

UNIVERSIDAD AUTÓNOMA DE MADRID
ESCUELA POLITÉCNICA SUPERIOR



PROYECTO FIN DE CARRERA

**DISEÑO DE INTERFACES CEREBRO-MÁQUINA
CONTROLADOS MEDIANTE REGISTROS DE EEG**

Álvaro Morán García

Enero 2015

DISEÑO DE INTERFACES CEREBRO-MÁQUINA CONTROLADOS MEDIANTE REGISTROS DE EEG

AUTOR: Álvaro Morán García
TUTOR: Pablo Varona Martínez

Grupo de Neurocomputación biológica (GNB)
Departamento de Ingeniería Informática
Escuela Politécnica Superior
Universidad Autónoma de Madrid

Enero – 2015

PROYECTO FIN DE CARRERA

Título: Diseño de interfaces cerebro-máquina controlados mediante registros de EEG

Autor: D. Álvaro Morán García

Tutor: D. Pablo Varona Martínez

Resumen: El avance en la tecnología de la electroencefalografía ha conseguido que el análisis de señales cerebrales haya dejado de ser una actividad reservada al ámbito médico. Nuevas interfaces permiten el uso de señales cerebrales para el control de dispositivos, de forma que personas con discapacidades motoras puedan utilizar distintos tipos de dispositivos para controlar su entorno. En este proyecto se propondrá la creación de una interfaz cerebro-máquina de fácil usabilidad y bajo coste, haciendo uso de unas señales cerebrales específicas y evocadas a través del sentido de la vista.

Palabras clave: Interfaz cerebro-máquina (BCI – Brain-Computer Interface), electroencefalografía (EEG), potenciales visuales evocados de estado continuo (SSVEP – Steady State Visual Evoked Potentials).

Agradecimientos

A todos los compañeros de carrera, pues conseguir llegar hasta aquí ha sido en muchas ocasiones una labor de trabajo en equipo. Y más en especial a Marta, Víctor y Jenniffer por haber convertido todas las horas de prácticas en algo mucho más llevadero, divertido y, en ocasiones, ¡hasta productivo! Y, por supuesto, a mi tutor y a todos los miembros del laboratorio, especialmente a Jacobo pues su contribución en el proyecto ha sido determinante, además de aquellos que se prestaron a realizar la fase de experimentación del proyecto como conejillos de indias.

A Garri que, junto a Bruno, me descubrió todo un nuevo grupo de personas y amigos que han sido de gran ayuda durante estos últimos años... aunque me siguen sin convencer sus reglas para jugar a la Fuga de Colditz, ahí hay gato encerrado...

Cristina, Charly, Karen... cuya compañía y ánimos han sido, en estos últimos tiempos, determinantes y necesarios para afrontar retos muy importantes en lo personal y profesional. Y a aquellos que me han estado apoyando desde siempre fuera de la universidad, especialmente mi familia que han tenido mucha paciencia y buenas palabras a lo largo de los últimos años, además de los amigos de siempre, de León o los que gracias a las maravillas de las telecomunicaciones he ido conociendo en Madrid y otras partes del mundo. Gracias a todos y espero seguir teniendo la suerte de contar con vosotros en el futuro.

*Álvaro Morán García
Enero de 2015*

ÍNDICE DE CONTENIDOS

| | |
|--|-----------|
| 1. Introducción | 9 |
| 1.1. Aspectos generales y motivaciones | 9 |
| 1.2. Objetivos | 11 |
| 1.3. Organización de la memoria | 12 |
| 2. Evaluación del estado del arte | 13 |
| 2.1. Interfaces de comunicación con máquinas | 13 |
| 2.1.1. Interfaces hombre-máquina | 13 |
| 2.1.2. Las interfaces cerebro-máquina | 16 |
| 2.2. Fenómenos neurológicos | 18 |
| 2.2.1. Origen fisiológico de las señales | 18 |
| 2.2.2. Señales endógenas o exógenas | 19 |
| 2.2.3. Adquisición de señales..... | 20 |
| 2.2.4. Señales comúnmente utilizadas | 23 |
| 2.2.4. Uso conjunto de diversas alternativas..... | 26 |
| 2.3. Problemática de las interfaces cerebro-máquina | 27 |
| 2.4. Aplicaciones | 28 |
| 2.5. Perspectivas de futuro | 30 |
| 3. Diseño de la interfaz | 32 |
| 3.1. Consideraciones generales | 32 |
| 3.2. Captura de datos..... | 34 |
| 3.3. Interfaz gráfica..... | 36 |
| 3.4. Análisis de señal..... | 37 |
| 3.5. Selección de frecuencias | 39 |
| 4. Desarrollo..... | 40 |
| 4.1. Captura de datos mediante el emotiv epoc..... | 40 |
| 4.2. Uso del sdk de emotiv epoc..... | 44 |
| 4.3. Desarrollo de la interfaz gráfica | 47 |
| 4.4. Implementación de la interfaz basada en Qt..... | 50 |

| | | |
|-----------|---|------------|
| 4.5. | Configuración del análisis de señal | 54 |
| 4.6. | Implementación online del análisis de señal | 56 |
| 4.7. | Selección de frecuencia resultado | 60 |
| 4.8. | Algoritmo de selección de frecuencias | 61 |
| 4.9. | Selección de frecuencias prefijadas | 65 |
| 5. | Experimentación | 66 |
| 5.1. | Estructura de las pruebas | 66 |
| 5.2. | Previsión de resultados de las pruebas | 69 |
| 5.3. | Resultados del experimento | 69 |
| 5.3.1. | Fiabilidad de la interfaz | 70 |
| 5.3.2. | Separación entre estímulos | 74 |
| 5.3.3. | Morfología de los estímulos | 75 |
| 5.3.4. | Reproducibilidad de los resultados | 77 |
| 5.4. | Conclusiones y mejoras | 78 |
| 6. | Módulos de interés añadidos | 80 |
| 6.1. | Control de robot | 80 |
| 6.2. | Gestión de usuarios y ficheros | 81 |
| 6.3. | Webcam en la interfaz de estimulación | 83 |
| | Referencias | 85 |
| | Glosario | 88 |
| | Anexo I – Programa principal | 91 |
| | Anexo II – Instalación y librerías | 94 |
| | Anexo III – Guía de usuario | 99 |
| | Presupuesto | 101 |
| | Pliego de condiciones | 102 |

ÍNDICE DE FIGURAS

| | <u>Página</u> |
|---|---------------|
| Figura 2-1: Esquema de interfaces hombre-máquina. | 14 |
| Figura 2-2: (A) Uso de tarjetas perforadas como dispositivo de entrada ... | 15 |
| Figura 2-3: Esquema de interfaces cerebro-máquina. | 18 |
| Figura 2-4: Respuesta hemodinámica a la activación cerebral... | 19 |
| Figura 2-5: (A) Ejemplo de matriz de electrodos utilizada para ECoG... | 21 |
| Figura 2-6: El casco Mindwave de NeuroSky es un ejemplo de instrumento ... | 22 |
| Figura 2-7: Relación visual entre los distintos métodos de obtención de ... | 23 |
| Figura 2-8: Aproximación visual de la relación entre la tasa de transferencia... | 25 |
| Figura 2-9: Aplicación de BCIs en función de la tasa de información... | 29 |
| Figura 3-1: Esquema general de la estructura general que tendrá la interfaz... | 34 |
| Figura 3-2: Colocación de electrodos según el estándar internacional 10-20... | 35 |
| Figura 3-3: (A) Representación de único estímulo plano y (B) estímulo... | 37 |
| Figura 3-4: Ejemplo de monitorización de SSVEP utilizando la resta de... | 38 |
| Figura 3-5: Ejemplo de espectro ideal de señal monitorizada (dB)... | 39 |
| Figura 4-1: (A) Emotiv Epoc y (B) disposición de los 14 electrodos... | 41 |
| Figura 4-2: Ejemplo de resultados para un usuario concreto a 7Hz... | 43 |
| Figura 4-3: Esquema de uso de la API para conectarse con el EmoEngine... | 44 |
| Figura 4-4: Secuencia de llamadas a las funciones proporcionadas por la API... | 45 |
| Figura 4-5: Esquema principal de ejecución de la interfaz estimuladora... | 51 |
| Figura 4-6: Ejemplo de cómo quedaría el segundo estímulo posicionado... | 54 |
| Figura 4-7: Esquema del análisis de señal implementado. | 55 |
| Figura 4-8: Flujo de la rutina encargada de realizar el análisis de señal. | 56 |
| Figura 5-1: Representación de la última fase del algoritmo ACL... | 67 |
| Figura 6-1: Esquema de comunicación con el programa de control... | 81 |
| Figura 6-2: Ventana de estimulación con los cuatro estímulos y... | 84 |
| Figura A1-1: Esquema general de funcionamiento [...] main.cpp. | 91 |
| Figura A1-2: Esquema general de funcionamiento [...] interfaz.cpp. | 92 |
| Figura A1-3: Esquema general de la dependencia entre funciones... | 93 |
| Figura A2-1: Esquema general de organización de los archivos y directorios. | 98 |
| Figura A3-1: Menú principal del programa. | 99 |
| Figura A3-2: Opciones de configuración de la interfaz gráfica... | 100 |

ÍNDICE DE TABLAS

| | <u>Página</u> |
|--|---------------|
| Tabla 2-1: Ventajas y desventajas de interfaces cerebro-máquina... | 20 |
| Tabla 2-2: Comparación entre distintos sistemas híbridos... | 26 |
| Tabla 4-1: Resultados obtenidos al intentar mostrar un evento... | 49 |
| Tabla 4-2: Análisis más detallado de desempeño entre Qt y SDL... | 50 |
| Tabla 5-1: Resultados del entrenamiento, siendo SR el success rate... | 71 |
| Tabla 5-2: Resultados de la experimentación. Las filas finales... | 73 |
| Tabla 5-3: Resultados para distinta distancia entre iconos. | 75 |
| Tabla 5-4: Resultados mostrando imágenes planas contra imágenes... | 76 |
| Tabla 5-5: Resultados mostrando las 4 frecuencias elegidas en... | 77 |

1. INTRODUCCIÓN

1.1. ASPECTOS GENERALES Y MOTIVACIONES

Recientemente se ha producido un gran impulso en lo referente a las interfaces hombre-máquina que nos permiten relacionarnos con distintos tipos de dispositivos. Hace no mucho tiempo este tipo de interacciones estaban limitadas básicamente al uso de teclado y ratón o algún otro tipo de dispositivo de interacción basado en presionar diferentes botones o palancas, en definitiva, haciendo uso de las extremidades. Sin embargo, en los últimos años ha habido un gran empuje en otro tipo de interfaces que pretenden mejorar la accesibilidad por parte de los usuarios de este tipo de sistemas: control de dispositivos mediante voz, interacción mediante vibraciones, detección de mirada o *eye tracking*, medición de parámetros biométricos, visores de realidad aumentada, pantallas táctiles... Todo ello ha sido posible, en gran medida, a la gran evolución tecnológica y a la mayor capacidad de procesamiento de datos que permiten los nuevos equipos y que facilitan que estas interfaces puedan trabajar en tiempo real.

Si bien es cierto que muchos de los dispositivos de interacción más antiguos siguen siendo insustituibles en gran parte de los entornos, existen casos particulares en donde pueden ser de gran utilidad sistemas basados en otro tipo de tecnologías más novedosas. Más en concreto, la necesidad de nuevas formas de interacción se hace evidente para aquellas personas que, por determinadas razones, no pueden utilizar sistemas basados en el uso de las extremidades ya sea por imposibilidad al estar realizando una determinada tarea o bien por algún tipo de discapacidad. En este contexto cobran importancia las interfaces cerebro-máquina como alternativa.

Una interfaz cerebro-máquina (*Brain Computer Interface*, o BCI) es un sistema de comunicaciones que permite el control de dispositivos mediante el análisis y codificación de señales cerebrales, ya sea la actividad eléctrica, magnética, vascular... Esto permite a los usuarios de un BCI ser capaces de interactuar con su entorno sin la necesidad de involucrar ningún tipo de actividad articular (Nicolas-Alonso and Gomez-Gil 2012), simplemente codificando estas señales en una orden que pueda ser ejecutada por una máquina o un dispositivo.

Este tipo de sistemas tienen una gran utilidad para personas que sufren algún tipo de discapacidad severa como aquellas que sufren Esclerosis Lateral Amiotrófica (ALS - *Amyotrophic lateral sclerosis*), lesiones en el cerebro o la espina dorsal y otro tipo de enfermedades que limitan o impiden por completo el uso de las articulaciones (Ortner, Guger, and Prueckl 2010). Mediante el uso de un BCI este tipo de usuarios encontrarán

una nueva forma de interactuar con su entorno permitiendo nuevas formas de comunicación y/o locomoción, siendo por tanto una pieza vital para incrementar su independencia y mejorar su calidad de vida. Existen, no obstante, otros usos para estos sistemas orientados al ocio y entretenimiento, si bien es cierto que este campo no ha sido prioritario y viable hasta muy recientemente (Krepki et al. 2007).

Hay múltiples alternativas para la creación de este tipo de interfaces. El uso de métodos no invasivos proporciona una vía de realizar las mediciones en la actividad cerebral limitando las molestias sobre el usuario. La electroencefalografía, magnetoencefalografía, imagen por resonancia magnética funcional... son algunos de los ejemplos que se pueden citar dentro de esta tendencia. Ofrecen una mayor comodidad con limitaciones en la calidad de la señal. Sin embargo, existen también métodos invasivos como la electrocorticografía o las grabaciones intracorticales que facilitan la obtención y calidad de las señales a cambio de suponer un alto riesgo para el usuario, existiendo por tanto un compromiso entre ambas tendencias. Este tipo de interfaces solo están justificadas en personas con patologías muy graves. En el presente proyecto se hará uso de la electroencefalografía para crear la interfaz cerebro-máquina de bajo coste y adaptada a cada usuario.

Además de diversas formas de realizar las mediciones, existen múltiples fenómenos neurológicos de los que el usuario se puede servir para hacer uso de la interfaz. El uso de estímulos sensoriales, por ejemplo, se traduce en la generación de potenciales característicos en el cerebro como respuesta a dichos estímulos que pueden ser cuantificados. Esta no es la única alternativa; por ejemplo, el uso de la visualización motora (Höhne et al. 2014) mediante la cual un usuario puede simular mentalmente que realiza una determinada acción motora tiene ya largo recorrido en la rehabilitación neurológica e investigación y su uso para el control de interfaces es viable con el debido entrenamiento. El fenómeno neurológico del que se hará uso para la realización del BCI en este proyecto son los llamados *Steady State Visual Evoked Potentials* (Ortner, Guger, and Prueckl 2010). Como su nombre indica, se trata de señales generadas en el cerebro y medibles desde el cuero cabelludo como efecto a un estímulo de tipo visual. Si dicho estímulo es periódico entonces el SSVEP generado también deberá serlo y su frecuencia estará directamente relacionada con la del estímulo observado, facilitando así su medición. No obstante, existen también interfaces que se fundamentan en otros estímulos como los auditivos (Gao et al. 2014) o interfaces que utilizan más de un sistema de análisis o de estimulación con el objetivo de mejorar las prestaciones que ofrece el sistema (Amiri, Fazel-rezai, and Asadpour 2013).

