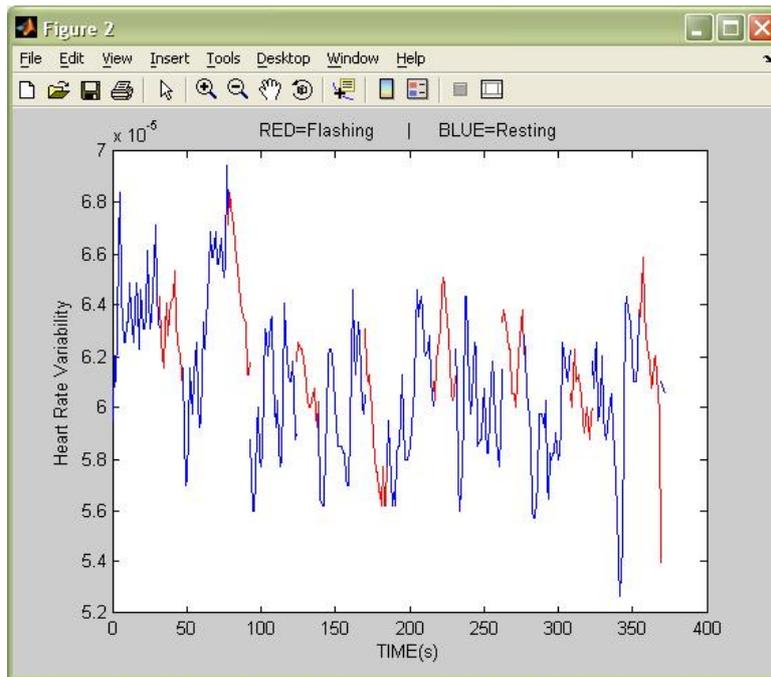


Fig. 9.2.2: HRV en todos los intervalos del experimento.

De igual modo, si pulsamos el botón ‘Lanzar BVP’, se mostrarán las gráficas del HR en media y varianza en cada intervalo y del HRV obtenidos a través del sensor BVP, así como de la dilatación vascular o “pinch”.

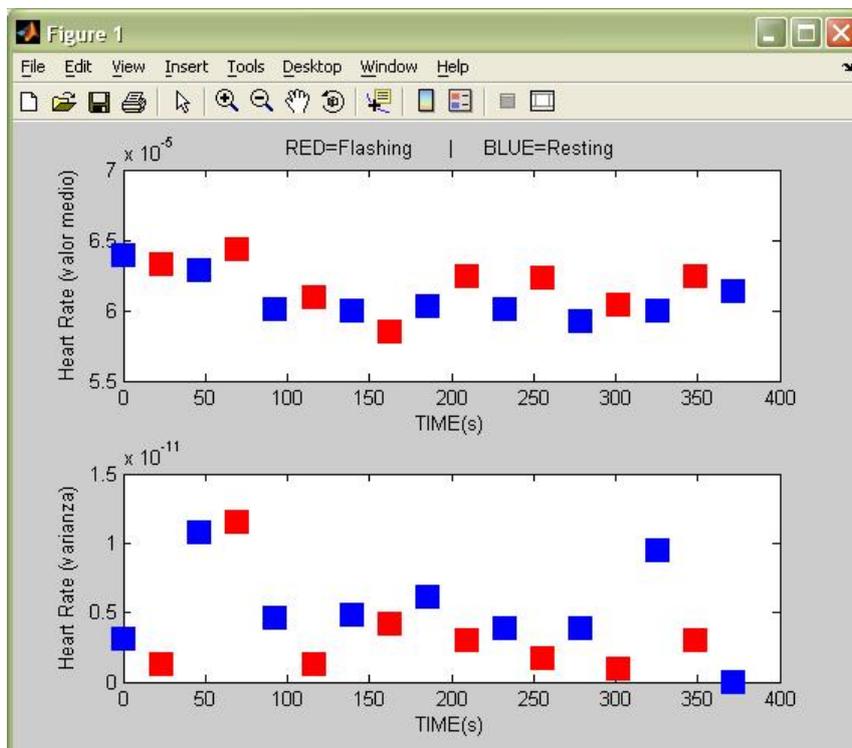


Fig. 9.2.3: Valor medio y varianza del BVP-HR en cada intervalo.

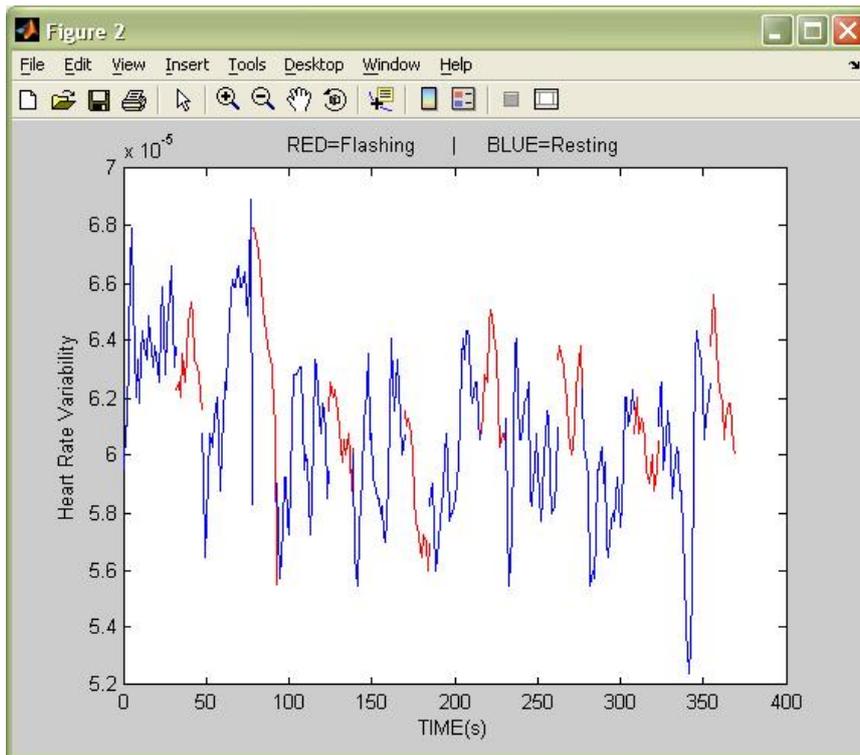


Fig. 9.2.4: BVP-HRV en todos los intervalos del experimento.

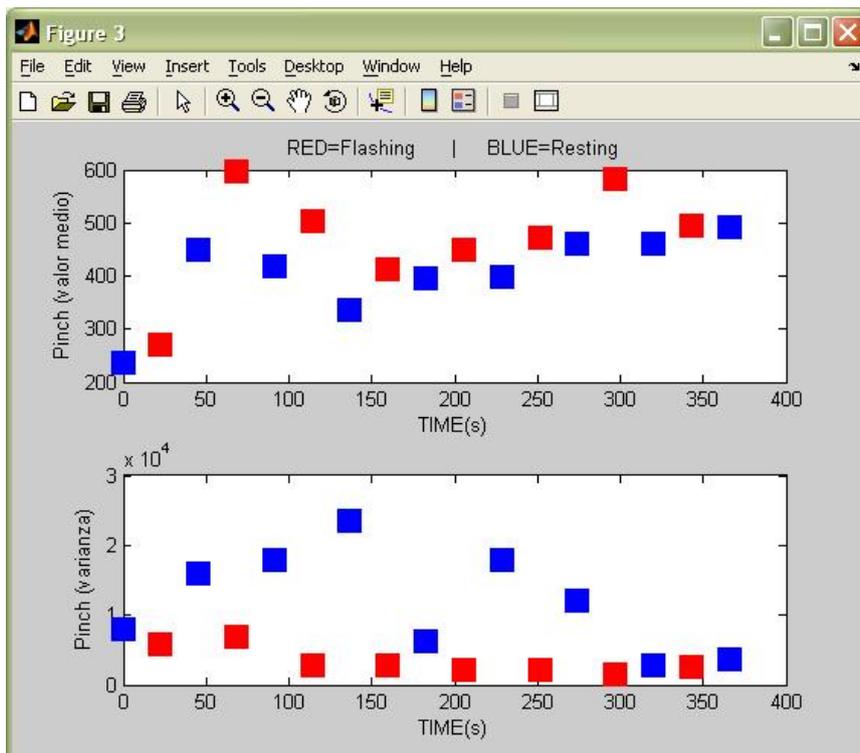


Fig. 9.2.5: Valor medio y varianza del “Pinch” en cada intervalo.

Al lanzar el biosensor RESP obtenemos las gráficas de la variable RPM (respiraciones por minuto) y PROF (profundidad de la respiración) en media y varianza en cada intervalo.

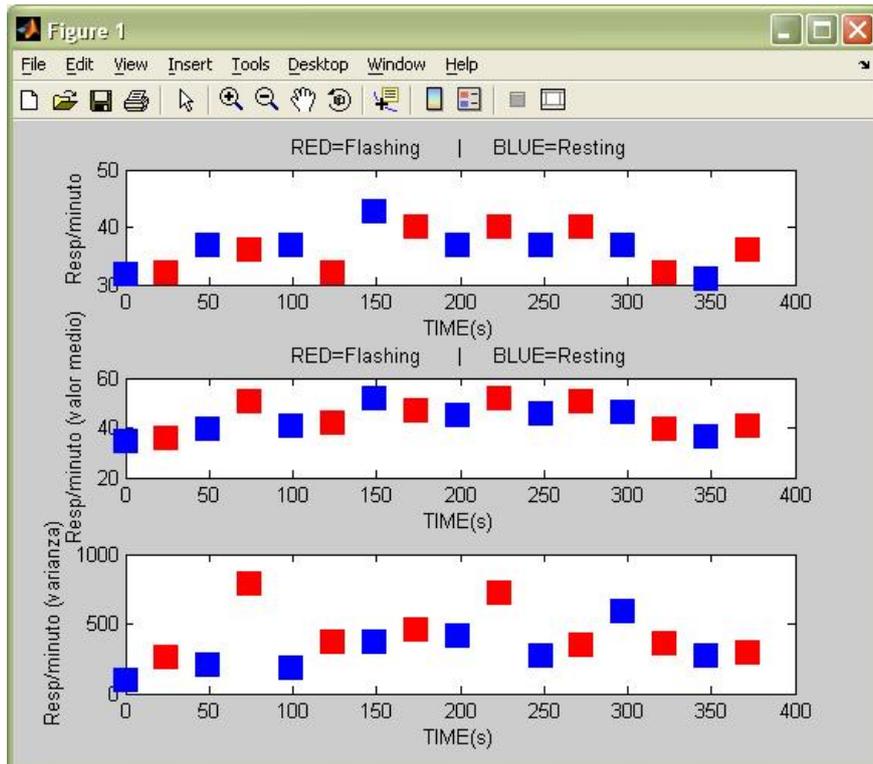


Fig. 9.2.6: Respiraciones por minuto. Media y varianza en cada intervalo.