Como se ha comentado anteriormente, la mayor capacidad de procesamiento que han experimentado los equipos informáticos ha permitido que este tipo de fenómenos, neurológicos en este caso, sean analizables en tiempo real. Ésta no es, sin embargo, la única ventaja que el desarrollo tecnológico ha traído al campo de las interfaces cerebro-máquina; la creación de estímulos sensoriales de los cuales se derivan algunos de los fenómenos neurológicos más interesantes se ha visto beneficiada de estas

innovaciones. Algunos ejemplos de ello es el uso de estímulos de tipo gráfico, auditivo, táctil o el uso de los mismos en configuración de lazo cerrado para facilitar el uso de las interfaces. Para la adecuada estimulación de los SSVEP en el presente proyecto será vital la creación de una interfaz gráfica que proporcione los estímulos visuales a la frecuencia deseada con la mayor precisión posible.

Uno de los principales retos a la hora de crear una interfaz cerebro-máquina es obtener una buena fidelidad del sistema, limitando las falsas decisiones. Una decisión errónea por parte del sistema podría ser fatal e incluso podría comprometer la integridad del usuario de la interfaz, especialmente si se trata de un usuario discapacitado que depende del BCI para interactuar con el entorno, desplazarse, comunicarse... Se trata por tanto de sistemas en los que es de vital importancia una correcta interpretación de las señales, siendo este aspecto una de las principales motivaciones a la hora de abordar este proyecto.

Estas interfaces son, por otro lado, todavía desconocidas para el gran público y su usabilidad es una de las asignaturas pendientes. Se pueden obtener buenos resultados utilizando cascos específicos pero en esos casos el tiempo y complejidad de montaje de los mismos puede resultar una experiencia desagradable para el usuario. En este proyecto se pondrá especial foco en hacer la experiencia de usuario más agradable mediante la utilización de un casco genérico de fácil montaje, una adaptación específica de la interfaz a las particularidades de cada sujeto y el desarrollo de un entorno que permita un fácil control de la misma, evitando en la medida de lo posible que ello signifique un compromiso en la eficiencia y fidelidad de uso. Como resultado se obtendrá una interfaz de bajo coste, cómoda y que compensará la baja calidad de la señal mediante la adaptación al sujeto y el uso de la realimentación o ciclo cerrado para ello.

De forma adicional, el cuidado en el análisis de la señal, desarrollo de software y poder trabajar en un entorno multidisciplinar son motivaciones extra para la realización de este tipo de proyectos.

1.2. OBJETIVOS

El principal objetivo del presente proyecto es la creación de una interfaz cerebro-máquina de bajo coste y alta adaptabilidad al sujeto que permita el control de un dispositivo mediante el análisis de la actividad cerebral del usuario. Siendo el desarrollo de un BCI completo un proceso ambicioso, este proyecto se centrará en el software de la interfaz y análisis de señal, así como en el sistema evocador de estímulos específicamente adaptado a cada usuario con el objetivo de obtener una buena fiabilidad. La accesibilidad será otro de los pilares básicos de este proyecto, utilizando métodos de adquisición de señal no invasivos y que sean lo más cómodos posibles para

el usuario. Para llevar a cabo con éxito todo lo anterior será preciso prestar especial atención a lo siguiente:

- **Diseñar de forma adecuada la interfaz evocadora de estímulos visuales.** Dicha interfaz ha de contar con todas las opciones que el dispositivo pueda realizar, siendo necesario el uso de distintas frecuencias de parpadeo para cada opción.
- **Optimizar la elección de las frecuencias de parpadeo.** Dicha elección no es trivial debido a que las señales generadas en el cerebro no son estacionarias y pueden variar entre distintos individuos, algo que puede comprometer el desempeño del sistema. Por otro lado, la interfaz visual se mostrará al usuario en una pantalla que tendrá una frecuencia de refresco determinada y que puede entrar en conflicto con las frecuencias de los estímulos específicos para cada usuario. Esto condicionará la elección de frecuencias de parpadeo, cuya selección se llevará a cabo mediante un protocolo de ciclo cerrado.
- **Énfasis en el análisis de señal** de cara a obtener los mejores resultados posibles, tratándose de una interfaz no invasiva con un casco de toma de datos de propósito general y bajo coste del que no se espera obtener una gran relación de señal respecto al ruido y contando, además, con la no estacionariedad de las señales.

1.3. ORGANIZACIÓN DE LA MEMORIA

La memoria no se centrará exclusivamente en el apartado técnico del proyecto si no que abarcará también un análisis del estado del arte en el campo de las interfaces cerebro-máquina. Basándonos en dicha evaluación se argumentarán las decisiones de diseño que se han tomado. Posteriormente se explicará el trabajo técnico y de desarrollo realizado.

El estado del arte abarcará una serie de hitos históricos que han desembocado en que la tecnología propuesta para realizar el proyecto haya sido viable. Posteriormente se hablará acerca de las tecnologías actuales disponibles, así como de las señales cerebrales que permiten que la construcción de un BCI sea posible. Una vez hecho esto, se abordarán las problemáticas que habrá que afrontar a la hora de crear la interfaz y se discutirán las alternativas de diseño de cara a realizar el proyecto, evaluando pros y contras y justificando las elecciones tomadas. A continuación se abordarán las cuestiones técnicas y se explicará cómo se ha llevado a cabo el diseño y construcción del proyecto, completando la información más detalladamente en los anexos adjuntados al final de la documentación. Posteriormente, se procederá a describir el procedimiento

para realizar la experimentación sobre usuarios, así como los resultados obtenidos y sus correspondientes conclusiones y discusión.

2. EVALUACIÓN DEL ESTADO DEL ARTE

2.1. INTERFACES DE COMUNICACIÓN CON MÁQUINAS

2.1.1. Interfaces hombre-máquina

Las interfaces hombre-máquina tienen como tarea la gestión de la interacción de una persona con un dispositivo. Esta tecnología ha sufrido muchos cambios a lo largo de la historia, siempre en persecución de una mayor accesibilidad y usabilidad por parte de los usuarios (Lebedev 2014).

El desarrollo de este tipo de interfaces no es algo trivial; el crear una interfaz de comunicación entre un usuario y un computador supone el trabajo en un entorno multidisciplinar que implica conocimientos del funcionamiento del propio computador y, también, acerca de la persona de cara a que la interfaz sea usable por ella:

- **Máquinas:** Conocimiento acerca del sistema de control, lenguajes de programación, creación de interfaces gráficas, análisis de señal, entornos de desarrollo en tiempo real...
- **Personas:** Lingüística, psicología, funcionamiento de procesos cognitivos, usabilidad, ergonomía...

Se puede definir, en definitiva, una interfaz hombre-máquina como el nexo comunicativo entre el usuario y el dispositivo. Existe, por tanto, un intercambio de información entre ambos que en todas estas interfaces tiene un esquema similar:

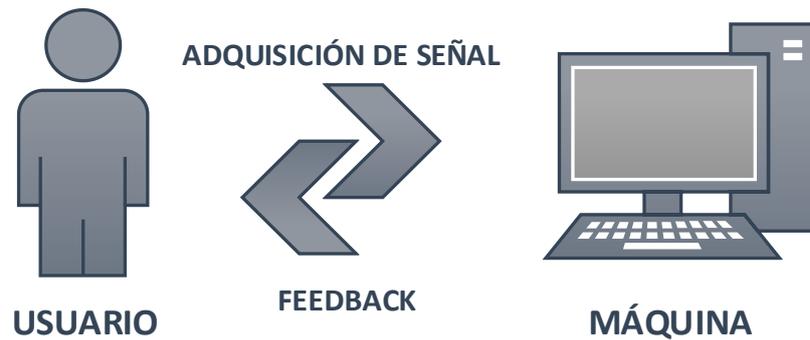


Figura 2-1: Esquema de interfaces hombre-máquina.

En muchas ocasiones, y como se desprende del esquema anterior, estas interfaces de comunicación han de ser dotadas de cierta capacidad que permita tanto la adquisición y procesado de la señal, de ser necesaria, como de la habilidad de proporcionar realimentación (*feedback*) al usuario. Estos dos factores han sido los limitantes a la hora de desarrollar sistemas de comunicación entre los usuarios y los dispositivos:

- Eficacia de las interfaces de salida y *feedback*, típicamente de las interfaces gráficas.
- Capacidad de procesamiento de las máquinas.

Antes de la existencia de las interfaces gráficas complejas el uso de las tarjetas perforadas estaba extendido como método para introducir, obtener y almacenar datos en los centros de computación de la época. Su utilización como interfaz de entrada y de salida de los ordenadores ha sido ampliamente superada por sistemas que ofrecen un mejor control del usuario respecto de la máquina y, lo que es más importante, en tiempo real.

A mediados de los años 60 se comenzaron a desarrollar nuevos tipos de interfaces que sustituirían los viejos modelos basados en las tarjetas perforadas. Dispositivos como el IBM 2260 o su evolución, el IBM 3270, serían algunos de los responsables de la transición (Wikipedia 2015). Estos computadores ya contaban con interfaces gráficas consistentes en primitivas consolas de texto. Con la mencionada evolución en la interfaz gráfica se hizo necesaria la introducción de una nueva interfaz de entrada de datos e instrucciones que facilitase la introducción de comandos en forma de texto que pudieran ser legibles en la pantalla y que, desde ésta a su vez, se pudieran evaluar los resultados de así ser requerido. Dicha interfaz de entrada sería el teclado.

Mientras el teclado se afianzaba como dispositivo de entrada en los ordenadores, el ratón ya estaba siendo desarrollado. No obstante, y debido al todavía frecuente uso de interfaces gráficas basadas en texto, tardaría décadas en popularizarse. Fue en el año 1981 con el Xerox Star 8010 cuando un ratón apareció por primera vez incluido junto a

la computadora, algo necesario pues este ordenador disponía de un hardware bastante potente y de una avanzada interfaz gráfica que dependía de este nuevo dispositivo de entrada. Tanto el IBBM 2260 como el Xerox Star 8010 son muestras de que las interfaces hombre-máquina tienen una fuerte dependencia de la capacidad de procesamiento de los dispositivos, siendo esta capacidad la que permite el análisis de nuevas señales de control cada vez más complejas y de mecanismos de *feedback* en tiempo real.

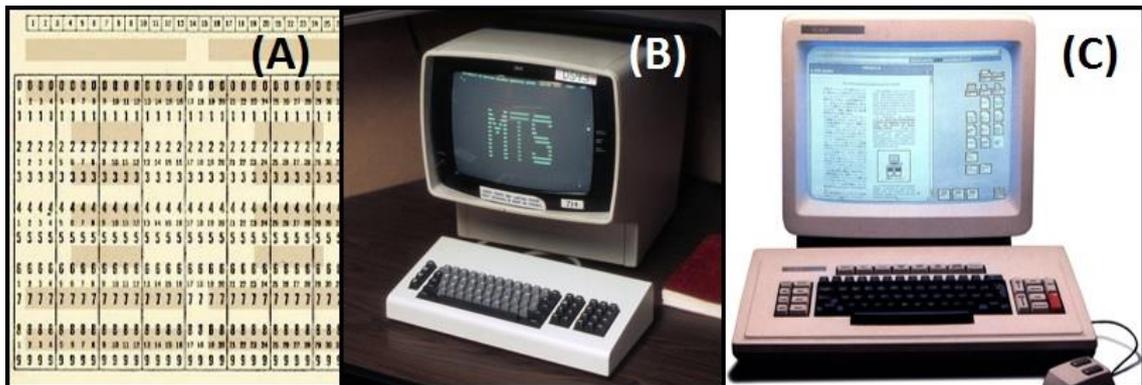


Figura 2-2: (A) Uso de tarjetas perforadas como dispositivo de entrada y salida de información, (B) evolución de interfaces hacia las consolas de texto representada por el IBM 3270 y (C) primeras interfaces gráficas representadas por el Xerox Star 8010.

A partir de este momento y dadas las mejoras en capacidad de procesamiento se comenzarán a popularizar nuevos tipos de interfaces de comunicación entre el usuario y la máquina: control mediante voz, interacción táctil, interacción mediante parámetros biométricos como la monitorización de movimiento ocular, interfaces cerebro-máquina....

Centrándonos en éstas últimas, se puede decir que el desarrollo de este tipo de interfaces comenzó con los primeros sistemas de adquisición electroencefalográficos, que datan de finales del s. XIX y principios del s. XX. Sería en el año 1924 cuando Hans Berger, continuando con el trabajo realizado por Richard Caton décadas antes, sería capaz de captar actividad cerebral por primera vez mediante la electroencefalografía. No obstante, dadas las limitaciones técnicas, no sería hasta la década de los 70 cuando se comenzó a investigar en el uso de la electroencefalografía como nueva interfaz de comunicación con dispositivos, consiguiendo desarrollar las primeras prótesis neuronales en 1990 y cuyo uso fundamental era ayudar a la recuperación de la audición, vista o movilidad dañadas de sus usuarios. También se comenzarían a incorporar otros tipos de registro además de la electroencefalografía para realizar la toma de datos sobre los usuarios, como las imágenes por resonancia magnética funcional, la magnetoencefalografía... El desarrollo tecnológico ha permitido que, recientemente, se puedan popularizar interfaces cerebro-máquina de uso incluso doméstico y con métodos no invasivos para una multitud de usos, ya no solo de índole médica sino

también en el contexto de aplicaciones de ocio o simplemente como un nuevo sistema de comunicación con los computadores.

2.1.2. Las interfaces cerebro-máquina

El fin de una interfaz cerebro-máquina es comunicar usuario y máquina sin requerir ningún tipo de actividad articular, siendo esta la principal característica diferenciadora con el resto de interfaces hombre-máquina que se han mencionado anteriormente. La filosofía de funcionamiento es aparentemente sencilla: la interfaz será la encargada de enviar una señal a partir de la actividad cerebral del usuario hacia un dispositivo externo que será el encargado de ejecutar una determinada acción. Sin embargo, esta definición implica que no solo es importante la detección de los estímulos cerebrales (Elshout 2009), algo que de por sí puede ser complejo, sino que además se necesita traducir dichos estímulos cerebrales en comandos que sean comprensibles por la propia máquina. Por tanto, el uso de las interfaces cerebro-máquina está íntimamente ligado al desarrollo de las distintas tecnologías que permitan la lectura de las distintas señales cerebrales, siendo el estudio del cerebro un campo de investigación relativamente nuevo y, en ocasiones, controvertido.

La electroencefalografía fue la disciplina que permitió, en primera instancia, que se pudiera considerar un uso no exclusivamente médico al estudio del cerebro, siendo por tanto la principal impulsora de los BCIs. Como se ha comentado anteriormente, el mayor impulso en la investigación y aplicación de la electroencefalografía se produjo a partir de los años 70. En el año 1976 se produjo la primera evidencia real de que se podría utilizar este tipo de señales para establecer sistemas de comunicación. Fue Jacques J. Vidal, la persona que acuñaría el término *Brain Computer Interface*, el que proporcionó evidencia de que el registro de determinados potenciales podía ser utilizado como canal de comunicación lo suficientemente eficiente como para controlar un cursor a través de una pantalla bidimensional. Ésta sería la primera prueba oficial de que se pueden utilizar señales cerebrales para el control de dispositivos externos al organismo (Arafat 2013). Paralelamente se irían sucediendo una serie de investigaciones alentadoras para el campo de la encefalografía; se produciría el nacimiento del primer oído biónico (1972), la primera interfaz basada en BCI para ayudar a los ciegos (1978), evolución de implantes que producían señales de alta calidad... A partir del año 2000 se producirían avances significativos en el campo de las interfaces cerebro-máquina: en el año 2002 una serie de monos con electrodos implantados fueron capaces de desplazar eficazmente un cursor a través de una pantalla. A partir de este momento el desarrollo de las interfaces cerebro-máquina ha sido imparable, ya no solo en lo referente a logros enfocados a aplicaciones médicas sino también para otro tipo de utilidades multimedia y de ocio:

- Posibilidad de twittear mediante un BCI (Universidad de Wisconsin, 2009).
- Posibilidad de controlar robots mediante la mente (Honda, 2009).
- Primer juego basado en un BCI (Departamento de Neurociencia de la Universidad de Brown, 2003).
- Reconstrucción de imágenes desde el cerebro y su almacenamiento en computadora (*ATR Computational Neuroscience Laboratories* en Kioto, 2008).

En los últimos cuatro años la investigación en interfaces cerebro-máquina se ha extendido por muchos laboratorios de todo el mundo lo que confirma el interés a global en este tipo de tecnologías. Con cada nueva aplicación, la viabilidad y usabilidad de este tipo de interfaces se va asentando paulatinamente y permite esperar en el futuro grandes (Arafat 2013):

- **Uso extendido de BCIs a personas con discapacidades.** Los implantes de cóclea (oídos biónicos) son ya algo común. Desde hace unos años se viene experimentando también con ojos biónicos, consiguiendo paulatinamente mejores resultados y el control de brazos y piernas usando el pensamiento no es algo raro hoy en día. Para el año 2030 hay previsiones de que este tipo de implantes estén generalizados.
- **Comprensión del funcionamiento del cerebro.** El estudio de esta parte del cuerpo humano está muy poco desarrollado todavía respecto a otras disciplinas. Sin embargo, se espera que, gracias al desarrollo tecnológico que se está experimentando, sea posible alcanzar un alto nivel de interpretación en las próximas décadas. Esto ayudará a desarrollar interfaces cada vez más eficaces.

Si bien el uso de los BCIs para paliar los efectos de discapacidades en los usuarios que las padezcan es uno de los fines de la tecnología, también puede ser muy útil en el estudio de las propias discapacidades. La constante monitorización de señales cerebrales de distinta índole, el uso de estímulos, el entrenamiento de interfaces y la evolución del usuario en el manejo del sistema pueden favorecer el estudio de enfermedades y discapacidades, además de proporcionar un análisis minucioso de las mismas y su evolución.

Existen muchos tipos diferentes de interfaces cerebro-máquina pero todas ellas tienen un esquema de funcionamiento bastante similar y que se deriva del esquema típico utilizado en las interfaces hombre-máquina descrito en la Figura 2-1:

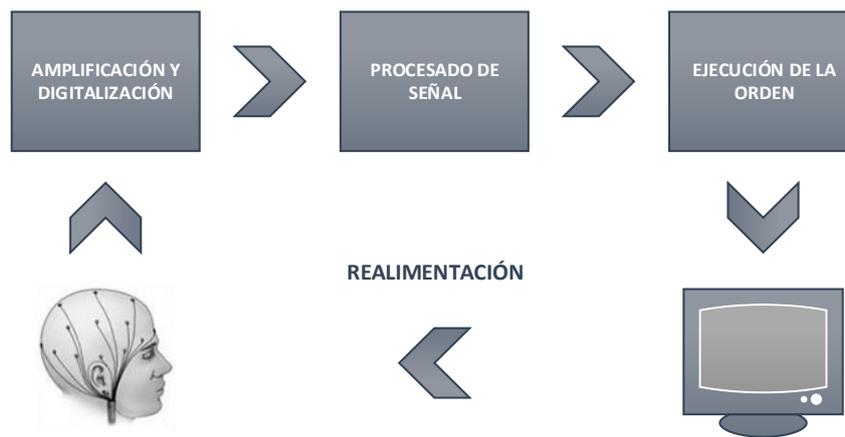


Figura 2-3: Esquema de interfaces cerebro-máquina.

Dado que las señales son de naturaleza analógica, será necesaria una fase de amplificación y digitalización para poder procesarlas e interpretarlas. Una vez que la señal está digitalizada, se analizarán sus características más representativas que dependerán del fenómeno psicológico que estemos evaluando y del objetivo de la interfaz. Una vez obtenidas dichas características, éstas permitirán clasificar la señal dentro de alguno de los conjuntos que constituyen las distintas órdenes o comandos que puede comprender y ejecutar la interfaz. Esto constituirá el procesado de señal y la interpretación y posterior clasificación de resultados, que ha de realizarse en tiempo real. De igual forma que sucedía para interfaces hombre-máquina genéricas, la realimentación será de vital importancia para que el usuario tenga control sobre el proceso que se está realizando. En el caso de las interfaces cerebro-máquina esta realimentación puede producirse de distintas formas; si se pretende el control de una silla de ruedas, por ejemplo, el propio movimiento de la silla constituirá el *feedback* que el usuario necesita. Si por el contrario se trata de controlar otro tipo de dispositivos, como un brazo robótico, es posible que sea necesaria realimentación visual por pantalla si el mecanismo queda fuera de la vista del usuario. Existe un gran abanico de posibilidades y el empleo de unas u otras dependerá del uso final de la interfaz.

2.2. FENÓMENOS NEUROLÓGICOS

2.2.1. Origen fisiológico de las señales

A la hora de analizar la actividad cerebral de los usuarios hay que tener en consideración que existen dos tipos principales de señales que pueden ser monitorizadas:

Señales electrofisiológicas:

Esta actividad es generada por transmisores electroquímicos que intercambian información entre neuronas. Las neuronas generan corrientes iónicas que se propagan a través de uniones neuronales (Nicolas-Alonso and Gomez-Gil 2012). Algunas de estas señales se generan de forma continua en el cerebro sin relación aparente con estímulos exteriores a la persona. Sin embargo, hay otras señales que surgen a partir de alguna estimulación exterior, fenómeno que es comúnmente utilizado para la realización de interfaces hombre-máquina. Este tipo de actividad puede ser medida mediante electroencefalografía, electrocorticografía, magnetoencefalografía...

Señales hemodinámicas:

Cuando se produce activación mental, el metabolismo neurológico ha de ser apoyado por una respuesta vascular local que lleve consigo sangre rica en oxígeno a la zona activa. Las zonas inactivas, por otro lado, no contarán con esta respuesta vascular y no contarán con tal cantidad de sangre rica en oxígeno (Pearlmutter et al. 2008). Estas respuestas hemodinámicas pueden ser cuantificadas mediante resonancias magnéticas y espectroscopia del infrarrojo cercano y permiten identificar zonas activas de zonas inactivas, pudiendo utilizar esta información para el desarrollo de interfaces.

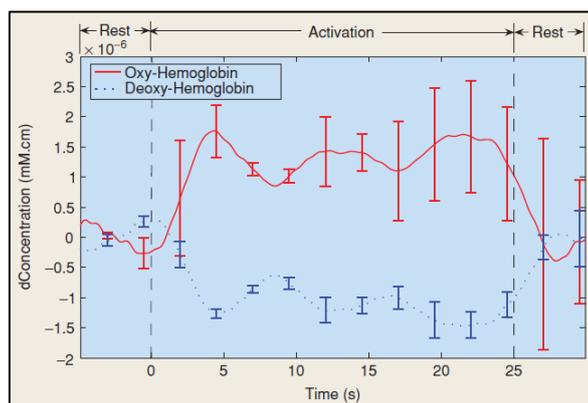


Figura 2-4: Respuesta hemodinámica a la activación cerebral, con incremento en la zona de oxihemoglobina rica en oxígeno (Pearlmutter et al. 2008).

Como se ha visto, en función del origen fisiológico de las señales que se deseen analizar hay distintos métodos y mecanismos para acceder a su análisis, lo cual condiciona el diseño de las interfaces y de los mecanismos de lectura de actividad cerebral.

2.2.2. Señales endógenas o exógenas

Otra característica a considerar sobre los fenómenos neurológicos de interés para realizar interfaces cerebro-máquina es el origen de las señales deseadas y de si éstas son producto de una estimulación externa o bien de una actividad inherente al propio

individuo. Esto deriva en dos aproximaciones muy diferenciadas a la hora de diseñar un BCI:

Señales evocadas o exógenas:

Se consideran como señales exógenas o evocadas a aquellas que se generan mediante una estimulación exterior. Muchas de las señales más comúnmente utilizadas a la hora de realizar interfaces cerebro-máquina son dependientes de estímulos visuales exteriores. No obstante, la estimulación exterior no tiene por qué ser solo visual sino que puede implicar alguna otra actividad sensorial como la auditiva (Gao et al. 2014). Las interfaces exógenas son BCIs que utilizan este tipo de señales.

Señales no evocadas o endógenas:

Las señales endógenas, en contraposición a las exógenas, son aquellas que se generan en el interior del cerebro sin necesidad de estimulación externa. Dependen de la habilidad del usuario para controlar su actividad electrofisiológica (Hornero, Corralejo, and Álvarez 2012). Las interfaces endógenas utilizan señales como las mencionadas anteriormente.

El utilizar interfaces que utilicen señales exógenas o endógenas tiene una serie de ventajas y desventajas asociadas a la naturaleza de las oscilaciones:

| | VENTAJAS | DESVENTAJAS |
|----------------------|--|--|
| INTERFACES EXÓGENAS | Poco entrenamiento, rápida configuración, alta tasa de transferencia y funcionalidad con pocos canales de información. | Necesario prestar atención continuada al estímulo lo que puede producir cansancio. |
| INTERFACES ENDÓGENAS | No dependen de estímulos, lo que lo hace útil para personas con órganos sensoriales afectados. Útil para manejo de cursores. | Largo entrenamiento que no garantiza obtener control sobre el sistema. Baja tasa de transferencia y necesidad de múltiples canales de información. |

Tabla 2-1: Ventajas y desventajas de interfaces cerebro-máquina atendiendo a la naturaleza de la señal (Nicolas-Alonso and Gomez-Gil 2012).

2.2.3. Adquisición de señales

La mayor parte de los BCIs obtienen la información relevante a partir de la electroencefalografía gracias, esencialmente, a su elevada resolución temporal, bajo

coste, alta portabilidad y escaso riesgo para los usuarios. Sin embargo, la eficacia a la hora de obtener resultados depende de múltiples factores; el cuero cabelludo, cráneo, ruido de fondo proveniente de otros dispositivos o señales del propio cerebro... son factores que pueden dificultar la necesaria obtención de las señales deseadas, introduciendo grandes componentes de ruido. Existen dos aproximaciones para hacer frente al problema de la captación de la información deseada:

Métodos invasivos:

Este tipo de métodos se centran en la obtención del mejor tipo de señales posibles limitando el ruido y los obstáculos. Para ello será típicamente necesario el acceso a través del cuero cabelludo a las zonas del cerebro donde se pretende captar la información. Un ejemplo claro de ello sería la electrocorticografía (ECoG - *Electrocorticography*), método consistente en colocar electrodos directamente sobre el cerebro en alguna región expuesta para registrar la actividad eléctrica con el fin de evitar las atenuaciones e interferencias del cráneo y cuero cabelludo. Existen métodos todavía más invasivos, como puede ser el caso de las grabaciones intraparenquimales o intracorticales. En estos casos se pueden llegar a colocar numerosos electrodos incluso dentro de la materia gris del cerebro. La calidad de las señales será por tanto altísima con una gran resolución espacial. Una demostración práctica es The Utah Intracortical Electrode Array (UIEA), una estructura microscópica de silicón que permite la implantación de un gran número de electrodos en una pequeña región del córtex (Maynard, Nordhausen, and Normann 1997). Mediante los métodos invasivos se pueden obtener comunicaciones más rápidas y precisas a la hora de desarrollar un BCI (McFarland and Wolpaw 2011) pero conlleva un riesgo alto para la salud del usuario, dado que requiere cirugía intracraneal. Esto además se traduce en que la experimentación de esta alternativa de toma de datos no ha podido ser probada extensamente en humanos. No obstante, para determinadas actividades es posible que sean necesarias aproximaciones invasivas en función del rendimiento que sea necesario, como para el control de prótesis (Konrad and Shanks 2010). En ocasiones este tipo de interfaces se utilizan en patologías como la epilepsia para encontrar el foco que provoca los ataques, y de forma concurrente se prueban durante el tiempo que están implantados para el control de dispositivos externos.

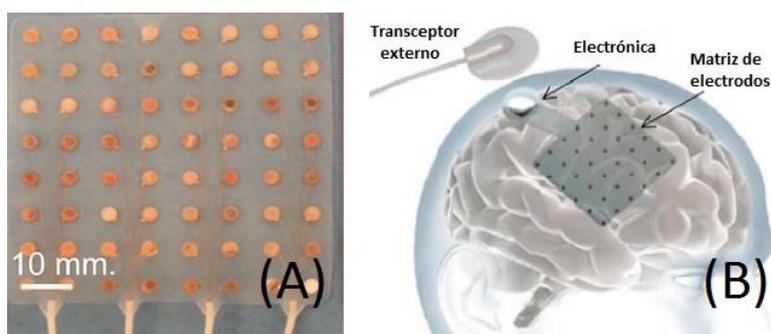


Figura 2-5: (A) Ejemplo de matriz de electrodos utilizada para ECoG y (B) su disposición sobre el cerebro (Schalk and Leuthardt 2011).

Métodos no invasivos:

Este tipo de métodos limitan el impacto de la adquisición de la señal sobre el usuario realizando una toma de señales desde el exterior de la cabeza del usuario. Uno de los métodos no invasivos más utilizados para la implementación de interfaces cerebro-máquina es la electroencefalografía (EEG - *Electroencephalography*), consistente en la colocación de electrodos directamente sobre el cuero cabelludo con ayuda de un gel o líquido conductor, lo que no proporciona una resolución espacial muy elevada ya que las medidas que se tomen corresponderán a la actividad eléctrica de una zona de la superficie del cerebro relativamente grande.



Figura 2-6: El casco Mindwave de NeuroSky es un ejemplo de instrumento no invasivo para tomar datos.