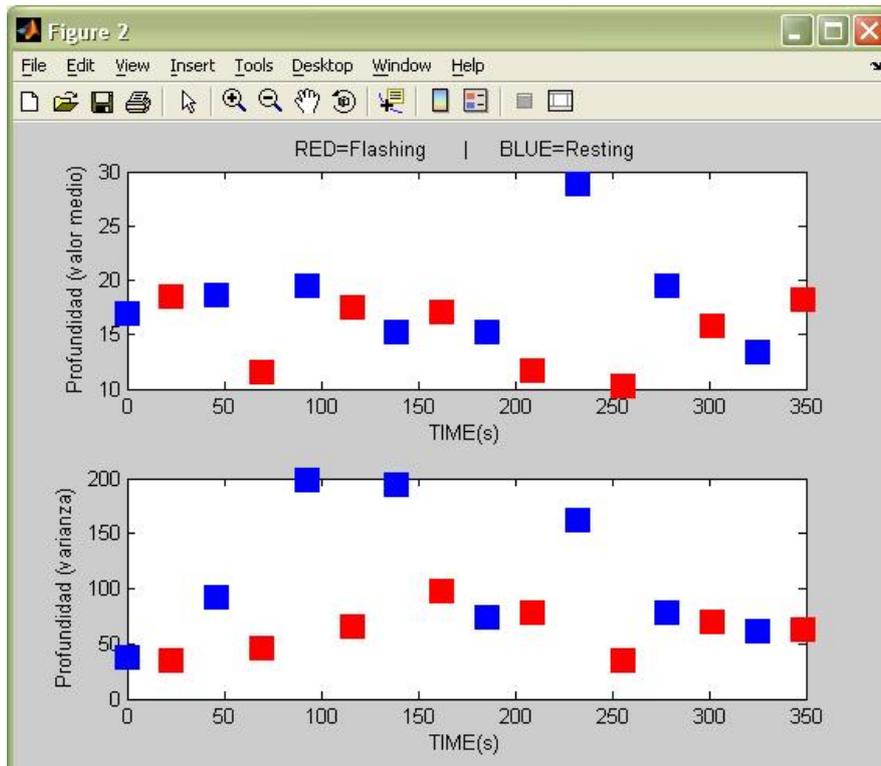


Fig. 9.2.7: Profundidad de la respiración en media y varianza en cada intervalo.

Pulsamos el botón ‘Lanzar SC’ y se nos muestra por pantalla una gráfica con la media y varianza de la señal, a la vez que se habilita el clasificador. Los números que vemos abajo son el resultado de la suma integral de las varianzas en cada intervalo.

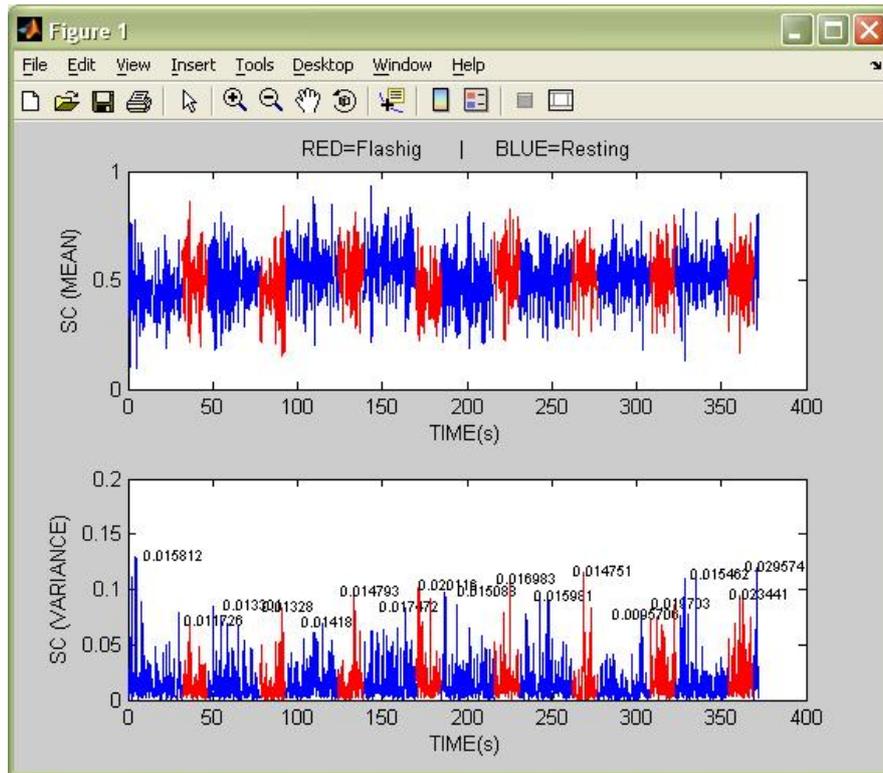


Fig. 9.2.8: Media y varianza de la señal SC.



Fig. 9.2.9: Menú principal con la función del clasificador habilitada.

9.3.- Biosensores: Tendencias

En cuanto a las tendencias (última columna del menú), cada botón nos mostrará una gráfica en la que se realiza una regresión a las variables que hemos visto anteriormente, lo que nos puede servir para, cualitativamente, evaluar las hipótesis relativas al estado de agotamiento. En las siguientes imágenes visualizamos dichas gráficas (rojo, actividad; azul, descanso; verde, tendencia general).

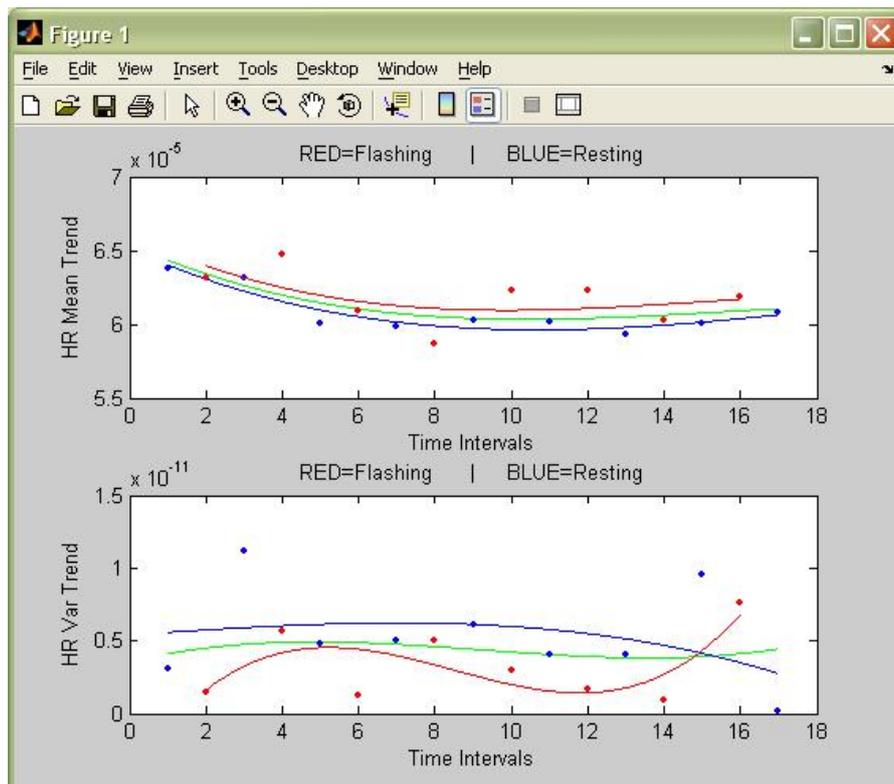


Fig. 9.3.1: Análisis de tendencias para el HR en media y varianza.

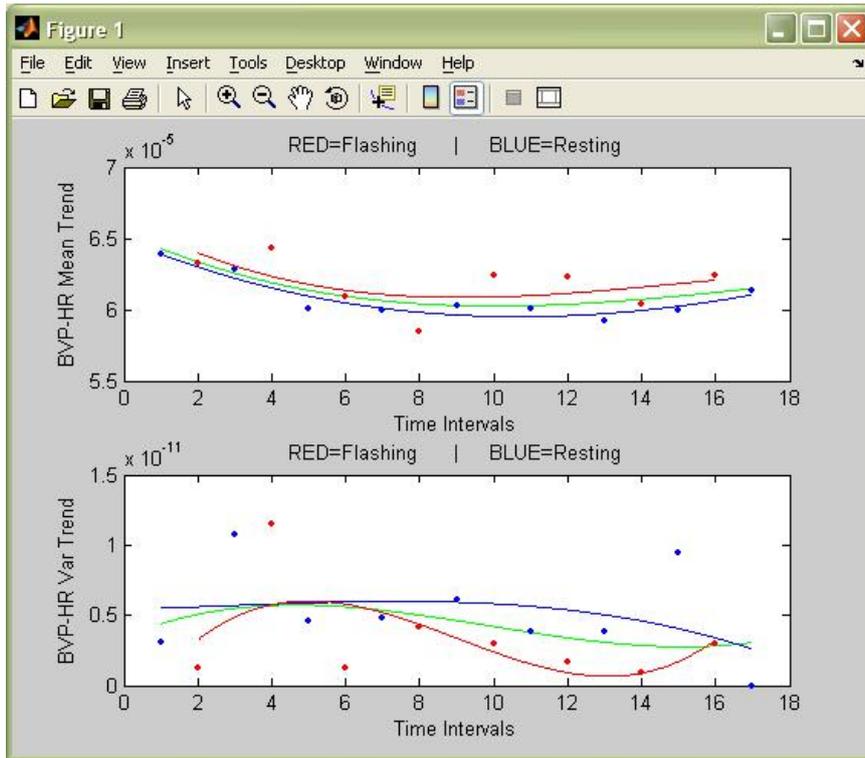


Fig. 9.3.2: Análisis de tendencias para el BPV-HR en media y varianza.

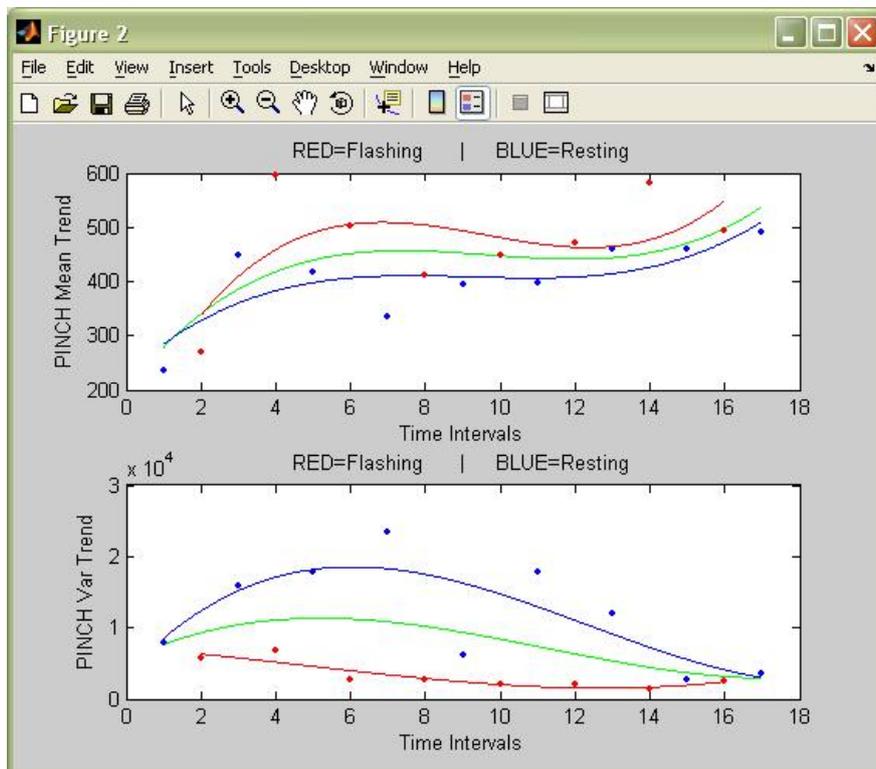


Fig. 9.3.3: Tendencias para la dilatación vascular en media y varianza.