Por otro lado, la resolución temporal es alta, el montaje para realizar mediciones es sencillo y barato y se pueden obtener resultados lo suficientemente buenos para gran parte de las aplicaciones que se puedan desempeñar con un BCI. Este método ha sido también utilizado en medicina, aunque otras técnicas más modernas lo han ido desplazando a un uso, en general, de control de algunas enfermedades. Dispositivos como el casco Mindwave constituyen alternativas baratas no invasivas basadas en la electroencefalografía. Otro método no invasivo y de gran interés es la imagen por resonancia magnética funcional (fMRI - *Functional Magnetic Resonance Imaging*), que permite solventar alguna de las carencias de la electroencefalografía; permite una mayor resolución espacial y acceder a estructuras internas del cerebro en lugar de analizar solo diferencias de tensión superficiales (Sulzer et al. 2013; Ruiz et al. 2014). Su funcionamiento se basa en el análisis de los cambios que se producen en el flujo de la sangre en el cerebro, teniendo siempre presente que el comportamiento de este flujo está ligado a la actividad cerebral. Para ello, se evalúa el diferente comportamiento magnético entre la sangre rica en oxígeno y la sangre pobre en oxígeno y volumen de sangre en ciertas zonas del cerebro. Para poder realizar estos análisis se hace uso de los escáneres basados en las propiedades de la resonancia magnética nuclear. También cabe destacar como alternativa no invasiva la magnetoencefalografía (MEG - *Magnetoencephalography*), consistente en el análisis de la actividad magnética del cerebro a partir del fenómeno de la inducción magnética. Los procesos neurofisiológicos que producen las señales analizables mediante MEG son los mismos que permiten un

análisis mediante EEG (Nicolas-Alonso and Gomez-Gil 2012), a diferencia de las imágenes por resonancia magnética, que tienen su origen en procesos de tipo vascular. Las ventajas de MEG respecto de EEG se deben a que los campos magnéticos son menos afectados por el cuero cabelludo y hueso que los campos eléctricos, produciendo una mejor resolución tanto espacial como temporal. Sin embargo, el uso de MEG es poco viable debido al equipamiento electrónico necesario.

Como se ha visto, existen diferentes métodos de adquisición de señales en función de la capa a la que necesiten acceder para realizar la toma de datos. A mayor profundidad y mejor captura de datos, se obtendrá una mayor fidelidad del sistema, pudiendo incluso acceder a zonas muy concretas del cerebro en la búsqueda de señales muy particulares y limitando la presencia del ruido, en lugar de realizar mediciones en áreas relativamente grandes con presencia de ruido debido a capas de tejidos inferiores, como el cuero cabelludo, y limitando también la presencia de otra infinidad de señales no deseadas provenientes de distintas zonas del cerebro.

La siguiente imagen muestra el grado de invasión de los electrodos en las distintas capas que constituyen el cerebro y su protección exterior, siempre considerando que una mayor profundidad implica una mayor facilidad de análisis de datos pero, por otro lado, una cirugía más invasiva:

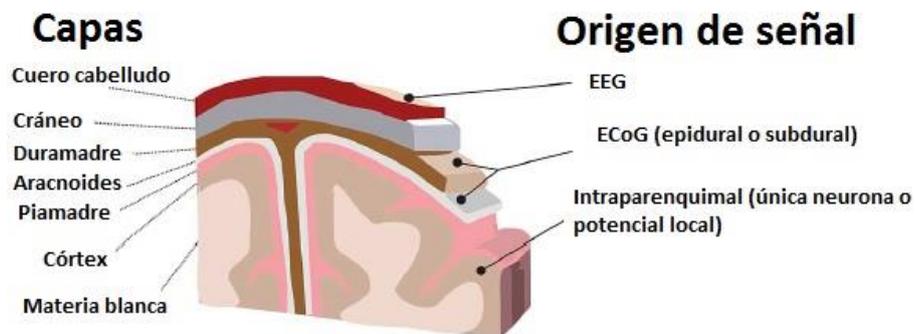


Figura 2-7: Relación visual entre los distintos métodos de obtención de señal que requieren uso de electrodos y capa de toma de datos (Schalk and Leuthardt 2011).

2.2.4. Señales comúnmente utilizadas

Algunos de las señales y fenómenos psicológicos más utilizados para la creación de interfaces son los siguientes (Nicolas-Alonso and Gomez-Gil 2012):

Potenciales Visuales Evocados:

También conocidos como VEPs (*Visual Evoked Potentials*), son potenciales que ocurren en la corteza visual, en la parte occipital del cráneo (Herrmann 2001; Capilla et

al. 2011). Este tipo de variaciones son fáciles de detectar ya que la amplitud de las mismas depende de lo centrado que esté el estímulo dentro del campo visual. Se suelen utilizar estímulos visuales procedentes de puntos de luz LED o pantallas, tratándose por tanto de potenciales de tipo exógeno, y pueden llegar a ofrecer unas tasas de información relativamente elevadas con muy buena fiabilidad. Este tipo de señales son generadas en base a diferentes criterios:

- **Morfología del estímulo visual:** Según este criterio, los potenciales visuales evocados pueden ser debidos a imágenes parpadeantes o patrones de imágenes de diversa naturaleza.
- **Frecuencia del estímulo visual:** En función de la frecuencia los potenciales evocados pueden clasificarse como transitorios (TVEPs – *Transient VEPs*) o como de estado continuo (SSVEPs – *Steady State VEPs*). Si el estímulo visual tiene una frecuencia por debajo de los 6Hz, el potencial evocado se considera TVEP, mientras que para una frecuencia más elevada se consideran como SSVEPs.
- **Área del estímulo:** En función de si el estímulo visual ocupa parcial o totalmente la pantalla y el campo visual del individuo se pueden distinguir estímulos de campo completo, medio campo y campo parcial.

Potenciales corticales lentos:

Los potenciales corticales lentos (SCPs – *Slow Cortical Potentials*) corresponden a cambios lentos de tensión en la actividad encefalografía de origen endógeno que pueden durar uno o varios segundos, teniendo por tanto una frecuencia de variación por debajo de 1Hz (McFarland and Wolpaw 2011). Están asociadas con cambios en los niveles de actividad cortical. Un nivel negativo de SCPs correspondería a un incremento de la actividad neuronal en esa zona y están típicamente relacionados con el movimiento y otras funciones, mientras que un valor positivo coincide con un decrecimiento de la actividad. Este tipo de señales tienen una gran dependencia respecto del estado emocional del usuario, además de otros factores como el sueño, dolor... Se ha demostrado que las personas pueden aprender a controlar estos potenciales (Hornero, Corralejo, and Álvarez 2012) llegando a obtener sistemas fiables en un 70% u 80%. Las interfaces que hacen uso de ellos, no obstante, precisan de un gran entrenamiento y la tasa de información que pueden entregar es baja debido a su frecuencia de variación.

Potenciales P300:

Se trata de picos positivos en el análisis electroencefalográfico debido a la percepción por parte del usuario de eventos infrecuentes ya sean estos de tipo auditivo, visual o somatosensorial, tratándose por tanto de potenciales evocados y señales de tipo

exógenas. Este tipo de potenciales suelen presentarse en torno a 300 milisegundos después de recibir el estímulo, derivando de ahí su nombre (Elshout 2009). Sin embargo, la correcta interpretación de estas señales no es trivial; la atención por parte del usuario, la probabilidad del suceso, el tiempo entre estímulos, la dificultad de la tarea, la habituación del usuario al estímulo... son factores que juegan en contra de una correcta interpretación (Guger et al. 2009). Debido a la latencia de estos potenciales, las tasas de transferencia de datos que se pueden alcanzar oscilan entre los 10 y los 25 bits por minuto.

Ritmos sensoriomotores o imágenes motoras:

El hecho de realizar determinadas tareas mentales, como la realización de cálculos aritméticos, imaginar la rotación de un cubo... o pensar en la ejecución de determinadas tareas motoras, como el movimiento de extremidades, produce cambios en la amplitud de los ritmos sensoriomotores (Hornero, Corralejo, and Álvarez 2012). Estas oscilaciones en la actividad cerebral están localizadas en la banda μ (7-13Hz) y la banda β (13-30Hz). Los ritmos sensoriomotores comprenden, además, dos tipos de variaciones en amplitud (Nicolas-Alonso and Gomez-Gil 2012), las de tipo ERD (*Event-Related Desynchronization*) y las ERS (*Event-Related Synchronization*). Las variaciones de tipo ERD implican una atenuación de la amplitud de los ritmos, mientras que las variaciones de tipo ERS implican un aumento de la amplitud de los mismos. El hecho de que se pueda o no producir movimiento articular al pensar en el movimiento de alguna extremidad no afecta a la amplitud observada en estos ritmos. Esto implica que los ritmos sensoriomotores están relacionados con la imaginación y la intención de realizar un movimiento, independientemente del movimiento en sí mismo. Sin embargo, esto añade una serie de dificultades a la hora de utilizar estas señales dado que muchas personas tienen grandes dificultades con la imaginación e intencionalidad de realizar movimientos, siendo por tanto necesario un buen entrenamiento (Amiri et al. 2013).

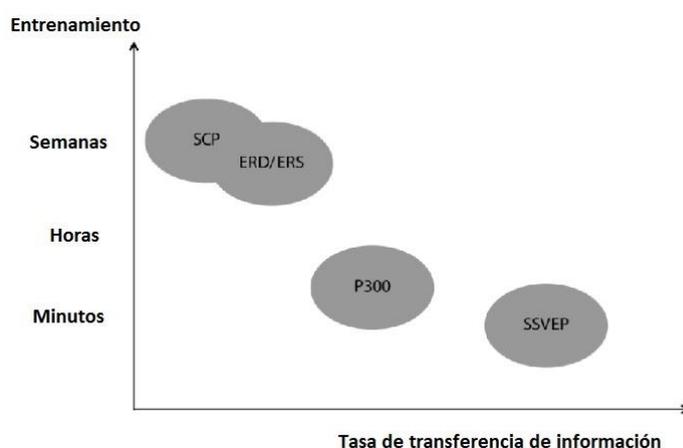


Figura 2-8: Aproximación visual de la relación entre la tasa de transferencia de información y el tiempo necesario para entrenar una interfaz en función del tipo de señal.

2.2.4. Uso conjunto de diversas alternativas

Para intentar mejorar la fiabilidad que puedan tener las interfaces cerebro-máquina, especialmente aquellas que cuenten con un método de adquisición de señales no invasiva que necesite mejorar sus prestaciones, se puede utilizar un uso conjunto de diversas señales o, directamente, combinarlo con distintas técnicas que involucren otras disciplinas, como otros análisis biométricos, *eye-gaze*... En función de cómo se realice la implementación de las distintas técnicas o análisis a realizar se pueden distinguir dos tipos de interfaces:

- **Secuenciales:** Interfaces organizadas de tal forma que inicialmente se hace uso de uno de los análisis y posteriormente se refuerza con otro.
- **Simultánea:** Los distintos análisis y mediciones que se realizan sobre el usuario se realizan en paralelo y a la vez.

El estudio de estos BCIs híbridos es un campo de investigación en auge en la actualidad, indicación de que las esperanzas de mejorar las prestaciones de estas interfaces están puestas en esta dirección (Amiri et al. 2013). En la tabla 2-2 se puede comprobar alguna de las recientes mejoras que se han logrado desarrollar en base al estudio de tecnologías híbridas.

| SISTEMA | IMPLEMENTACIÓN | MEJORAS |
|-------------|----------------|---|
| ERD, SSVEP | Simultánea | Mejorada la precisión respecto a ERD y algo mejor que SSVEP |
| ERD, SSVEP | Secuencial | Reducción de la tasa de falsos positivos |
| P300, SSVEP | Secuencial | Mejora en la tasa de información |
| P300, ERD | Secuencial | Incremento de la fiabilidad |
| EEG, EMG | Simultánea | Mejoras en desempeño del sistema |

Tabla 2-2: Comparación entre distintos sistemas híbridos y las mejoras que han logrado respecto al uso de las tecnologías por separado (Amiri, Fazel-rezai, and Asadpour 2013).

2.3. PROBLEMÁTICA DE LAS INTERFACES CEREBRO-MÁQUINA

La obtención de información relevante de la actividad cerebral de un usuario no es una tarea sencilla y existen una serie de problemas que hay que tener en consideración antes de realizar una interfaz basada en fenómenos neurológicos.

La adquisición de las señales es una tarea compleja. Como se ha comentado anteriormente, existen métodos invasivos que pueden ofrecer una muy buena lectura de las señales deseadas pero muchas veces el utilizar este tipo de métodos no es una opción, bien por los riesgos para la salud que se puedan generar en los pacientes el hecho de tener que realizar incluso cirugía según los casos o bien porque se desea la creación de una interfaz que sea utilizable de la forma más cómoda posible. Existe, por tanto, un compromiso entre la calidad de la señal y el método de extracción utilizado (Konrad and Shanks 2010). El uso de métodos invasivos ha demostrado ofrecer un tipo de comunicación mucho más rápida y precisa que el uso de métodos de grabación no invasiva sobre el cuero cabelludo, ofreciendo un rendimiento que puede ser imprescindible para según qué tipo de interfaces y entornos (McFarland and Wolpaw 2011).

No se debe menospreciar el problema del filtrado de la información. Incluso utilizando métodos invasivos que proporcionen señales muy claras y con muy poco ruido queda el problema de filtrar la información relevante de aquella que no lo es. Hay que tener en consideración que en el cerebro se están realizando multitud de tareas continuamente y en paralelo, generando multitud de señales de muy diversa naturaleza que interferirán con aquellas que serán útiles para la interfaz. Para algunos tipos de señales, como P300 por ejemplo, la aparición de artefactos derivados de la realización de movimientos oculares, parpadeos, el estado anímico... puede llegar a hacer las señales muy poco reconocibles. Este problema es también muy perceptible para señales endógenas en donde, ya no solo es complicado el hecho de diferenciar los potenciales deseados del resto sino que, además, existe la posibilidad de que los usuarios no puedan generar señales codificables mediante la interfaz, como sucede para algunas personas con interfaces basadas en ritmos sensoriomotores (Nicolas-Alonso and Gomez-Gil 2012). Para poder discriminar las señales deseadas de aquellas que no lo son es necesario conocer la naturaleza y forma de las señales buscadas, así como la región del cráneo o cerebro desde la que resultan más fácilmente legibles.

Sin embargo, no basta solo con conocer las características de la señal deseada para la realización de la interfaz. El cerebro humano se puede considerar como un sistema que genera señales de tipo no estacionarias, tanto para señales endógenas como exógenas, lo que afecta a la forma en cómo se deben buscar estas señales. Esta característica implica que un mismo estímulo o una misma intención generarán una

señal diferente en el cerebro en cada instante de tiempo, lo que dificulta su identificación dado que no tendrán correspondencia inmediata con respecto a algún patrón. En este escenario hay que buscar ciertas características de la señal registrada que puedan ser lo suficientemente relevantes e invariantes como para permitir identificar la intención del usuario. Por ejemplo, cuando se analizan SSVEPs se espera que la señal leída tenga un pico en frecuencia cercano a la frecuencia del estímulo visual que el usuario está observando, aunque debido a todo lo comentado puede haber cierta deriva en frecuencia, variaciones en amplitud...

Además de la no estacionaridad de las señales deseadas y de la existencia de otros muchos potenciales simultáneamente en la región del cerebro a analizar hay que considerar también otros factores de ruido externo, incluyendo ruido introducido por el sistema de lectura de señales, riesgo de pérdida de información relevante debido a una mala digitalización o amplificación de la señal...

Dados los problemas anteriormente mencionados se puede deducir que la lectura de características relevantes mediante la encefalografía no es algo trivial. Por ello, se debe poner un gran cuidado en todas las etapas que componen toda la interfaz cerebro-máquina para que su fiabilidad sea aceptable. En el caso de señales de tipo endógenas, éstas son producidas en el interior del cerebro a partir de una intención o movimiento y sin ningún estímulo externo que sirva de guía. Esto explica el hecho de que el entrenamiento de las interfaces endógenas sea tan costoso en tiempo dado que no existe una referencia exterior al sistema no estacionario y éste puede estar sujeto a múltiples factores variables. Por otro lado, para interfaces de tipo exógenas se debe poner extremo cuidado en la generación de los estímulos, dado que la actividad cerebral a analizar dependerá directamente de ellos. El tratamiento en la generación de los estímulos será vital para que se generen patrones de señales en el cerebro que ayuden a deducir las intenciones del usuario, considerando que siempre habrá una cierta no estacionaridad.

2.4. APLICACIONES

Las aplicaciones de las interfaces cerebro-máquina son múltiples pero su utilidad viene dada por una serie de factores:

- **Movilidad de los usuarios:** Los BCIs han tenido típicamente el objetivo de mejorar las capacidades de aquellas personas que sufren algún tipo de discapacidad. Para personas que no tienen ningún problema motor la utilidad de estas interfaces es muy limitada y actualmente está prácticamente enfocada en el ocio.

- **Transferencia de información:** Una de las mayores desventajas de las interfaces cerebro-máquina es su baja transferencia de información. Algunas de las utilidades que podrían derivarse del uso de BCIs se ven dificultadas por el hecho de requerir una mayor transferencia de datos, especialmente si se trata de actividades de entretenimiento.

En función de los factores anteriores, se puede definir de forma gráfica la relación de la utilidad de los BCIs en función de las discapacidades de los usuarios y la transferencia de información necesaria para realizar determinadas tareas:

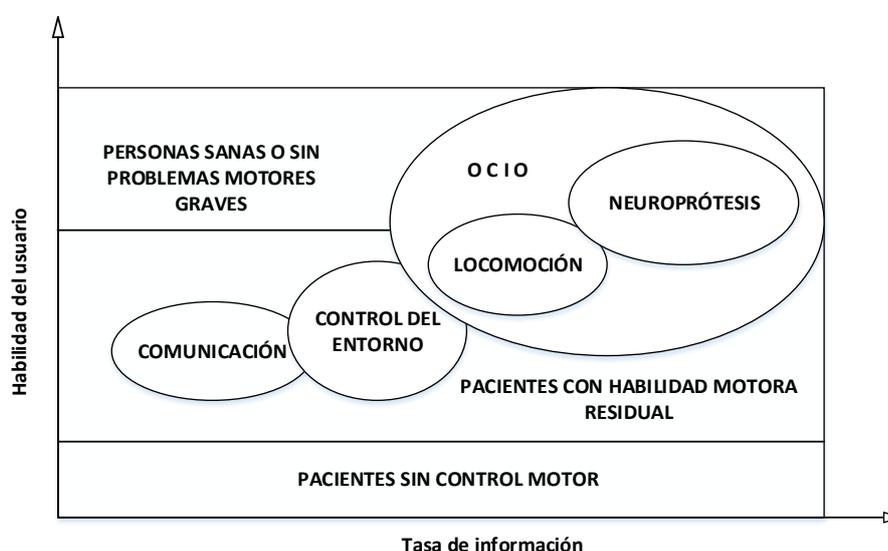


Figura 2-9: Aplicación de BCIs en función de la tasa de información y movilidad del usuario (Nicolas-Alonso and Gomez-Gil 2012).

Teniendo en cuenta los factores anteriores, existen múltiples aplicaciones en donde las interfaces cerebro-máquina se utilizan con éxito. Algunos ejemplos del potencial de esta tecnología pueden ser:

- **Comunicación** - Interfaces de ortografía (Chumerin et al. 2012): Unas de las aplicaciones más típicas para ayudar a las personas con escasa movilidad motora y que no pueden comunicarse son las interfaces de escritura; utilizando, por ejemplo, potenciales SSVEP es posible realizar una disposición de conjuntos de palabras o letras en una pantalla, cada uno de ellos parpadeando a una frecuencia diferenciable del resto, lo que permite al sistema descubrir qué palabras o letras está mirando el usuario. Esto le permite componer palabras complejas y frases al usuario, lo que le abre una nueva forma de comunicación.

- **Control del entorno** - Utilización de robots (Ortner, Guger, and Prueckl 2010): Las interfaces cerebro-máquina pueden hacer de intérpretes entre el usuario y un dispositivo robótico. Este dispositivo robótico puede sustituir alguna de las funcionalidades motoras del paciente, como puede ser el caso del uso de un brazo robótico. Se le pueden mostrar opciones de movimiento del dispositivo robótico al usuario mediante una interfaz evocadora que genere potenciales asociados en su cerebro, codificando dichos potenciales en movimiento real del robot. Esto le permite al usuario utilizar dispositivos de índole domótica, brazos robóticos... para sustituir en la medida de lo posible a sus extremidades.
- **Locomoción** - Sillas de ruedas (Nicolas-Alonso and Gomez-Gil 2012; Diez et al. 2013): De igual forma que sucede para el control del entorno, se puede utilizar un BCI para interpretar intenciones del usuario y que éstas se traduzcan en movimiento, en este caso de algún dispositivo tipo silla de ruedas que puedan paliar su necesidad de desplazamiento en caso de que tenga dicha zona afectada.
- **Neuroprótesis**: Las neuroprótesis consisten en la implantación directa de un chip en el cerebro con el fin de restaurar parte de la movilidad u otro tipo de deficiencias que pueda presentar el usuario.
- **Ocio** - Nueva interfaz para videojuegos (Krepki et al. 2007): Las interfaces cerebro-máquina ofrecen una nueva forma de interacción para que los usuarios, en este caso típicamente sanos, desarrollen acciones sobre algún tipo de aplicación o juego. Por ejemplo, se han desarrollado interfaces que permiten a los usuarios jugar a versiones del Comecocos mediante encefalografía. Además, el uso de estas interfaces en videojuegos ofrecen alternativas nunca antes vistas en este campo, dado que se puede analizar el estado emocional, ansiedad, cansancio... de los jugadores, modificando a partir de ello parámetros del juego en tiempo real o bien ayudando en diseños futuros.

Además de todo lo anterior, el uso intensivo de las interfaces cerebro-máquina puede conllevar evoluciones en lo que a la tecnología de captura de señal y análisis se refiere. Estas mejoras podrán ser utilizadas posteriormente en otros campos.

2.5. PERSPECTIVAS DE FUTURO

Además de las aplicaciones citadas anteriormente, existen grandes expectativas para desarrollar nuevas aplicaciones en el futuro que lleven la utilidad de las interfaces cerebro-máquina a un mayor nivel.

El uso de estas interfaces dentro del campo del ocio es relativamente novedoso y es de esperar que sufra grandes evoluciones, especialmente si se logra mejorar la tasa de transmisión de información, algo que puede ser necesario para la realización de determinadas acciones. Junto con otra serie de novedades en el campo, es de esperar que las interfaces cerebro-máquina formen parte del proceso de creación de realidades virtuales (Leeb et al. 2007) y sean capaces de ofrecer una nueva inmersión en estos entornos.

También será importante el desarrollo de la nanotecnología en la búsqueda de nuevas aplicaciones para los BCIs, algo que permitirá la creación de chips más pequeños y potentes, siendo esto de especial ayuda para incrementar las posibilidades de las neuroprótesis. En este contexto caben innovaciones que permitan el desarrollo de ojos biónicos, mejorar la tecnología utilizada para oídos biónicos... La simulación del funcionamiento del cerebro humano es algo también bastante perseguido y que puede ayudar a desarrollar nuevas interfaces con usos todavía por definir (Arafat 2013).

Sin embargo, el mayor reto pendiente es el de popularizar las interfaces cerebro-máquina. Se están dando los pasos adecuados en la investigación y desarrollo de los BCIs, incluida la reducción de los costes, pero es deseable que estos avances se transmitan a la sociedad de forma más generalizada y se tome una mayor conciencia de sus posibilidades. Al lograrlo se motivará un mayor crecimiento e innovación en el campo, pudiendo hacer realidad las perspectivas de uso en ocio y neuroprótesis anteriormente mencionadas y, adicionalmente, mejorando los usos ya existentes y popularizándolos especialmente en el área de las personas discapacitadas que necesitan alternativas viables que sustituyan sus capacidades afectadas.

Todo lo anterior depende de las mejoras en fiabilidad y rendimiento de las interfaces, pero sin comprometer su facilidad de uso dado que las interfaces invasivas que proporcionan mejores resultados no son viables más que en casos aislados. Por ello es de vital importancia que se realicen mejoras en los sensores y electrodos utilizados, especialmente para los casos de obtención de señal de manera no invasiva. Mejorar la relación señal a ruido, ofrecer una mejor resolución espacial e incluso llegar a poder realizar un análisis a capas más internas del cerebro sin necesidad de implantar electrodos dentro del mismo ayudaría a mejorar las prestaciones de los BCIs, especialmente los basados en EEG que es la alternativa más popular hasta este momento.

3. DISEÑO DE LA INTERFAZ

3.1. CONSIDERACIONES GENERALES

El objetivo del presente proyecto consiste en la creación de una interfaz cerebro-máquina con principal enfoque en la usabilidad del sistema y la adaptación al usuario, sin perder de vista la fiabilidad de funcionamiento. Esta interfaz tendrá que identificar las intenciones del usuario dentro de un conjunto de acciones posibles que podrán ser llevadas posteriormente a cabo por un dispositivo capaz de interactuar con el entorno. Teniendo en cuenta lo anterior se han tomado una serie de decisiones encaminadas a desarrollar el proyecto de la forma más viable e interesante posible.

En relación a la toma de señales cerebrales del usuario se ha optado por una aproximación no invasiva. Los motivos para ello son, esencialmente, la dificultad inherente de las alternativas invasivas, las cuales pese a proporcionar una mejor calidad de señal implican el uso de cirugía para la implantación de electrodos en las zonas de interés del cerebro, con el consiguiente alto riesgo para la salud del usuario. Dentro de los métodos no invasivos existen, como ya se ha visto, múltiples alternativas para realizar la adquisición de información. La electroencefalografía ofrece la mejor aproximación para los intereses de este proyecto: se trata de un método no invasivo, barato y cuya calidad final de señal es lo suficientemente buena para los objetivos que se persiguen, esto es, distinguir la acción que el usuario desea que se realice dentro de un conjunto no muy grande de acciones posibles. De cara a controlar un dispositivo como un brazo robótico o una silla de ruedas que permita a los usuarios que tengan algún tipo de minusvalía interactuar con el entorno, el tener en cuenta la resolución temporal también es importante, algo en lo que también la electroencefalografía cumple adecuadamente. Los otros métodos comentados en el análisis del estado del arte anterior, como la imagen por resonancia magnética funcional o la magnetoencefalografía implican métodos de obtención de datos más complejos y caros, siendo además sus ventajas como la alta resolución espacial irrelevantes para este proyecto.

Siendo la electroencefalografía la técnica de adquisición de información elegida, existen muchas alternativas para realizar la toma de datos. Algunos de los cascos que se utilizan para la adquisición de datos implican un montaje que puede ser largo y duro para el usuario debido a la complejidad de colocación de los electrodos. En este aspecto se volverá a poner énfasis en el que es uno de los pilares del proyecto: la usabilidad. Se experimentará con cascos de toma de datos cuyo montaje sea lo más inmediato y cómodo para el usuario, intentando evitar que ello comprometa en la medida de lo

posible la calidad de la señal. Es también importante señalar el bajo coste del casco utilizado, lo que puede contribuir aún más a la difusión de este tipo de interfaces.

Otro de los puntos a considerar a la hora de establecer el diseño del proyecto es el tipo de señales que se intentarían capturar del usuario. Alternativas como los ritmos sensoriomotores se han descartado por ofrecer una tasa de transferencia de datos muy baja y un tiempo de entrenamiento necesario muy elevado, características que comparten con los potenciales corticales lentos. Además la dificultad para controlar ambos sistemas por parte del usuario es bastante elevada y la eficiencia no es grande, dependiendo del estado emocional del usuario. Por tanto, las alternativas más propicias para la realización del proyecto son los potenciales evocados P300 o los SSVEP. Dada la latencia inherente a los potenciales P300, los SSVEP pueden alcanzar tasas de transferencias de información ligeramente mayores, contando con un tiempo de entrenamiento muy similar. Además, el Grupo de Neurocomputación Biológica cuenta con experiencia en el desarrollo de interfaces cerebro-máquina basadas en SSVEP, por lo que se ha optado por dar continuidad a esa alternativa.

Hay que tener en cuenta que las señales elegidas para su estudio son de tipo evocado. Por tanto, el desarrollo de la interfaz implicará también la generación de los estímulos para el usuario, que serán en este caso visuales. Existen dos formas fundamentales de afrontar esta parte del proyecto: crear una interfaz de estimulación basada en diodos LED o bien el uso de una pantalla genérica que genere estos estímulos. Se ha elegido realizar los estímulos a partir de una pantalla genérica de frecuencia de refresco de 60Hz por los retos que ello implica y por la flexibilidad para el diseño; la frecuencia de las señales SSVEP dependerán de la frecuencia con la que se estén mostrando los estímulos en el sistema evocador, estando éste limitado a 60Hz. Por tanto, se realizará un estudio de cómo desarrollar la interfaz gráfica y cómo se representan frecuencias no divisoras de 60Hz, y si existe una deriva importante en frecuencia para esos estímulos o si son utilizables de cara a desarrollar la interfaz. Es importante tenerlo en cuenta para poder ampliar el abanico de frecuencias disponibles para la estimulación.

En este punto solo queda por determinar qué tipo de procesado y análisis de señal se va a realizar y cómo se realizará la identificación de la frecuencia del estímulo al que el usuario estaría observando. A la hora de realizarlo se ha de tener en consideración que, producto de la no estacionaridad de las señales evocadas en el cerebro, pueden darse derivas en la frecuencia de las mismas, entre otro tipo de alteraciones que puedan sufrir. Se tendrá también en cuenta otra serie de parámetros como la frecuencia de muestreo del casco, qué señal o señales del mismo se utilizarán en el análisis, filtrado, inventariado...

El correcto funcionamiento global del sistema depende, en gran medida, del entrenamiento. El desempeño de las señales evocadas es distinto para cada usuario, siendo algunas personas más vulnerabilidad a determinadas frecuencias de parpadeo por parte de los estímulos visuales. La búsqueda personalizada de estas frecuencias

óptimas que generen señales cerebrales más distintivas será determinante para maximizar la eficiencia del sistema.