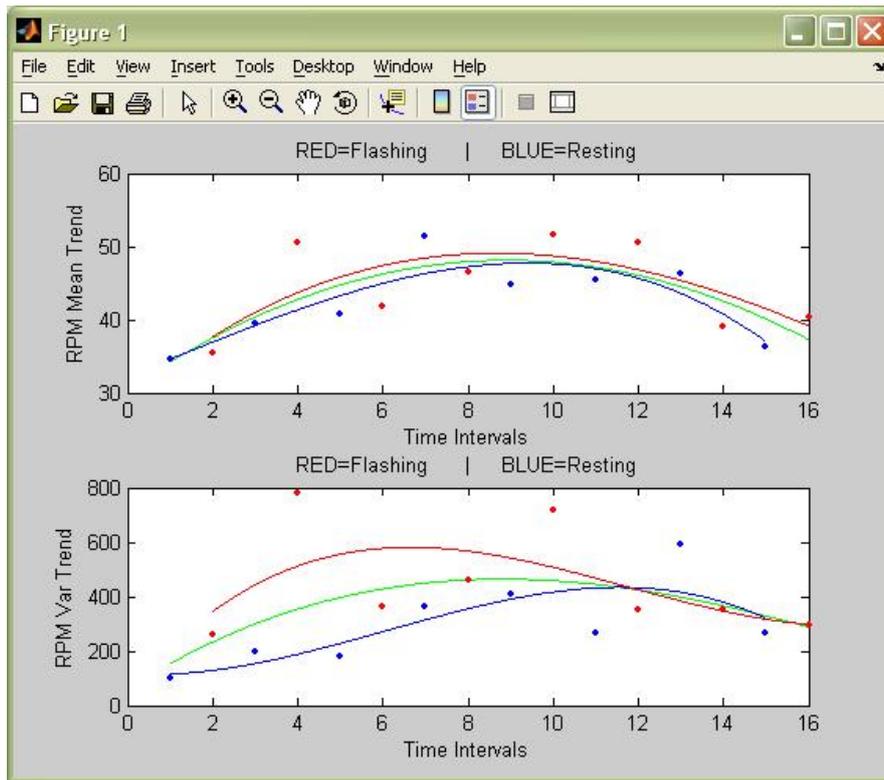


Fig. 9.3.4: Tendencias para el RPM en media y varianza.

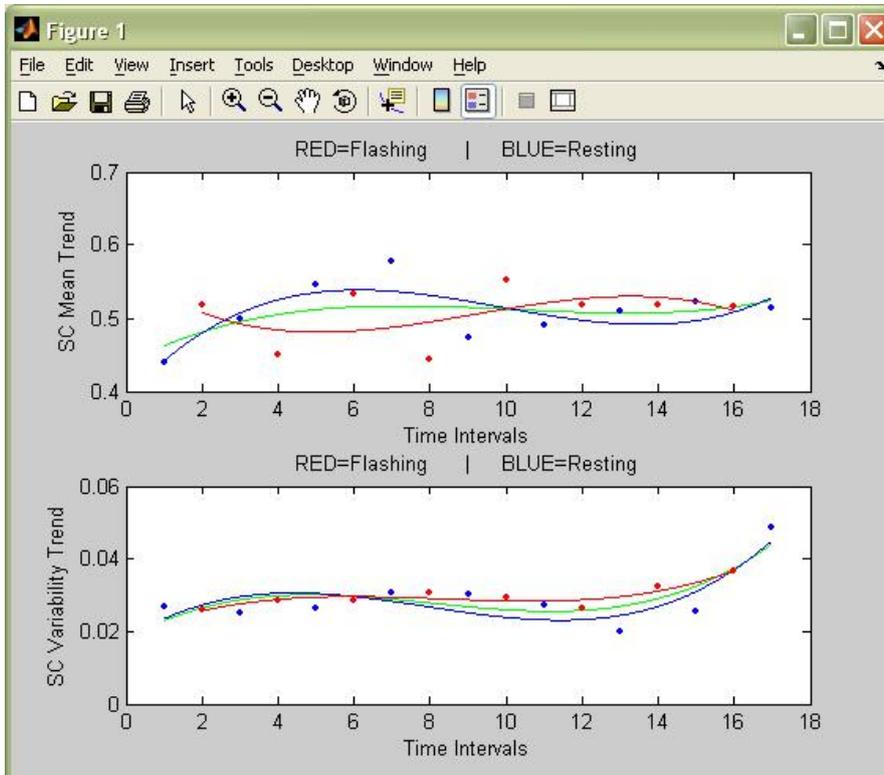


Fig. 9.3.5: Tendencias para el SC en media y varianza.

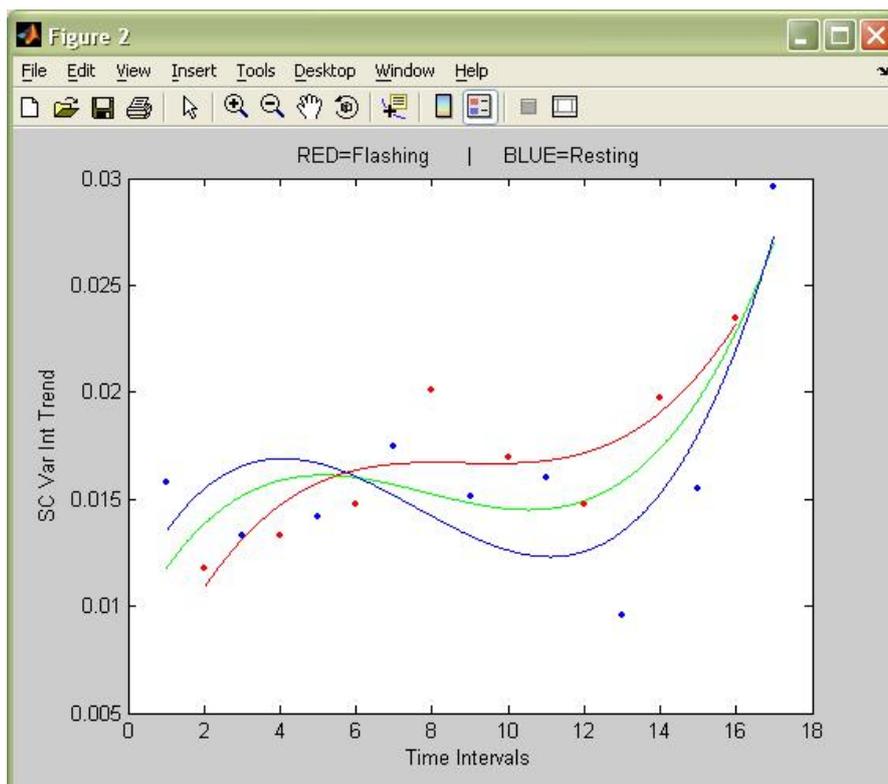


Fig. 9.3.6: Tendencia de la integración de las varianzas del SC en cada intervalo.

9.4.- EEG: Procesado

Pasaremos ahora a la otra parte de la interfaz, la que ejecutará los programas correspondientes al estudio del EEG. Si pulsamos el botón 'lanzar EEG' nos aparecerá un mensaje en la línea de comandos que nos preguntará el tamaño de ventana temporal deseado (número de muestras). Esto sirve para calcular las FFT de los canales del EEG, de manera que cuanto mayor sea la ventana, mayor número de muestras tendrán las señales que se transformarán, y tendremos un menor número de transformadas resultantes. En este caso, por ejemplo, trabajaremos con un tamaño de ventana de 300 muestras (a una frecuencia de muestreo de 256 Hz, esto supondría una ventana de $300/256 = 1,171$ segundos).

Una vez obtenidas las FFT, se guardan, para cada canal, en un vector las FFT pertenecientes a los estados de 'Flashing' y en otro las de 'Resting', y se procede a implementar el algoritmo r^2 para averiguar en qué canales se producen las mayores diferencias en cuanto a la varianza entre dichos estados. Acto seguido se representa el espectrograma, para que se pueda observar cualitativamente los canales que contengan más información.

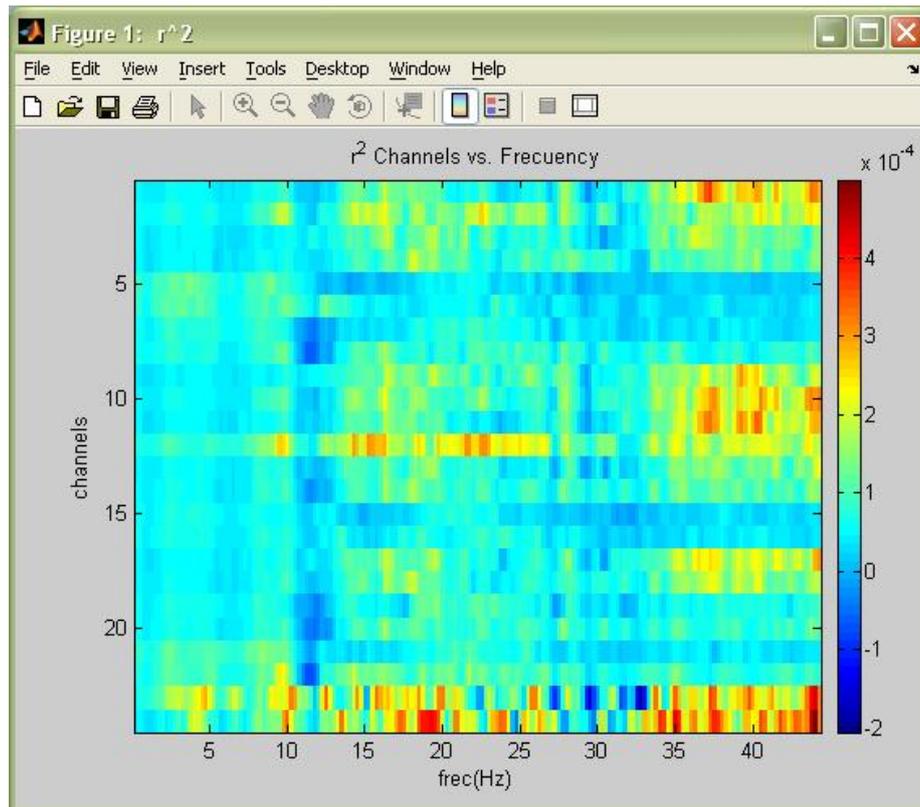


Fig. 9.4.1: Representación del espectrograma para todos los canales del EEG.

Si pulsamos el botón izquierdo del ratón en un punto cualquiera del espectrograma, se nos mostrará una representación topográfica (mapeo) del r^2 , es decir, podremos ver dónde están situados los canales que aportan una mayor cantidad de información para una frecuencia determinada de la FFT.