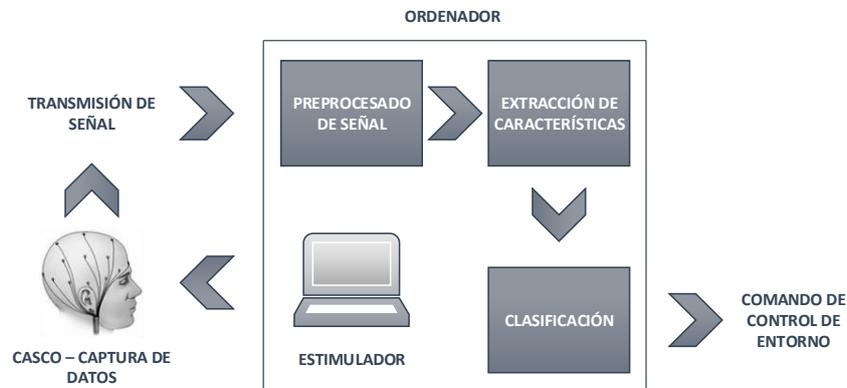


Figura 3-1: Esquema general de la estructura general que tendrá la interfaz propuesta.

3.2. CAPTURA DE DATOS

Existen múltiples alternativas para la captura de datos de señales de forma electroencefalográfica. El hecho de ser una técnica no invasiva no implica que exista cierto grado de incomodidad de uso por parte del usuario. En la medida de lo posible se utilizará la alternativa más cómoda de uso. Además es importante que el casco permita acceder a la zona en la que las señales evocadas son mesurables de forma más sencilla. En el caso de los SSVEP, estos aparecen en la corteza visual en la parte occipital del cráneo, por lo tanto será necesario que el casco a utilizar permita emplazar electrodos cerca de dicha localización.

La localización de los electrodos en los cascos de electroencefalografía se basa en estándar internacional 10-20. Según dicho estándar, la localización más propicia para los electrodos de cara a capturar señales de tipo SSVEP corresponde a la posición OZ, siendo las posiciones O1, O2 y POZ suficientemente cercanas como para ser tenidas en cuenta. Algunos cascos comerciales de un uso casi doméstico no tienen electrodos en dicha zona. Existen también gorros con aberturas siguiendo el estándar anteriormente citado sobre los cuales se introducen los electrodos, estando el gorro ya colocado sobre la cabeza del usuario. Esta última aproximación, pese a ser no invasiva, puede derivar en una experiencia poco cómoda para el usuario pues, en ocasiones, la colocación de estos electrodos no es trivial. También está la alternativa de poder colocar los electrodos a placer con ayuda de unas ventosas especiales o algún otro tipo de dispositivo fijador.

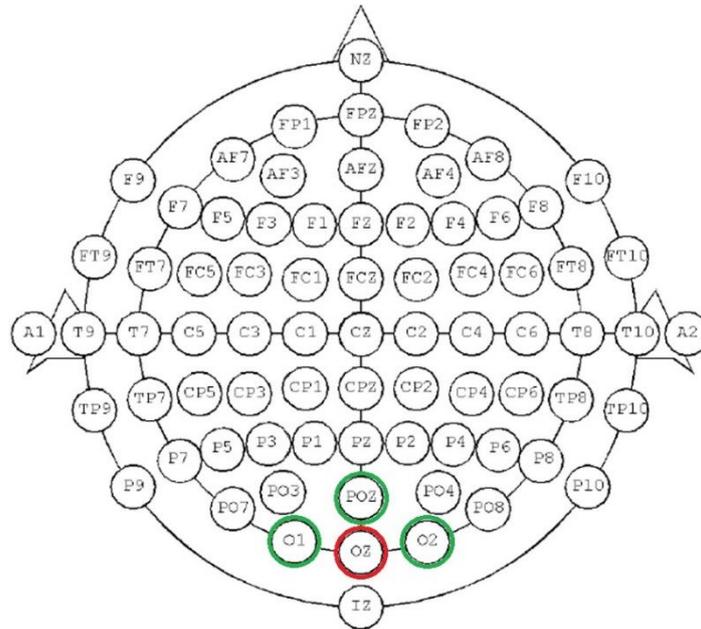


Figura 3-2: Colocación de electrodos según el estándar internacional 10-20, resaltada en rojo la zona óptima para SSVEP y en verde las zonas próximas de interés.

El Grupo de Neurocomputación Biológica cuenta con dos alternativas a considerar para la captura de SSVEP:

- **Gorro de licra Aegis Array y amplificador BRAINBOX® EEG-1166 de 64 canales de Braintronics:** Capaz de capturar la señal a 1024Hz. Este casco permite colocar electrodos en las posiciones de interés. Sin embargo, no es un casco cuya colocación sea trivial, llegando a necesitar bastantes minutos para ello y la ayuda de una segunda persona.
- **Casco de propósito general Emotiv EPOC:** Se trata de un casco comercial apto tanto para el entretenimiento como para la investigación. Dispone de una frecuencia de muestreo a 128Hz y, pese a no disponer de electrodos en la posición óptima OZ, si se pueden emplazar electrodos en O1 y O2. Su colocación es trivial y la amplificación y conversión analógico/digital de señal es realizada por el propio casco.

Dadas las consideraciones anteriores y las alternativas disponibles se ha optado por utilizar el casco Emotiv EPOC (<http://www.emotiv.com>) debido a que se trata de la opción más asequible y apropiada en términos de usabilidad. Se tendrá que realizar una evaluación de cómo puede afectar el hecho de no disponer de electrodo en la posición OZ y si las señales que se puedan adquirir en posiciones aledañas serán lo suficientemente claras como para permitir el desarrollo de la interfaz.

3.3. INTERFAZ GRÁFICA

Para que los estímulos evocados sean generados de forma inequívoca es necesario que la estimulación visual sea la adecuada. Como los potenciales SSVEP que se utilizarán para este proyecto se generan a partir de estímulos parpadeantes será necesario que el estimulador muestre una serie de destellos y objetos parpadeantes. En este caso se va a utilizar una pantalla genérica, pero no es la única opción; el uso de dispositivos basados en LEDs de colores con hardware dedicado puede proporcionar una respuesta más distintiva en el análisis de las señales cerebrales. En este proyecto, sin embargo, se ha optado por el uso de una pantalla común con un refresco de 60Hz con la intención de crear una interfaz que sea lo más universal posible y no dependa de la creación de un hardware particular.

Se barajarán distintas librerías gráficas para proporcionar la estimulación gráfica de cara a evaluar el funcionamiento de las mismas y utilizar la que más se adapte a los objetivos que se pretenden. Esto es necesario porque existen múltiples factores que pueden alterar la correcta generación de estímulos:

- **Debidos al estimulador (pantalla):** Como ya se ha mencionado anteriormente, la frecuencia de refresco de la pantalla es una barrera importante a la hora de utilizarla como estimulador para eventos de tipo parpadeantes. Uno de los puntos de interés será comprobar cómo se comportan distintas librerías gráficas para frecuencias no divisoras de 60Hz de cara a ampliar el abanico de posibles frecuencias utilizables en el análisis.
- **Debidos al hardware y gestión del tiempo:** Algunas librerías gráficas pueden ser más sensibles a la carga de trabajo a la que esté sometido el hardware de la máquina que otras, hasta llegar al punto en que se muestren inestables para mostrar un evento parpadeante o, simplemente, no cuenten con operaciones de gestión del tiempo lo suficientemente precisas.

Otro factor a considerar será qué tipo de estímulos se mostrarán. En general, para interfaces de este tipo se suelen utilizar dos tipos de imágenes parpadeantes: imágenes planas o imágenes de tipo tablero de ajedrez. Se realizarán estimulaciones a usuarios para determinar si en el caso que cubre este proyecto es deseable el uso de un tipo de imágenes por encima de otro o si ambos tienen un desempeño final similar. También serán caso de estudio otra serie de factores para la estimulación como la distancia de los ojos respecto de la pantalla, la introducción de colores en los estímulos...

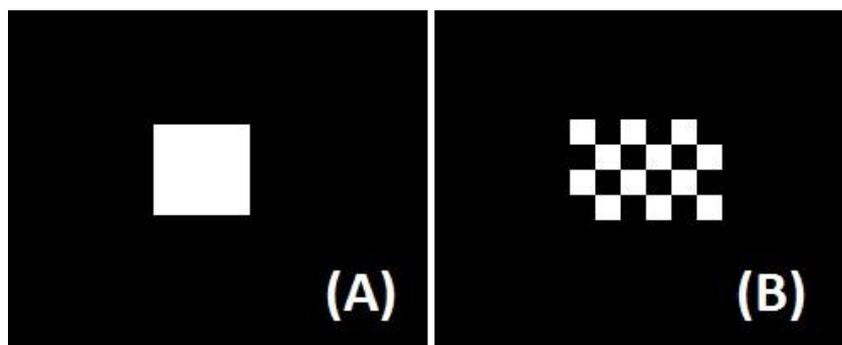


Figura 3-3: (A) Representación de único estímulo plano y (B) estímulo tipo tablero de ajedrez.

3.4. ANÁLISIS DE SEÑAL

Los apartados anteriores han servido para cubrir la problemática de realizar la extracción de información electroencefalográfica y las consideraciones a la hora de generar los estímulos. El otro gran tema a tratar una vez que se han generado los estímulos y se ha obtenido la señal es el de analizar los datos que se hayan recogido.

El sistema a crear tiene que funcionar en un entorno online a tiempo real. Esto limita la cantidad de muestras que se pueden recoger para realizar el análisis de cara a no obtener un resultado con mucho retardo. Sin embargo, ventanas de tiempo muy cortas pueden comprometer la correcta adquisición de características de la señal. Para ello se tendrá muy en cuenta la técnica del solapado entre ventanas, de forma que permita ampliar la región temporal de análisis sin que ello se traduzca en un retardo perceptible para el usuario, algo que en este tipo de dispositivos podría ser muy molesto e incluso peligroso. En cualquier caso, al realizar el eventanado habrá que tener siempre presente que se producirá un efecto de borde que alterará la señal para su análisis.

Cuando se analizan señales de tipo SSVEP se observa que, además de presentar picos distintivos en la frecuencia a la que se realiza la estimulación también existe una gran importancia a la hora de evaluar los armónicos. Esto se puede considerar información muy útil a la hora de realizar el análisis de señal y supone también un compromiso a la hora de realizar el eventanado dado que existen distintos tipos de ventanas, algunas de las cuales magnifican las frecuencias fundamentales a costa de atenuar irreversiblemente los armónicos, otras atenúan un poco dichas frecuencias fundamentales pero permiten un análisis de los armónicos... Adicionalmente a las consideraciones del eventanado, habrá que realizar las correspondientes transformadas discretas de Fourier para el correcto análisis del espectro de la señal, para lo cual se hará uso del algoritmo FFT (*Fast Fourier Transform*).

Siempre hay que tener presente que se está tratando con señales de tipo no estacionario. Además, estas señales estarán sometidas a ruido, siendo crítico en el análisis el ruido procedente de otros procesos que se estén realizando en el cerebro o a interferencias externas. En este escenario surge la duda de cómo filtrar correctamente la señal.

Una de las opciones para limitar el ruido es monitorizar señales procedentes desde distintas partes del cerebro para evaluar la interferencia que se produce debido a otros procesos que se estén realizando en él. Tomando estos datos, se puede eliminar parte del ruido procedente de otras partes del cerebro que están interfiriendo con la zona de toma de datos que, como se ha visto, para SSVEP es la posición Oz.

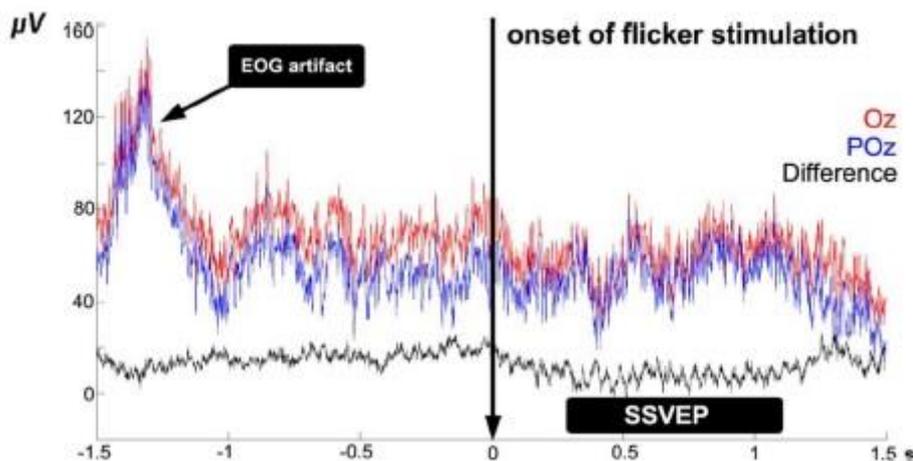


Figura 3-4: Ejemplo de monitorización de SSVEP utilizando la resta de señales entre las posiciones Oz y POz. Al comienzo de la grabación se observa mucho ruido producido por un artefacto (ej. parpadeo) que es completamente anulado al hacer la diferencia entre señales. Tras el encendido del estímulo en el segundo 0 se pueden apreciar los potenciales evocados con la periodicidad que caracteriza al estímulo (Fernandez-Vargas et al. 2013).

Las operaciones entre distintas señales pueden ayudar a eliminar parte del ruido que se esté produciendo en ese momento debido a eventos particulares que se estén llevando a cabo en el cerebro. No obstante, el cerebro tiene siempre una gran actividad de fondo al margen de estos eventos particulares. Para eliminar este ruido permanente se hace uso de la grabación de un *baseline*. Esto consiste en monitorizar la actividad del cerebro sin ninguna estimulación, esto es, sin ningún estímulo en pantalla o bien sin ningún LED encendido en el caso en el que se esté utilizando un dispositivo de estimulación dedicado. El fin de esta grabación será monitorizar la actividad de fondo del cerebro cuando no está sometido a la estimulación visual, esto es, lo que

consideraremos ruido. Realizando posteriormente la división de las señales capturadas durante la estimulación entre este *baseline* se obtendrá la relación señal a ruido y, con fortuna, se apreciarán los potenciales evocados en las frecuencias en las que el estimulador esté parpadeando.

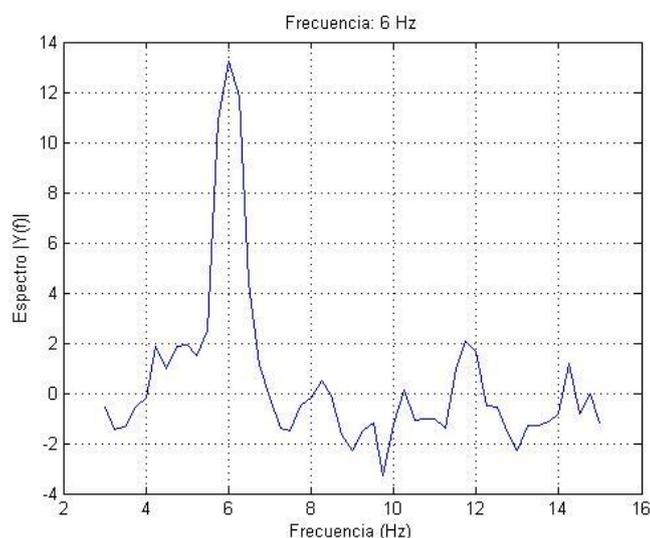


Figura 3-5: Ejemplo de espectro ideal de señal monitorizada (dB) tras aplicar la relación señal a ruido a través del baseline y operar entre los canales O2-AF4 estimulando a 6Hz. Se observa el pico esperado en 6Hz y valores por encima de la unidad en alguno de los valores adyacentes y en su primer armónico (12Hz).

3.5. SELECCIÓN DE FRECUENCIAS

Se ha comentado con anterioridad la problemática especial de crear una estimulación adecuada a partir de una pantalla típica de refresco a 60Hz, debido a la dificultad de la misma de representar con suficiente precisión temporal alguna de las frecuencias que pudieran ser útiles para la generación de los SSVEPs. No es, sin embargo, el único problema a considerar cuando se trata de elegir las frecuencias de estimulación óptimas para el sistema. El usuario de una interfaz cerebro-máquina basada en SSVEPs puede ser más sensible a unas determinadas frecuencias que a otras a la hora de realizar la estimulación (Fernandez-Vargas et al. 2013; Ivan Volosyak et al. 2011; I. Volosyak, Cecotti, and Gräser 2009). En este sentido es de vital importancia que la interfaz se pueda adaptar a cada usuario, sin perder de vista las limitaciones que pueden tener el estimulador y el propio casco de captura de datos.

Para intentar obtener el mejor desempeño final del sistema se optará por dos vías diferenciadas:

- **Uso de frecuencias prefijadas:** En este caso se elegirían una serie de frecuencias de estimulación cuyos valores dependerían de la distancia entre ellas para evitar interferencias entre los valores adyacentes, la capacidad de la pantalla de representarlas fielmente y evitando la presencia en la elección de frecuencias de pares de frecuencia fundamental y primer armónico en el mismo set de frecuencias de estimulación.
- **Uso de algoritmos de selección de frecuencias:** Las ventajas del uso de uno de estos algoritmos será adaptarse a las frecuencias que generan señales más distintivas en el cerebro de cada usuario. Ello implica relajar alguna de las restricciones que se impondrían en un caso de frecuencias prefijadas con el fin de ampliar el abanico de opciones disponibles para realizar la adaptación.

En el presente proyecto se realizará una comparación entre el uso de frecuencias prefijadas de acuerdo a las características anteriores y el uso de algoritmos de selección de frecuencias óptimas para cada usuario mediante el uso de una exploración en ciclo cerrado; si el uso de estos algoritmos implica un mejor desempeño de la interfaz que el uso de frecuencias prefijadas implicará que conviene más adaptarse a las características del usuario que centrarse estrictamente en las posibles limitaciones del sistema, como la frecuencia de refresco de la pantalla.

4. DESARROLLO

4.1. CAPTURA DE DATOS MEDIANTE EL EMOTIV EPOC

El Emotiv Epoc (Emotiv 2014) es el medio de adquisición de datos de EEG que se utilizará en el presente proyecto. Se trata de un casco cuya principal virtud es su fácil configuración y uso, requiriendo unos pocos minutos de preparación por parte del usuario que lo vaya a utilizar. El casco cuenta con 14 canales de detección de señales EEG y dos referencias, que toman lecturas de la actividad cerebral a una frecuencia de 128 Hz y realizándose la transmisión de datos desde el casco al ordenador de forma inalámbrica. Adicionalmente cuenta con dos giroscopios. Presenta, además, todo tipo de facilidades ya que cuenta con un software dedicado con el que se puede acceder

directamente a los datos tomados por el casco, aplicaciones que interpretan dichos datos intentando discernir de ellos el estado de ánimo del usuario, gestos faciales... Más relevante para el uso que se le dará en esta ocasión es el hecho de que se puede acceder al estado de calidad de señal de los electrodos y grabación de los datos que se están tomando mediante la aplicación Testbench.

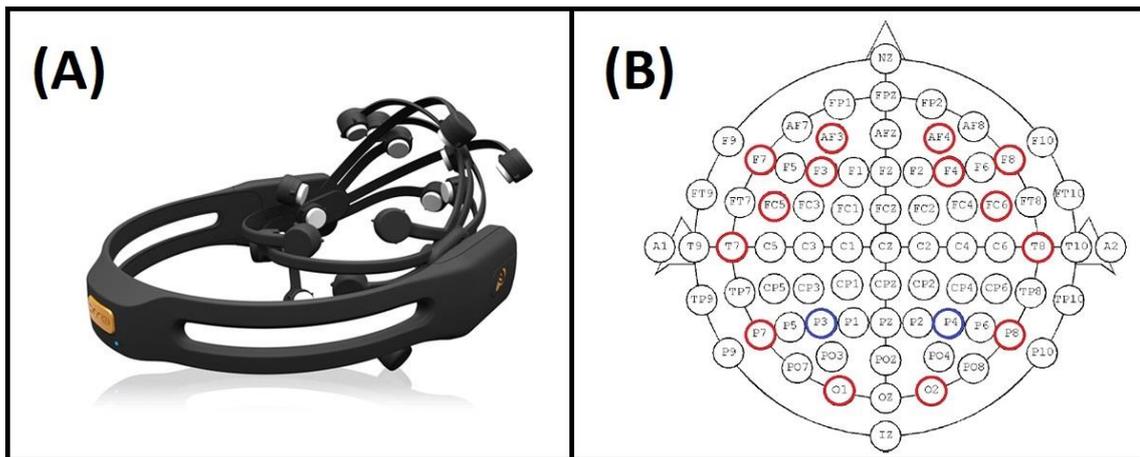


Figura 4-1: (A) Emotiv Epoc y (B) disposición de los 14 electrodos en las posiciones AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8, AF4 y con las dos referencias en P3 y P4.

La principal desventaja de usar este casco para la detección de SSVEP es que no cuenta con un electrodo en la posición OZ, parte occipital del cerebro donde se generan las señales deseadas. Para intentar solventar este problema se han hecho pruebas utilizando distintos canales que ofrezcan una buena detección de estas señales y, a la vez, un buen comportamiento frente al ruido procedente desde otras partes del cerebro. Las condiciones bajo las que se han realizado las pruebas son las siguientes:

- **Baseline:** Se ha optado por grabar una línea de *baseline* muy larga de 60 segundos de duración de cara a monitorizar toda la actividad de fondo del cerebro con la mayor fidelidad posible, así como registrar también eventos como parpadeos y pequeños movimientos involuntarios.
- **Estimulación:** Sesiones de estimulación largas, de unos 20 segundos, con eventos parpadeantes a bajas frecuencias, entre los 6 Hz y los 13 Hz.

Esta primera fase de experimentación tiene como objetivo comprobar si este casco es una alternativa viable para la grabación de los SSVEP. Por ello mismo el análisis de las señales que se han tomado se ha realizado de manera offline y, aunque se ha realizado

eventanado de la señal de cara a fases posteriores del proyecto, este primer análisis se han utilizado con condiciones tales que la señal no sea muy alterada al realizar las sucesivas operaciones sobre ella:

- Ventanas de gran longitud, 5 segundos a 128 muestras por segundo.
- Desplazamiento entre ventanas de solo 64 muestras.
- Eventanado de tipo Hanning, para no atenuar en exceso los armónicos.
- Consideración de armónicos y subarmónicos en la puntuación final de cada frecuencia.

Para realizar la estimulación se ha utilizado una interfaz gráfica con la que contaba con anterioridad a la realización del presente proyecto el Grupo de Neurocomputación Biológica en su laboratorio, utilizando SDL (*Simple DirectMedia Layer*), cuyo desempeño para frecuencias bajas es lo suficientemente bueno como para realizar una primera búsqueda de SSVEPs y verificar si el casco puede detectar las señales. No obstante, en este proyecto se analizará si se trata de la mejor alternativa para la generación de estímulos. El objetivo de este experimento es analizar de forma visual con el análisis anterior mediante MATLAB si las componentes frecuenciales que se deberían observar gracias a los SSVEP en la parte occipital del cerebro son capturadas mediante el casco. Para ello se ha realizado el experimento anteriormente descrito con una pequeña muestra de 5 sujetos dado que no se ha considerado necesario el realizar el experimento con un número elevado de personas ya que solo se necesita conocer la viabilidad del casco para la realización del proyecto. Adquiridos los datos, en el análisis offline se ha buscado la mejor combinación posible entre las 14 señales de las que el casco dispone. Como se pretende capturar las señales en la parte occipital, los electrodos en las posiciones O1 y O2 son de obligado uso. Se ha considerado el utilizar una serie de electrodos más en la parte frontal de la cabeza con el fin de atenuar el ruido procedente de dichas zonas de no interés. Las combinaciones que finalmente dieron unos resultados alentadores en la búsqueda de los SSVEP fueron:

- **Hemisferio izquierdo:** Combinaciones de señal O1-F3, O1-T7 y O1-AF3 válidas en casos muy concretos y solo a muy bajas frecuencias (entre 6 Hz y 8 Hz).
- **Hemisferio derecho:** Combinaciones de señal O2-F4, O2-T8 y O2-AF4 válidas en general y en frecuencias más elevadas, incluso hasta el límite de 13 Hz impuesto en esta prueba inicial.
- **Combinación de hemisferios:** Desempeño similar al hemisferio derecho, con un poco más de protección frente al ruido. Las combinaciones que mejor resultado han dado en este caso son (O1+O2)-(F3+F4) y (O1+O2)-(AF3+AF4).

Dentro de todas las alternativas viables observadas se ha optado por decidir entre aquellas que son resultado de la combinación entre hemisferios pues el uso de la lateralidad puede hacer el desempeño de la interfaz muy dependiente del usuario en concreto, dado que en algunos de ellos se observan mejor los SSVEP en un hemisferio que en otro. Como se pretende, sin embargo, que este BCI funcione de la forma más universal posible, la lateralidad queda descartada puesto que no ofrece muchas mejores prestaciones aunque podría explorarse en un futuro para hacer la interfaz más adaptativa a cada usuario. Finalmente, dentro de las alternativas viables en combinación de hemisferios aquella que da mejores resultados es la operación $(O1+O2)-(F3+F4)$ y será, por tanto, la utilizada.

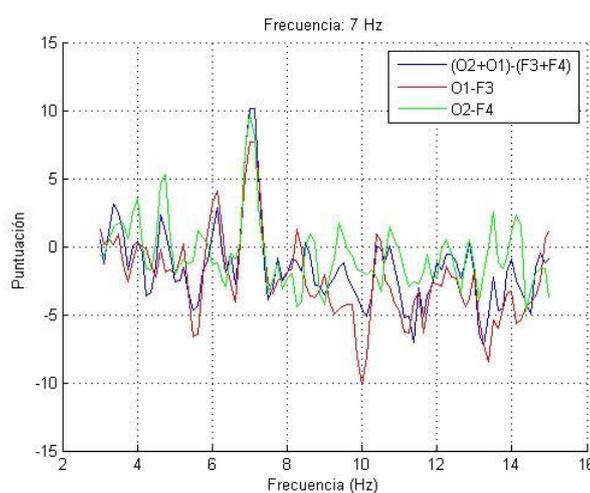


Figura 4-2: Ejemplo de resultados para un usuario concreto a 7Hz con los mejores exponentes de ambos hemisferios y su combinación. Se aprecia como a la frecuencia de estimulación se obtiene una mayor puntuación con la combinación, mientras que al resto de frecuencias obtiene un desempeño de ruido aceptable.

En pruebas posteriores se intentó probar el desempeño del casco en otros rangos de frecuencias superiores y hasta los 20 Hz, una vez elegidas las señales útiles para el análisis. A medida que se incrementaba la frecuencia de estimulación se podía observar como la importancia de los armónicos crecía, pese a ser la frecuencia fundamental a veces indetectable. Esto pone de manifiesto la necesidad de utilizar los armónicos en los análisis de señal y detección de SSVEP, dado que nos permitirán ampliar el rango de frecuencias útiles.

Según lo comentado anteriormente, el Emotiv Epoc es, por tanto, un casco viable para realizar el análisis de SSVEP. Se observa, no obstante, que la relación de la señal respecto al ruido no es todo lo alta que sería deseable y es algo que puede llevar a falsas detecciones. Especialmente, cuando se utilizan ventanas más pequeñas, puede haber

momentos en los que las detecciones son poco claras y predomine el ruido. Se intentará paliar estos efectos cuando se realice la implementación del análisis de señal online.

4.2. USO DEL SDK DE EMOTIV EPOC

Emotiv proporciona, junto al casco, un SDK (*Software Development Kit*) para poder desarrollar aplicaciones basadas en el uso del EPOC. Las funciones y librerías que se describen están escritas en C/C++, que será el lenguaje que se utilizará para el resto del proyecto. La API (*Application Programming Interface*) está declarada en tres cabeceras (edk.h, EmoStateDLL.h, edkErrorCode.h).

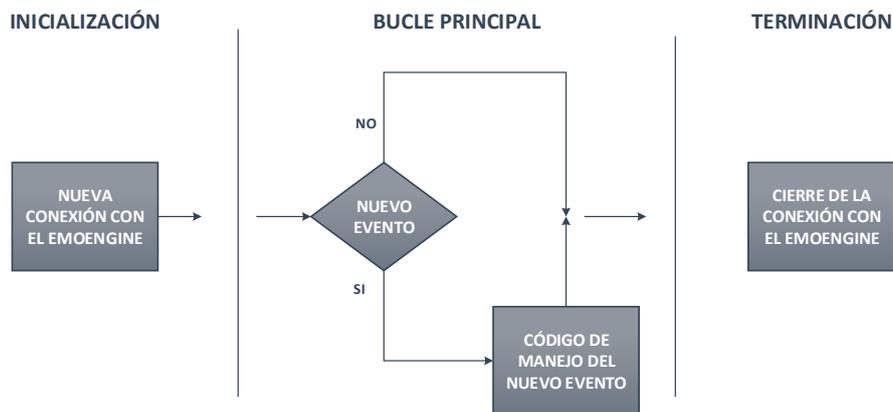


Figura 4-3: Esquema de uso de la API para conectarse con el EmoEngine proporcionado por el manual del SDK de Emotiv.

La abstracción lógica referida a la funcionalidad proporcionada en las cabeceras anteriores y en los archivos asociados se denomina EmoEngine. El EmoEngine es el encargado de comunicarse con el casco Emotiv EPOC y recibirá los datos proporcionados por los 14 canales que monitorizan la actividad EEG y los giroscopios, proporcionando además gestión de usuarios, postprocesado de señal... y, finalmente, transformará y traducirá toda la información de los datos tomados a una estructura llamada EmoState.