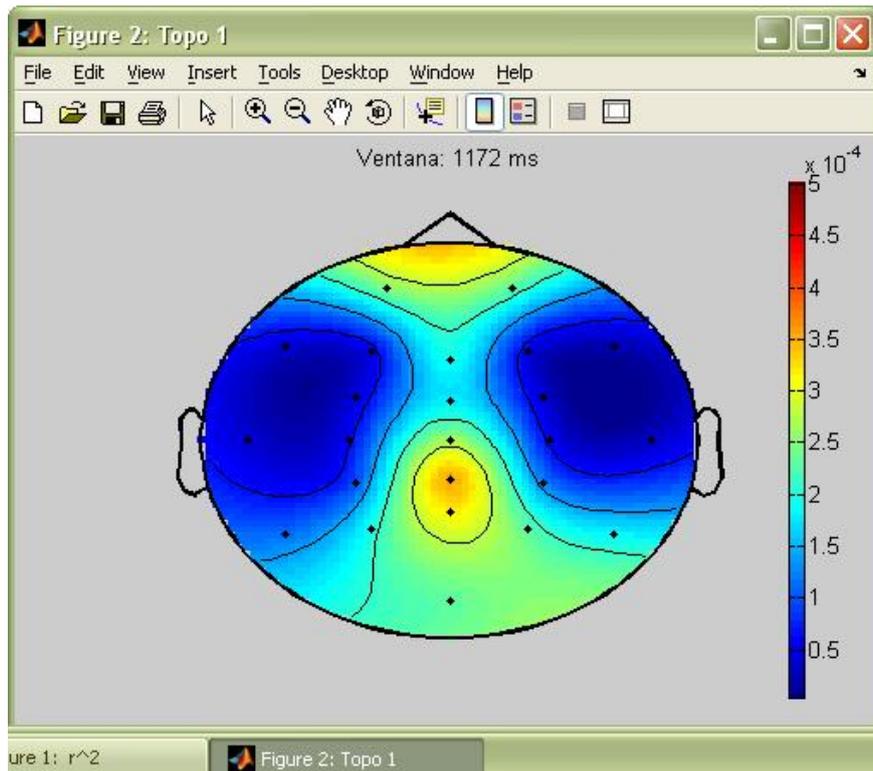


Fig. 9.4.2: Ejemplo de representación topográfica del r^2 .

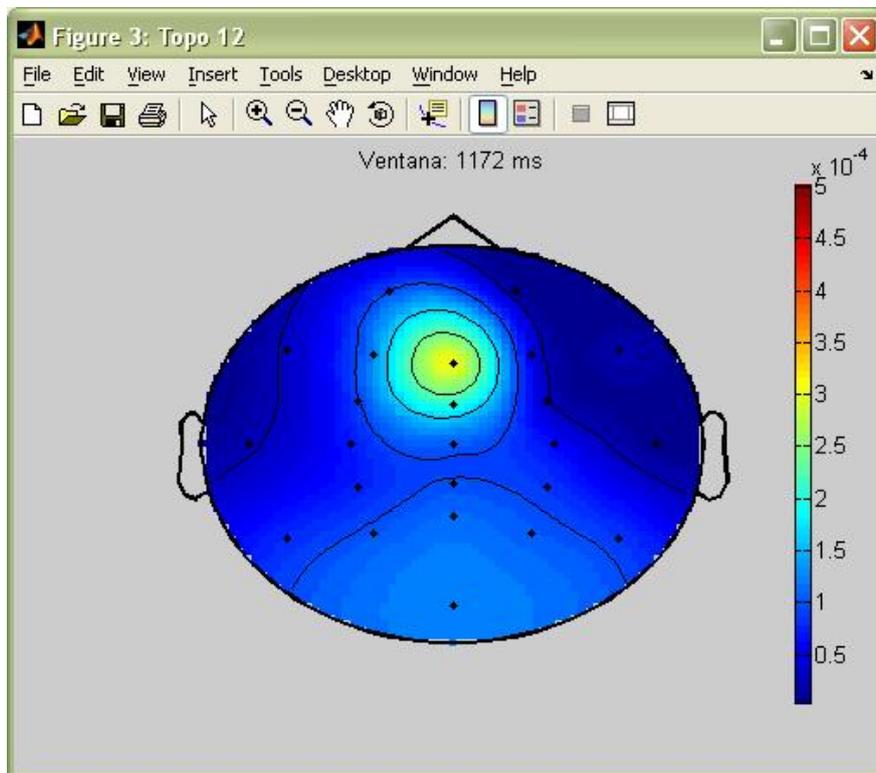


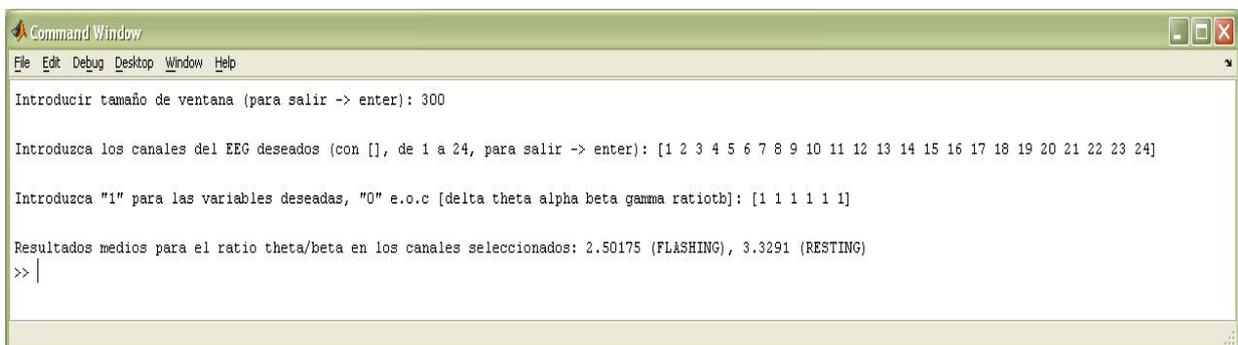
Fig. 9.4.3: Ejemplo de representación topográfica del r^2 .

A continuación se nos preguntará, en la línea de comandos, qué canales queremos seleccionar para construir la matriz que se dará como entrada al clasificador de estados. En este momento podemos descartar aquellos canales que no nos proporcionen información relevante para efectuar la clasificación, y así aligerar la carga de datos que tendrá que manejar el software. Como respuesta, pondremos entre corchetes los números de los canales elegidos. En este caso, a modo de ejemplo, hemos seleccionado todos los canales.

Asimismo, podemos elegir las bandas de frecuencia que queramos que entren a formar parte del estudio (recordemos, en la introducción, el capítulo dedicado a los ritmos electroencefalográficos, 1.2), así como el ratio theta / beta. Esta selección se hará colocando entre corchetes un '1' ó un '0' si queremos que la correspondiente variable sea tenida en cuenta. Como ejemplo, elegiremos todas las bandas.

Posteriormente, y según nuestras indicaciones, se preparará la matriz de entrada a la red neuronal que actuará como clasificador. Además, calculará la media de todos los ratios theta / beta para períodos de actividad y descanso y nos la mostrará en la línea de comandos (aunque no se menciona en el capítulo 4, referido al estudio de la atención, cabe comentar aquí que según las fuentes bibliográficas los valores normales del ratio theta / beta para un usuario sano son entre 2 y 3, cuando presta atención a un estímulo, es decir, lo que correspondería a un período de actividad, y mayores de 3 cuando no la presta, es decir, cuando descansa).

Mostramos la línea de comandos de MATLAB para ilustrar los párrafos anteriores:



```
Command Window
File Edit Debug Desktop Window Help
Introducir tamaño de ventana (para salir -> enter): 300
Introduzca los canales del EEG deseados (con [], de 1 a 24, para salir -> enter): [1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24]
Introduzca "1" para las variables deseadas, "0" e.o.c [delta theta alpha beta gamma ratioth]: [1 1 1 1 1]
Resultados medios para el ratio theta/beta en los canales seleccionados: 2.50175 (FLASHING), 3.3291 (RESTING)
>> |
```

Fig. 9.4.4: Mensajes en la línea de comandos tras el procesamiento del EEG.

Nótese que tras el procesado del EEG y los biosensores, ya tenemos disponible en nuestra pantalla principal el clasificador particular para cada uno de ellos, así como el clasificador total que toma como entrada los datos provenientes de ambos:



Fig. 9.4.5: Menú principal con la función de los clasificadores EEG y Total habilitada.

9.5.- EEG: Atención y ‘Workload’

Pero antes de correr los distintos clasificadores, abordaremos el estudio de la atención y la carga de trabajo o ‘workload’, accionando el botón que le corresponde.

Para empezar, para adecuar este proceso a los estudios en que nos basamos se dividirá el cerebro en tres regiones longitudinales: frontal, central y posterior, de manera que cada región incluirá los datos de un grupo de canales, los que correspondan a sensores que estén situados en dicha región.

Se calcularán las potencias medias de las FFT para cada banda y en total, así como los ratios theta / beta, y se representarán en cada región y para cada estado (cada gráfica tiene 6 filas: frontal, central y posterior para ‘Flashing’; frontal, central y posterior para ‘Resting’. Cada columna es una FFT).

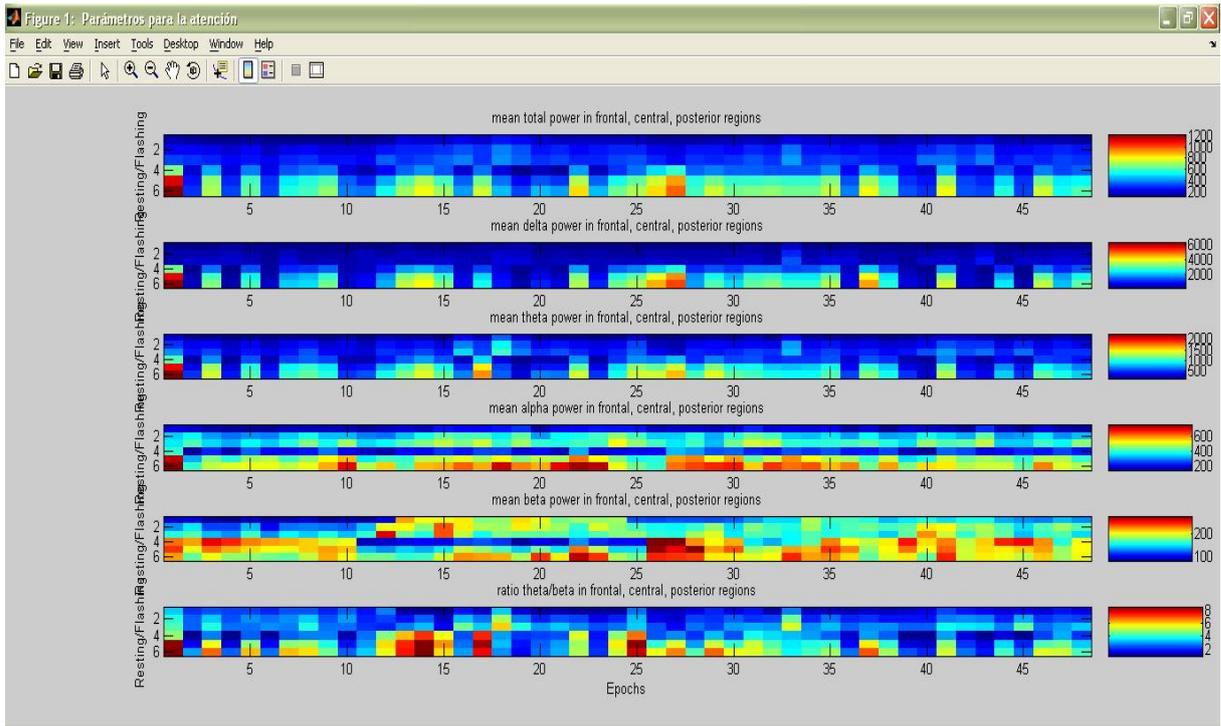


Fig. 9.5.1: Potencias medias del EEG en total y en cada ritmo o banda, ratios theta / beta.

Para ver estos resultados más detalladamente, se representarán estos mismos datos usando gráficas separadas para 'Flashing' y 'Resting', y en términos relativos:

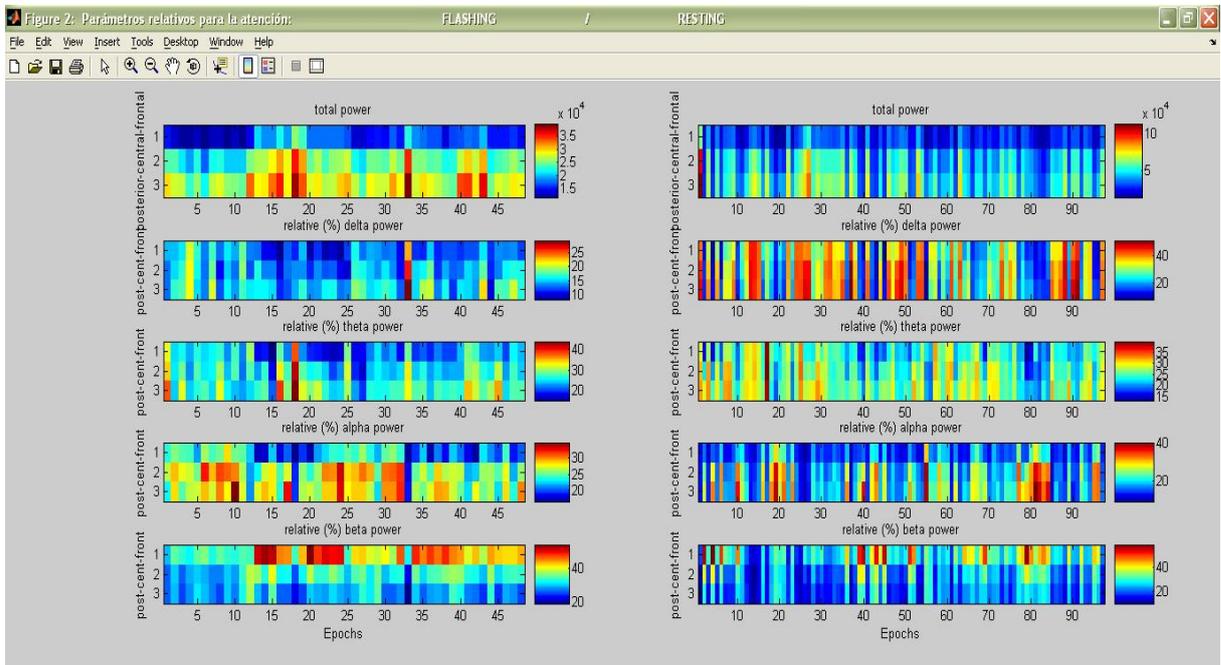


Fig. 9.5.2: Potencia media del EEG en total y relativas en cada ritmo o banda.

Finalmente, y como resultado más importante, se nos mostrará un mapa topográfico cerebral donde podremos observar cómo se distribuye la potencia del EEG por bandas y según el estado ('Flashing' o 'Resting'). En un gráfico adicional, se dibujará el ratio theta / beta, de forma similar. Además, en la ventana de comandos, se mostrará por pantalla la media de cada magnitud, como veremos a continuación.

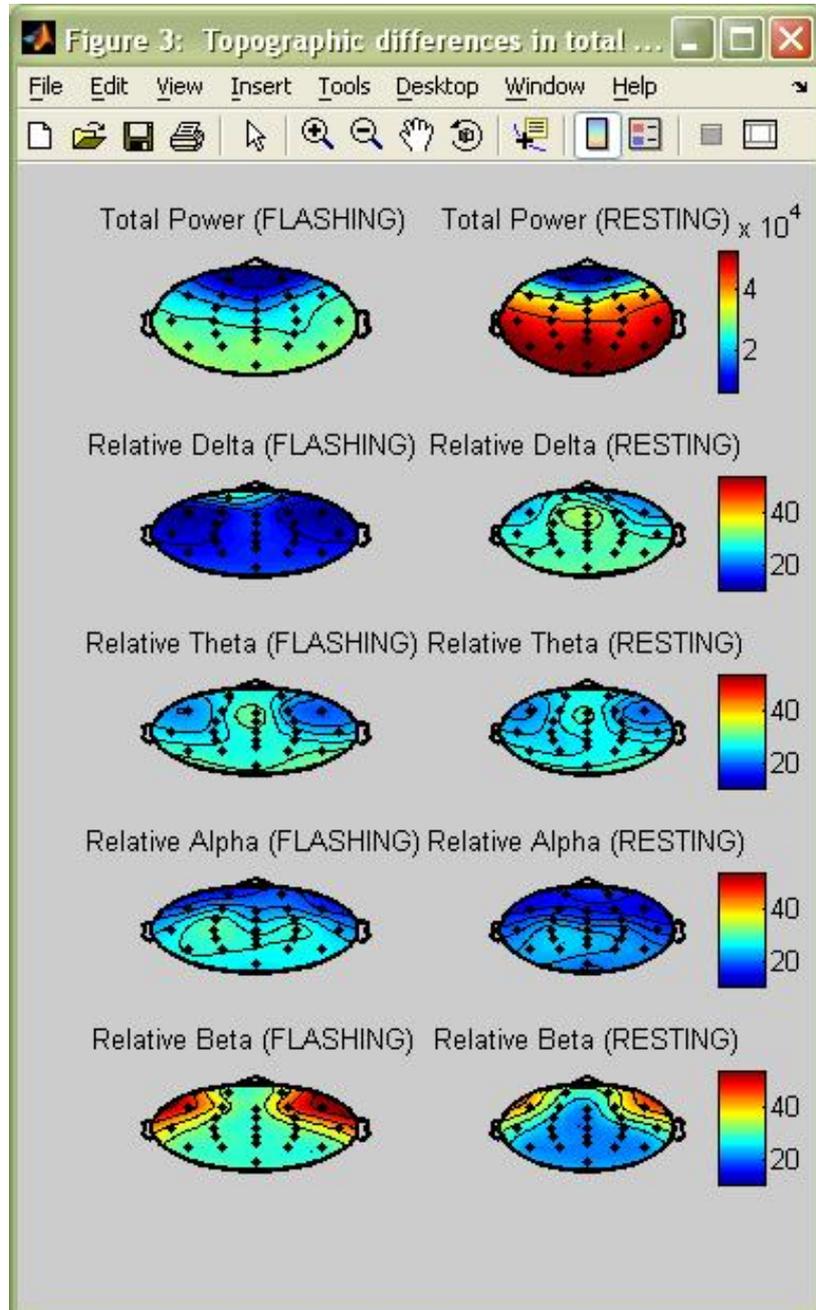


Fig. 9.5.3: Mapeo topográfico de la distribución de potencia del EEG.

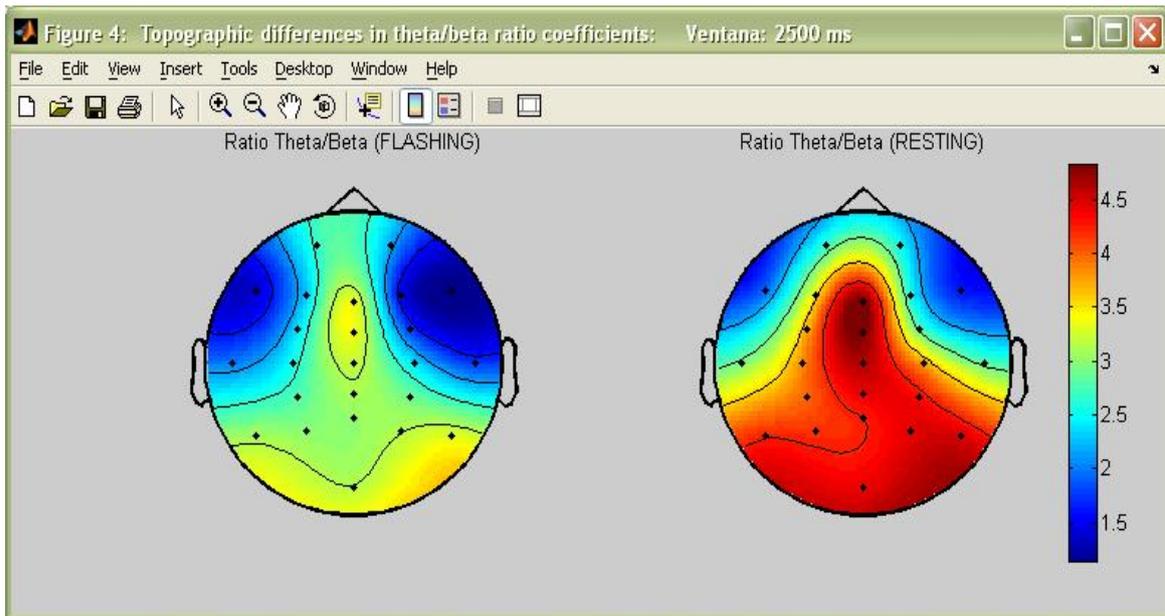


Fig. 9.5.4: Mapeo topográfico de la distribución del ratio theta / beta.

	Flashing	Resting
Total power frontal:	1.639	2.8337
Total power central:	2.508	4.5311
Total power posterior:	2.883	5.1537
Delta frontal:	0.2144	0.85707
Delta central:	0.3644	1.4809
Delta posterior:	0.4651	1.6926
Theta frontal:	0.3827	0.68495
Theta central:	0.6629	1.1759
Theta posterior:	0.8238	1.3934
Alpha frontal:	0.3538	0.42817
Alpha central:	0.6666	0.88246
Alpha posterior:	0.7735	1.0659
Beta frontal:	0.6884	0.86351
Beta central:	0.8142	0.99175
Beta posterior:	0.821	1.0017
Theta/beta frontal:	1.801	2.9311
Theta/beta central:	2.525	4.1293
Theta/beta posterior:	3.104	4.6858

Fig. 9.5.5: Resultados en media de todos los parámetros del estudio.

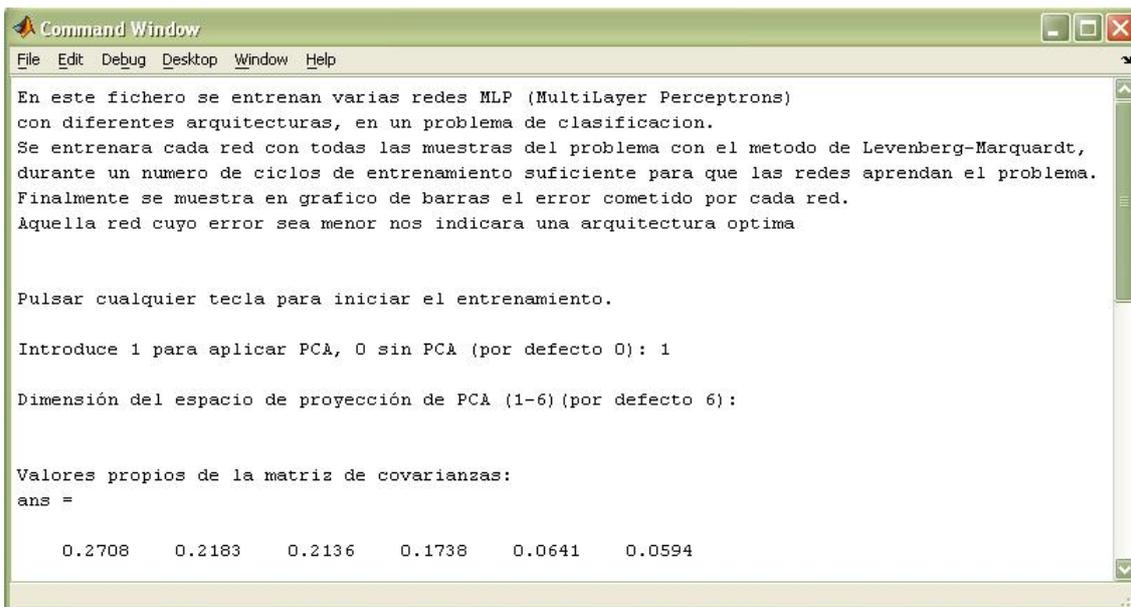
9.6.- Clasificación

9.6.1.- Usando sólo datos de Biosensores

Después de haber descrito con detalle la interfaz, y haber comprobado cómo se ponen en funcionamiento a través de ella todas las etapas del procesado, sólo nos queda averiguar de qué modo se elaborará la separación entre los estados de relajación y estrés, que es nuestro objetivo general en este proyecto (en este capítulo sólo mostraremos los pasos a seguir dentro de la interfaz, para una explicación más detallada de la red neuronal con la que vamos a trabajar, acudiremos al Anexo I).

Si nos referimos a la clasificación usando exclusivamente la información de los biosensores, recordemos que habíamos formado una matriz con los resultados del procesamiento de la señal generada por éstos. Esta matriz consta de tantas filas como variables vayamos a tener en cuenta (de cada biosensor obteníamos varias variables, por ejemplo, del BVP obteníamos el 'BVP-HR', el 'BVP-HRV' y la dilatación vascular o 'PINCH'). En cuanto a las columnas, habrá tantas como muestras consideremos, y el número de muestras vendrá dado por la longitud del experimento en particular (si tenemos más muestras de una variable que de otra, se desecharán para que el número de columnas coincida). Se colocarán a la izquierda las muestras correspondientes a la actividad o 'Flashing', y a la derecha las que pertenezcan a intervalos de descanso o 'Resting', y se pasará al clasificador una variable adicional que contenga esta información (nº de muestras de 'Flashing', nº de muestras de 'Resting'). De esta manera, ya tendremos todos los datos colocados en la forma estándar para que la red neuronal los acepte como entrada, cosa que ocurrirá cuando pulsemos el botón 'Clasificador Biosensores'.

Lo primero que aparecerá en la línea de comandos será una breve explicación de lo que vamos a hacer, que será elegir la arquitectura óptima de la red (nosotros consideraremos aquí que ya hemos hecho las pruebas suficientes y conocemos cuál es esa arquitectura, para más información, ver Anexo I). Además, se nos preguntará si queremos llevar a cabo un Análisis de Componentes Principales (PCA).



```

Command Window
File Edit Debug Desktop Window Help
En este fichero se entrenan varias redes MLP (MultiLayer Perceptrons)
con diferentes arquitecturas, en un problema de clasificacion.
Se entrenara cada red con todas las muestras del problema con el metodo de Levenberg-Marquardt,
durante un numero de ciclos de entrenamiento suficiente para que las redes aprendan el problema.
Finalmente se muestra en grafico de barras el error cometido por cada red.
Aquella red cuyo error sea menor nos indicara una arquitectura optima

Pulsar cualquier tecla para iniciar el entrenamiento.

Introduce 1 para aplicar PCA, 0 sin PCA (por defecto 0): 1

Dimensión del espacio de proyección de PCA (1-6) (por defecto 6):

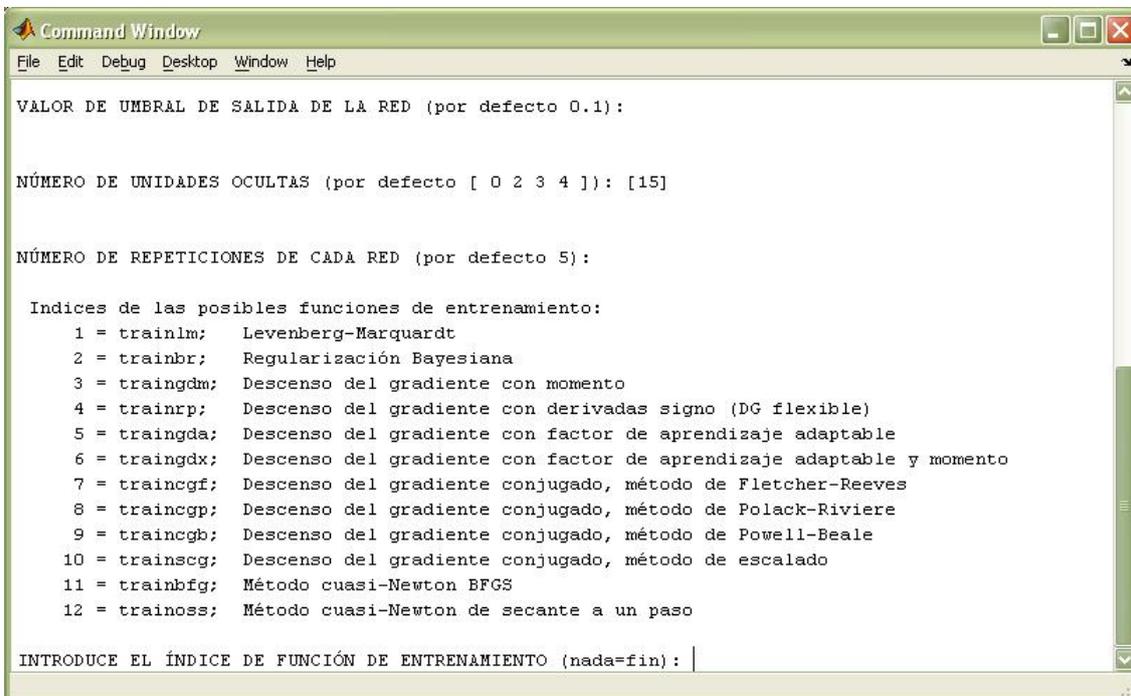
Valores propios de la matriz de covarianzas:
ans =

    0.2708    0.2183    0.2136    0.1738    0.0641    0.0594

```

Fig. 9.6.1.1: Ejecución del clasificador. Ventana de comandos.

Ahora se nos pedirá que escribamos cuántas configuraciones queremos probar, y el número de neuronas ocultas que queremos que tenga cada una. Nosotros supondremos que sólo vamos a probar una, con 15 neuronas en la capa oculta (ver Anexo I). En el tipo de entrenamiento pondremos siempre Levenberg-Marquardt.



```

Command Window
File Edit Debug Desktop Window Help
VALOR DE UMBRAL DE SALIDA DE LA RED (por defecto 0.1):

NÚMERO DE UNIDADES OCULTAS (por defecto [ 0 2 3 4 ]): [15]

NÚMERO DE REPETICIONES DE CADA RED (por defecto 5):

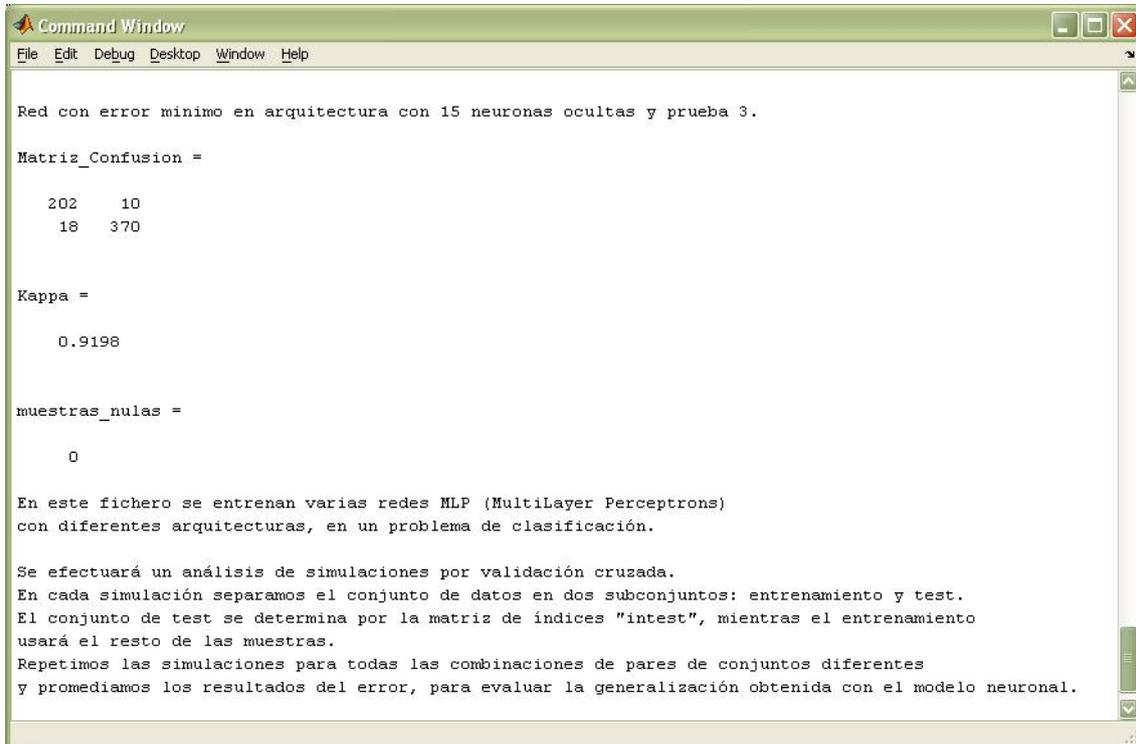
Indices de las posibles funciones de entrenamiento:
 1 = trainlm; Levenberg-Marquardt
 2 = trainbr; Regularización Bayesiana
 3 = traingdm; Descenso del gradiente con momento
 4 = trainrp; Descenso del gradiente con derivadas signo (DG flexible)
 5 = traingda; Descenso del gradiente con factor de aprendizaje adaptable
 6 = traingdx; Descenso del gradiente con factor de aprendizaje adaptable y momento
 7 = traingcf; Descenso del gradiente conjugado, método de Fletcher-Reeves
 8 = traingcp; Descenso del gradiente conjugado, método de Polack-Riviere
 9 = traingcb; Descenso del gradiente conjugado, método de Powell-Beale
10 = trainscg; Descenso del gradiente conjugado, método de escalado
11 = trainbfg; Método cuasi-Newton BFGS
12 = trainoss; Método cuasi-Newton de secante a un paso

INTRODUCE EL ÍNDICE DE FUNCIÓN DE ENTRENAMIENTO (nada=fin): |

```

Fig. 9.6.1.2: Ejecución del clasificador. Parámetros de entrenamiento.

Cuando introduzcamos un '1' en ese último campo, ya tendremos a la red entrenándose. Al terminar, nos dirá qué arquitectura ha obtenido los mejores resultados, como sólo hemos puesto una, lógicamente será esa. Acto seguido, se nos informará del siguiente paso, evaluar la red mediante validación cruzada (ver Anexo I).



```

Command Window
File Edit Debug Desktop Window Help

Red con error minimo en arquitectura con 15 neuronas ocultas y prueba 3.

Matriz_Confucion =

    202    10
     18   370

Kappa =

    0.9198

muestras_nulas =

     0

En este fichero se entrenan varias redes MLP (MultiLayer Perceptrons)
con diferentes arquitecturas, en un problema de clasificación.

Se efectuará un análisis de simulaciones por validación cruzada.
En cada simulación separamos el conjunto de datos en dos subconjuntos:
entrenamiento y test.
El conjunto de test se determina por la matriz de índices "intest",
mientras el entrenamiento usará el resto de las muestras.
Repetimos las simulaciones para todas las combinaciones de pares de
conjuntos diferentes y promediamos los resultados del error, para
evaluar la generalización obtenida con el modelo neuronal.

```

Fig. 9.6.1.3: Ejecución del clasificador. Resultados del entrenamiento.



Fig. 9.6.1.4: Ejecución del clasificador. Resultados del entrenamiento (gráfica).

Nótese que lo que vemos en la anterior imagen es el resultado del entrenamiento, no de la evaluación, es decir, este gráfico sólo nos revela cómo de bien se entrena la red o se aprende las muestras, no cómo predice. Eso precisamente es lo que vamos a averiguar con la validación cruzada.

En adelante, obviaremos algunos detalles como el análisis PCA y algunos gráficos de información estadística, para una mayor claridad (para más información, ver Anexo I).

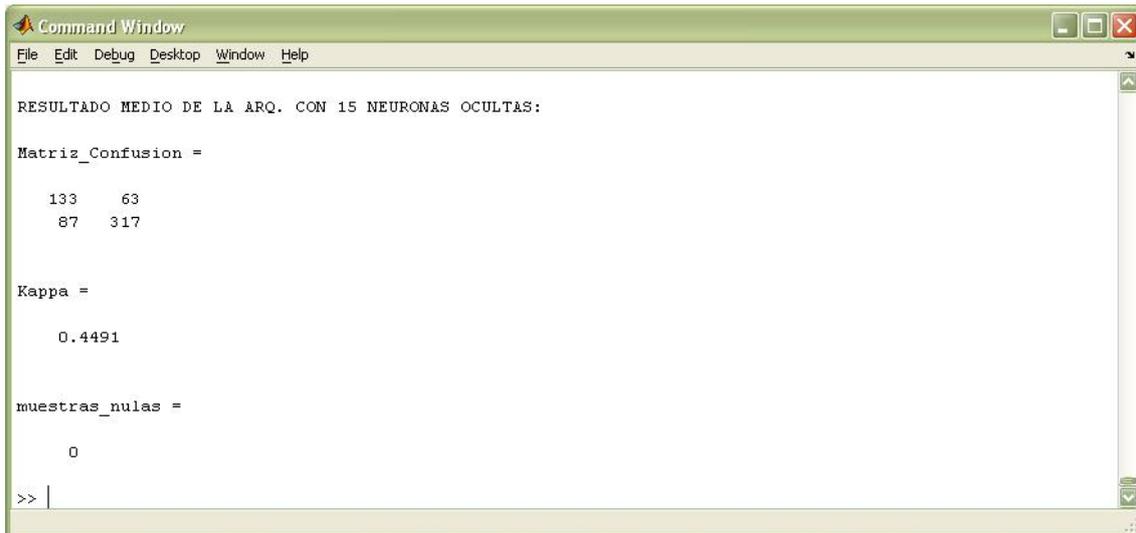


Fig. 9.6.1.5: Ejecución del clasificador. Resultados de la evaluación.

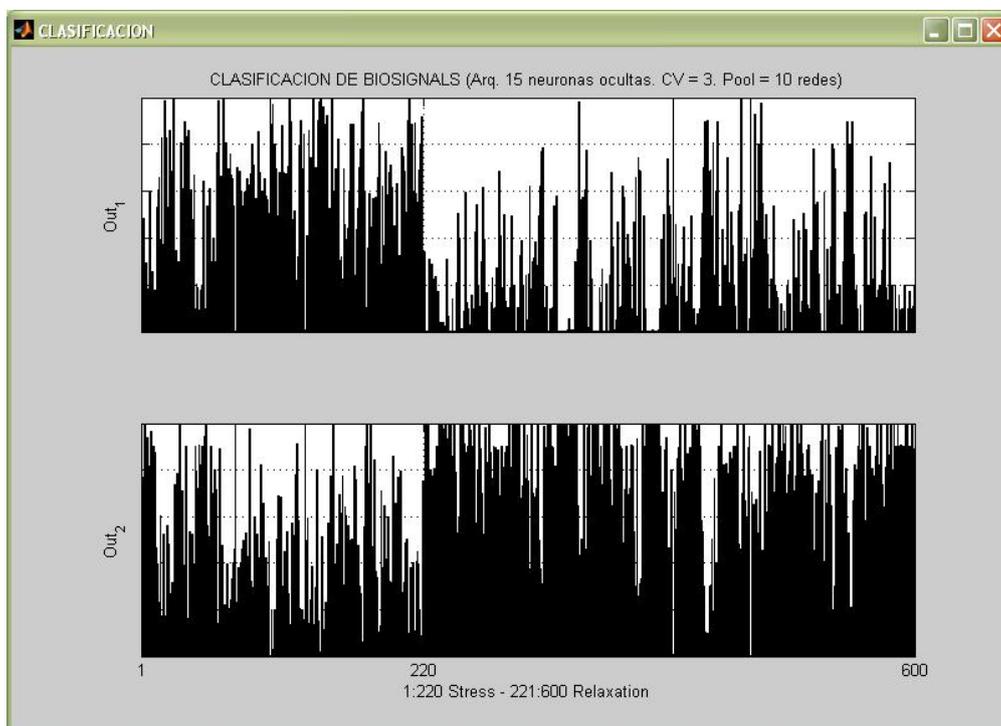


Fig. 9.6.1.6: Ejecución del clasificador. Resultados de la evaluación (gráfica).

Podemos observar que el resultado de la evaluación siempre será peor que el que nos muestra el entrenamiento, lógicamente, pero esta diferencia se reducirá considerablemente dependiendo de varios factores, entre otros: la calidad de las señales con las que trabajemos, la bondad del procesamiento al que sometamos a dichas señales, y la correcta elección de la arquitectura de la red neuronal basada en perceptrón multicapa (MLP) que utilizamos como clasificador de estados (para más información, ver Anexo I).

9.6.2.- Usando sólo EEG

En cuanto al clasificador del EEG, la matriz que se entrega al clasificador estará formada por las potencias de las bandas de frecuencia que hayamos seleccionado, promediadas según los canales elegidos. De esta manera habrá tantas filas como variables hayamos querido tener en cuenta, con un máximo de 6 (potencias de los ritmos delta, theta, alfa, beta, gamma, además del ratio theta / beta). Cada muestra de estas variables será el resultado de promediar las potencias de esos ritmos en los canales más representativos (con mayor varianza entre los dos estados), y el número de columnas vendrá dado por el número de ventanas (o la longitud total del experimento, partido por el tamaño de ventana), ya que los datos provienen de las FFT que hemos aplicado a cada una de las ventanas en las que hemos dividido el experimento.

Igual que en el caso de las bioseñales, las muestras estarán ordenadas según pertenezcan a intervalos de actividad o descanso, y el número de muestras de cada intervalo será parte de la información que se pasará como parámetro cuando llamemos a la función que implementa la red neuronal.

En este caso, no mostraremos detalladamente los mensajes en la línea de comandos, ya que serán similares a los vistos con anterioridad. Simplemente mostraremos los gráficos y los resultados más importantes. Supondremos que tenemos la matriz ya construida a partir del procesado del EEG, en el que recordemos que, a modo de ejemplo, seleccionamos los 24 canales y las 6 variables, con un tamaño de ventana de 300 muestras.

Aplicaremos análisis PCA y evaluaremos una red con 10 neuronas en la capa oculta (para más información, ver Anexo I), del mismo modo en que lo hicimos en el subcapítulo anterior.

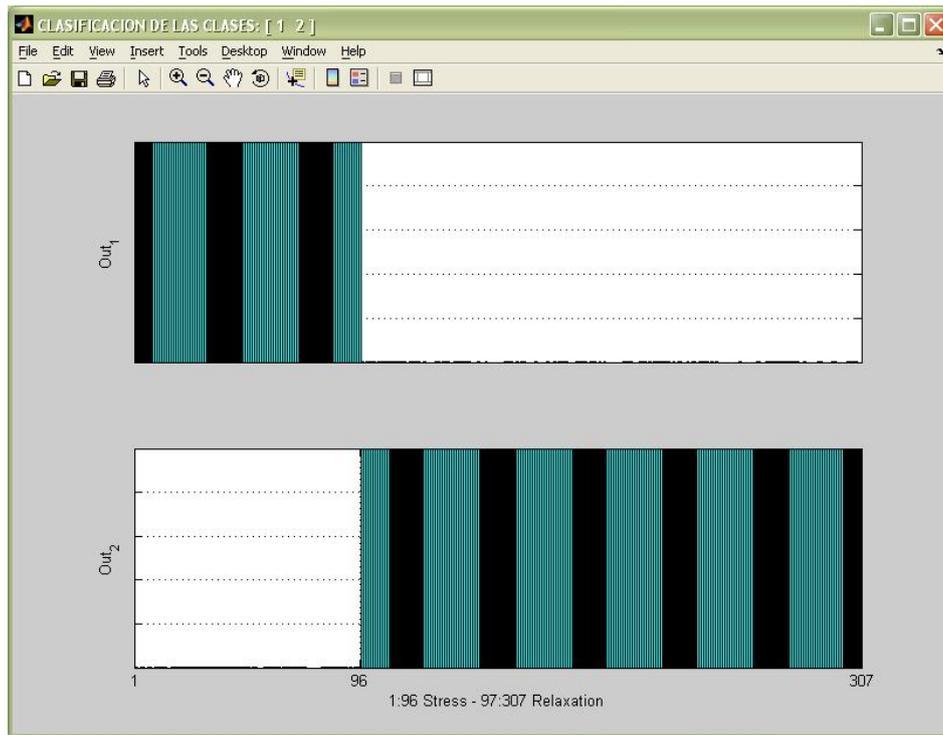


Fig. 9.6.2.1: Ejecución del clasificador EEG. Resultados del entrenamiento.

Observemos que con estos datos, la red neuronal elegida es suficiente para que el entrenamiento se realice a la perfección. Esto es debido a que el análisis PCA ha sido de más ayuda en este caso (detalles en Anexo I). Evaluamos la clasificación:

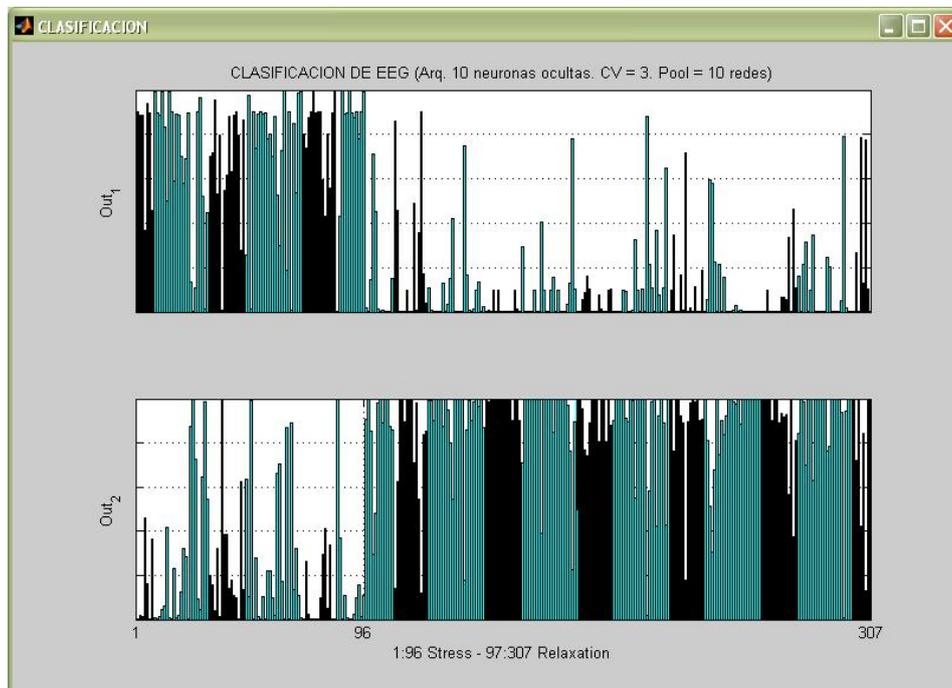
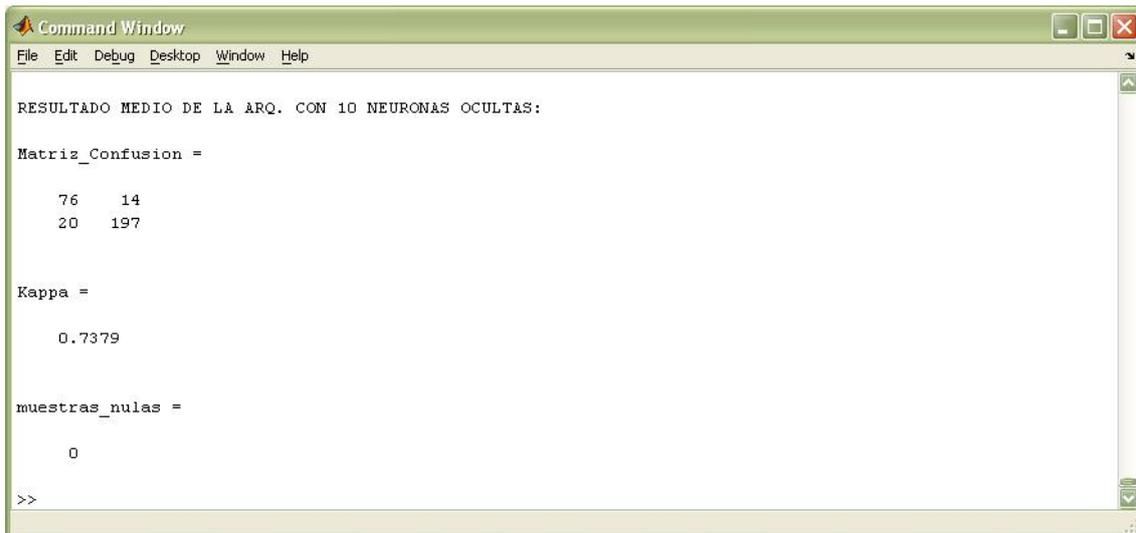


Fig. 9.6.2.2: Ejecución del clasificador EEG. Resultados de la evaluación (gráfica).



```
Command Window
File Edit Debug Desktop Window Help

RESULTADO MEDIO DE LA ARQ. CON 10 NEURONAS OCULTAS:

Matriz_Confusión =

    76    14
    20   197

Kappa =

    0.7379

muestras_nulas =

    0

>>
```

Fig. 9.6.2.3: Ejecución del clasificador EEG. Resultados de la evaluación.

Podemos constatar que la diferencia entre los resultados del entrenamiento y la evaluación se ha reducido en este caso, veamos ahora cuáles serán dichos resultados si efectuamos una clasificación conjunta.

9.6.3.- Usando EEG y Biosensores

Lo primero que se debe hacer para formar una nueva matriz de entrada al clasificador a partir de las dos matrices anteriores es recortar el número de columnas de aquella que más muestras contenga, de manera que eliminando algunas de ellas tengamos el mismo número de muestras, tanto de 'Flashing' como de 'Resting'.

Seguidamente se unirán ambas matrices colocando las filas de una a continuación de las de la otra, a modo de nuevas variables. Si pulsamos el botón 'Clasificador Total', se realizará todo este proceso y se ejecutará el clasificador, que será similar a los anteriores.

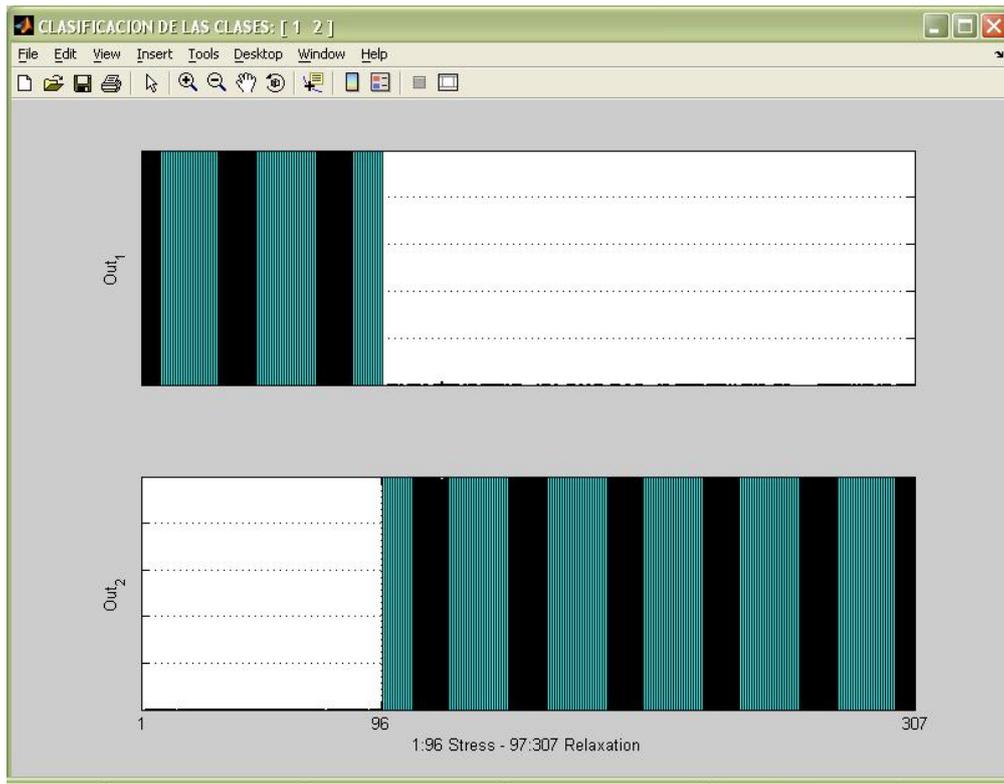


Fig. 9.6.3.1: Ejecución del clasificador total. Resultados del entrenamiento.

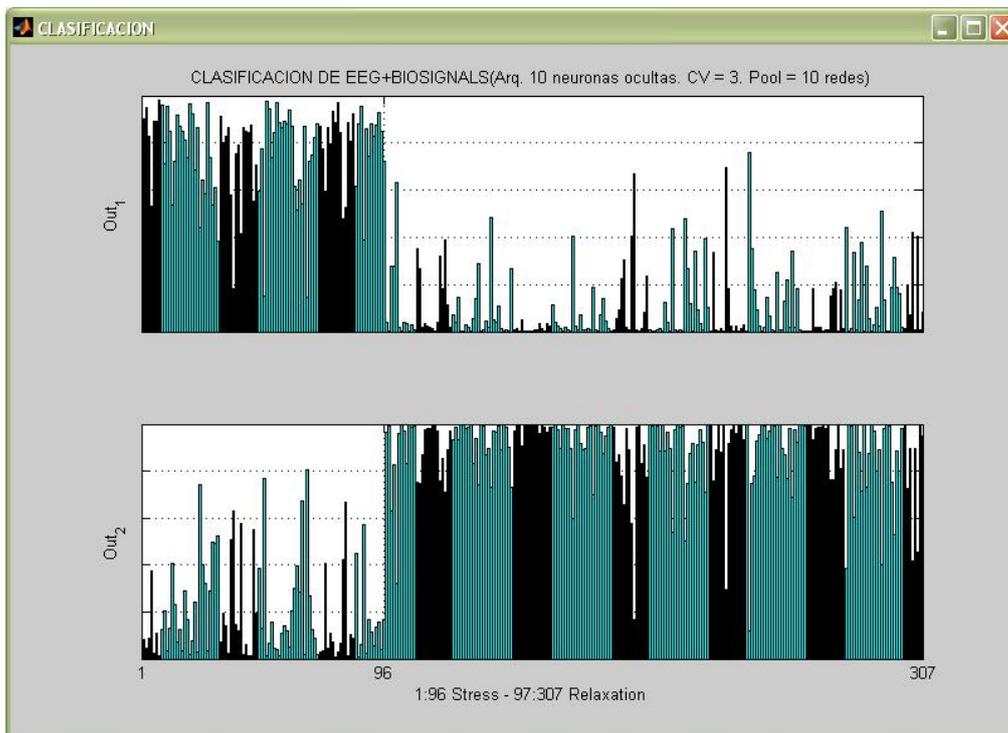
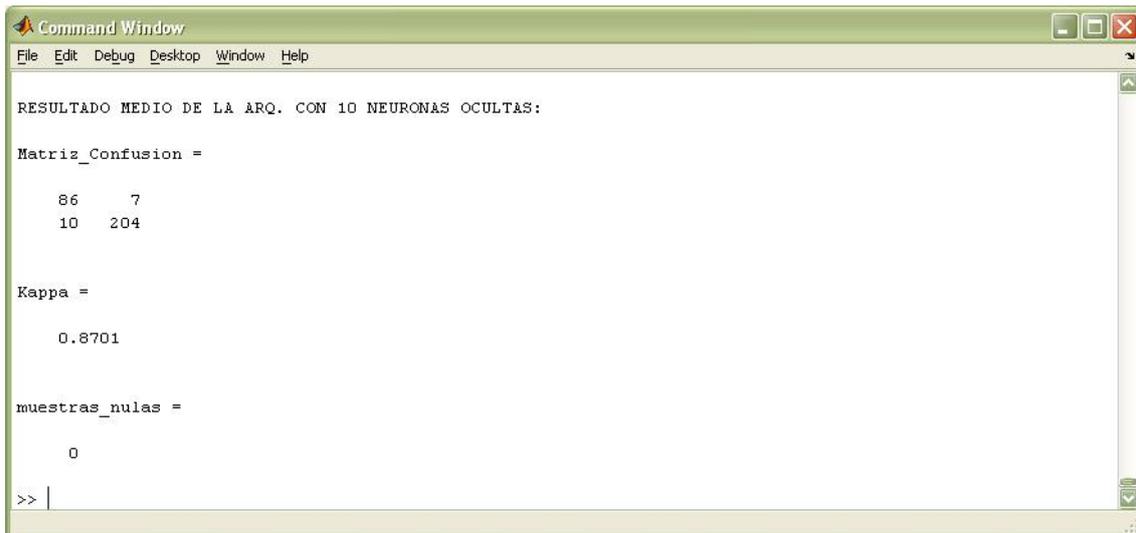


Fig. 9.6.3.2: Ejecución del clasificador total. Resultados de la evaluación (gráfica).



```
Command Window
File Edit Debug Desktop Window Help

RESULTADO MEDIO DE LA ARQ. CON 10 NEURONAS OCULTAS:

Matriz_Confusión =

    86     7
    10   204

Kappa =

    0.8701

muestras_nulas =

    0

>> |
```

Fig. 9.6.3.3: Ejecución del clasificador total. Resultados de la evaluación.

Se aprecia con facilidad la ventaja que supone usar conjuntamente la información que aportan ambas fuentes, aunque cabe comentar que el tiempo de cálculo aumenta de forma considerable, por diversos motivos (como el aumento del número de variables o del espacio de proyección de PCA, para más información, ver Anexo I). Este hecho será de gran importancia en el futuro, si se desea acometer la implementación en tiempo real del software como mejora del presente proyecto.