Dentro del programa que se ha desarrollado y siguiendo el esquema anterior, las llamadas a las funciones proporcionadas por la API son las siguientes, de forma esquemática:

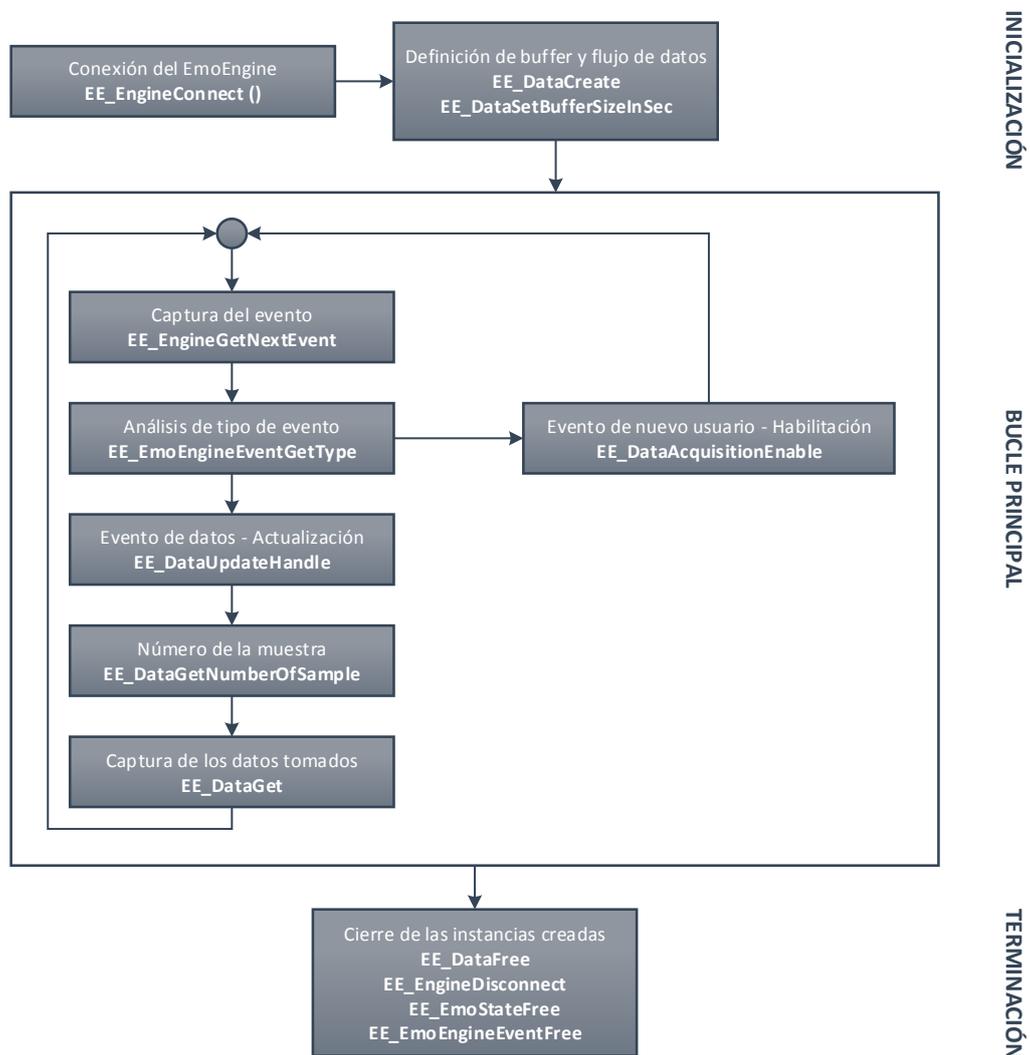


Figura 4-4: Secuencia de llamadas a las funciones proporcionadas por la API para el control de los datos proporcionados por el casco.

Lo anterior permite ir almacenando los datos proporcionados por el casco en tiempo real. Dentro del código, este control se encuentra en la función siguiente, contenida en el archivo `interfazEEG.cpp` (para mayor información sobre la estructura general del programa creado consultar el Apéndice I – Organización de archivos):

```

/**
 * @brief envioDatosCasco: Función encargada de capturar los datos del casco Emotiv Epoc mediante el SDK
 proporcionado
 * @param idCola: Identificador de la cola a la que se enviarán los datos capturados por el casco
 * @param segundosSumulacion: Duración máxima del bucle principal de ejecución (opcional)

```

```
* @return 0 (correcto) o 1 (error)
*/
int envioDatosCasco(int idCola, int segundosSimulacion)
```

Sin embargo, para realizar el análisis de forma online de las señales que se están recibiendo será necesario otro proceso en paralelo que sea el encargado de gestionar esta información. Como el programa ha sido realizado en un entorno de tipo Linux, para la apertura de los distintos procesos necesarios se hará uso de la instrucción *fork()*. Para comunicar el proceso capturador de datos con el/los procesos analizadores de información se hará uso de las llamadas al sistema *msgrcv* y *msgsnd* que permiten enviar y recibir mensajes desde una cola caracterizada por su ID. La unidad de información que manejará esta cola es una estructura que ha sido creada para el proyecto de la siguiente forma:

```
// Tipos de mensajes
#define IPC_MENSAJE_CABECERA 2
#define IPC_MENSAJE_DATOS 1
#define IPC_MENSAJE_FIN_TRANSMISION 9

typedef struct
{
    long tipoComunicacion;           // Tipo de mensaje enviado
    double texto;                   // Valor del dato tomado por el casco
} MensajeIPC;
```

Esta estructura permite identificar el tipo de mensaje que se envía y su contenido de forma que se pueda sincronizar el inicio y el fin de la transmisión de datos con tan solo definir este tipo de comunicación. Cuando se trata de un mensaje de datos, la estructura es rellenada para cada uno de los canales que se utilizarán en cada período de tiempo. Esto significa que cada canal de información meterá 128 mensajes en la cola por segundo, dados los 128 Hz de frecuencia de toma de datos del casco. Los canales se describen en el código de la siguiente forma:

```
// Lista total de canales a considerar por el EmotivEpoC
EE_DataChannel_t targetChannelList[] = {
    ED_COUNTER,
    ED_AF3, ED_F7, ED_F3, ED_FC5, ED_T7,
    ED_P7, ED_O1, ED_O2, ED_P8, ED_T8,
    ED_FC6, ED_F4, ED_F8, ED_AF4, ED_GYROX, ED_GYROX, ED_GYROX, ED_TIMESTAMP,
    ED_FUNC_ID, ED_FUNC_VALUE, ED_MARKER, ED_SYNC_SIGNAL
};
```

Aunque, como ya se ha definido, para el proyecto solo serán necesarios 4 de estos canales, se ha optado por enviar todos los mensajes a las colas entre los distintos procesos por simplicidad y en previsión de poder utilizar otro tipo de canales para otras aplicaciones de manera más inmediata. En total, se enviarán 22 mensajes por cada muestra tomada, siendo el número de muestras 128 por segundo, lo que significa 2816 estructuras de tipo MensajeIPC por segundo que tienen que ser enviadas y leídas de las colas por los respectivos procesos implicados. Pese a ello no ha habido problemas de desbordamiento de la capacidad de las colas en ninguna de las pruebas realizadas con este método. Se utilizaron otros métodos previamente para establecer esta comunicación entre procesos como el uso de tuberías FIFO (con los comandos *mkfifo*, *write*, *read*...), implementando memoria compartida FIFO (*shmfifo*, *shmfifo_get*, *shmfifo_put*...) pero ninguna de las alternativas dio tan buen resultado como la mencionada anteriormente, especialmente debido a problemas de sincronización entre procesos.

4.3. DESARROLLO DE LA INTERFAZ GRÁFICA

La interfaz gráfica es la encargada de generar los estímulos al usuario que generará las señales SSVEP evocadas. Por ello, es de vital importancia que las frecuencias de parpadeo de los iconos que se han de mostrar sean lo más exactas posibles a lo pretendido. También será imprescindible evaluar cómo se comportará la interfaz gráfica para mostrar parpadeos no divisores de la frecuencia de refresco de la pantalla.

Dadas las consideraciones anteriores se han elegido una serie de librerías y se ha comparado su rendimiento entre ellas:

- **SDL (Simple DirectMedia Layer):** Conjunto de librerías escritas en C que proporcionan funciones básicas para la representación gráfica en dos dimensiones, control de sonido o música y gestión de imágenes. Entre sus ventajas se puede contar la sencillez de uso para representación de imágenes bidimensionales y el soporte multiplataforma (<https://www.libsdl.org>).
- **OpenGL (Open Graphics Library):** Conjunto de librerías en C con el fin de escribir aplicaciones gráficas bidimensionales y tridimensionales. Cuenta con centenas de funciones dedicadas a este apartado y proporciona un método sencillo para construir imágenes a partir de figuras geométricas simples mediante la definición de puntos, líneas y otras formas. Cuenta con soporte multiplataforma (<https://www.opengl.org>).

- **OpenCV (Open Source Computer Vision):** Conjunto de librerías en C++ cuyo fin principal es el de la gestión, análisis y procesado de imágenes en tiempo real, aunque su uso como generador de imágenes es algo más limitado que otras alternativas. Como todas las alternativas anteriores, cuenta con soporte multiplataforma (<http://www.opencv.org>).
- **Qt:** Biblioteca multiplataforma para C++ ampliamente utilizada para el desarrollo de interfaces de usuario que también cuenta con una serie de herramientas para el desarrollo de programas bajo línea de comandos. Cuenta con un gran soporte y desarrollo para su uso como generador de interfaces bajo prácticamente cualquier dispositivo, incluidos dispositivos móviles (<http://www.qt-project.org/>).

Se ha desarrollado interfaces encargadas de mostrar un único estímulo en pantalla para cada una de las alternativas anteriores de cara a evaluar el comportamiento de todas ellas. Los criterios que se han utilizado para evaluar el rendimiento son:

- Simplicidad en la programación.
- Fidelidad al mostrar estímulos parpadeando a una frecuencia dada.

Las frecuencias utilizadas para realizar las pruebas para cada una de las librerías han sido una mezcla entre frecuencias naturalmente representables por una pantalla clásica de refresco a 60 Hz y frecuencias no divisoras, para las cuales la capacidad de representación de la pantalla se verá comprometida. El conjunto de frecuencias de parpadeo utilizadas en los test son:

- Divisores de 60: 6 Hz, 10 Hz, 15 Hz, 20 Hz, 30 Hz.
- No divisores de 60: 8 Hz, 13 Hz, 17 Hz, 25 Hz, 32 Hz.

También se realizaron experimentos en función del tipo de imagen mostrada. Se hicieron tres distinciones:

- Imágenes pequeñas: Cuadros blancos de 150 píxeles de lado.
- Imágenes grandes: Cuadros blancos de 400 píxeles de lado.
- Imagen tipo tablero de ajedrez: Formado por pequeños cuadros de 70 píxeles de lado, formando una imagen de lado total de 350 píxeles.

Tras las primeras pruebas realizadas se demostró que el tamaño o forma de las imágenes no influía en la capacidad de la pantalla para representarlas en ninguna de las frecuencias seleccionadas.

Teniendo en cuenta lo anterior, los resultados obtenidos a distintas frecuencias para imágenes pequeñas por las distintas librerías son los siguientes:

| Frecuencia (Hz) | 6 | 8 | 10 | 13 | 15 | 17 | 20 | 25 | 30 |
|-----------------|-----|-----|-----|------|------|------|------|------|------|
| SDL | 5.5 | 7.2 | 10 | 12.5 | 12.5 | 16.7 | 16.7 | 25 | 25 |
| OpenGL | 6 | 7.5 | 9.9 | 13 | 15 | 15 | 15 | 15 | 15 |
| OpenCV | 4.5 | 5.5 | 6.4 | 7.5 | 8.2 | 8.4 | 9.3 | 10.4 | 11.3 |
| Qt | 6 | 8.1 | 10 | 13.1 | 15.1 | 17.2 | 20 | 25 | 28.8 |

Tabla 4-1: Resultados obtenidos al intentar mostrar un evento parpadeante a determinada frecuencia entre distintas librerías gráficas

Según los resultados que se muestran en la tabla anterior, la alternativa de OpenCV queda directamente descartada. No es sorprendente dado que es una librería cuyo fin no es el de generar y mostrar imágenes, si no el de analizar imágenes en tiempo real. Se puede apreciar de forma constante como el evento parpadeante es incapaz de seguir la frecuencia deseada. Por otro lado, SDL muestra de forma incorrecta frecuencias concretas y es incapaz de mostrar eventos más allá de los 25 Hz. OpenGL ha demostrado una buena fidelidad a la hora de intentar reproducir imágenes parpadeantes a las frecuencias deseadas hasta los 15 Hz. Sin embargo, en ningún momento ha sido capaz de mostrar estos parpadeos por encima de dicha frecuencia. Qt, por otro lado, ha mostrado un comportamiento excepcional en todo rango de frecuencias hasta los 30 Hz (en ningún momento se esperaba que ninguna librería pudiera mostrar los 30 Hz de forma fiel al ser el límite teórico para una pantalla con refresco de 60 Hz). Se ha podido representar de forma fiel todas las frecuencias, incluso las frecuencias no divisoras de 60 Hz, aunque estas son las que presentan una mayor deriva de hasta 0.2 Hz respecto de la frecuencia deseada.

Con respecto a la complejidad de programar las distintas interfaces, Qt, SDL y OpenCV han sido relativamente sencillas de desarrollar, al menos para un único evento parpadeante. OpenGL ha mostrado cierta complejidad añadida.

En base a todo lo anterior, Qt ha sido una de las alternativas elegidas para desarrollar la interfaz encargada de mostrar los eventos parpadeantes a través del estimulador. SDL también ha sido una alternativa considerada debido a su sencillez de implementación y aparentemente aceptable representación de estímulos en función de su frecuencia. Sin embargo, al realizar una segunda fase de decisión entre ambas librerías comprobando un rango más amplio de frecuencias y con una mayor precisión, su desempeño final ha sido bastante pobre.

| | | | | | | | | | |
|-----------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| Frecuencia (Hz) | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| SDL | 7.153 | 8.354 | 10.01 | 10.01 | 10.01 | 12.47 | 12.47 | 12.47 | 12.47 |
| Qt | 8.068 | 9.098 | 10.01 | 11.1 | 12.19 | 13.16 | 14.31 | 15.16 | 16.14 |
| Frecuencia (Hz) | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
| SDL | 16.65 | 16.65 | 16.65 | 16.65 | 16.65 | 16.65 | 16.65 | 25.01 | 25.01 |
| Qt | 17.22 | 18.54 | 19.23 | 20.03 | 21.74 | 23.8 | 23.8 | 25.01 | 25.01 |

Tabla 4-2: Análisis más detallado de desempeño entre Qt y SDL, librerías candidatas.

Como se puede observar según la tabla anterior, Qt se adapta mejor a los distintos cambios en frecuencia mientras que SDL se adapta mucho peor, incrementando su frecuencia en base a escalones de escasa resolución. Adicionalmente, Qt cuenta con una muy buena precisión incluso para frecuencias difícilmente replicables por una pantalla con refresco a 60 Hz. En base a ello se ha optado por elegir Qt como la librería gráfica a utilizar para realizar la interfaz de estimulación al usuario.

4.4. IMPLEMENTACIÓN DE LA INTERFAZ BASADA EN QT

La interfaz gráfica estimuladora ha sido enteramente programada en C++ haciendo uso de las funciones propias de las librerías proporcionadas por Qt.

Uno de los mecanismos más importantes de Qt y del que se ha podido sacar partido en este proyecto es el uso de *slots* y *signals*. Se trata de un mecanismo que permite la comunicación entre objetos de forma que, si se conecta una señal que emite un objeto a un slot de otro objeto, este slot será llamado o activado con el parámetro de la señal en el momento preciso, lo cual es muy útil para controlar el parpadeo de las imágenes en pantalla.

El esquema general y descripción de la interfaz generada en función de los objetos involucrados es el siguiente:

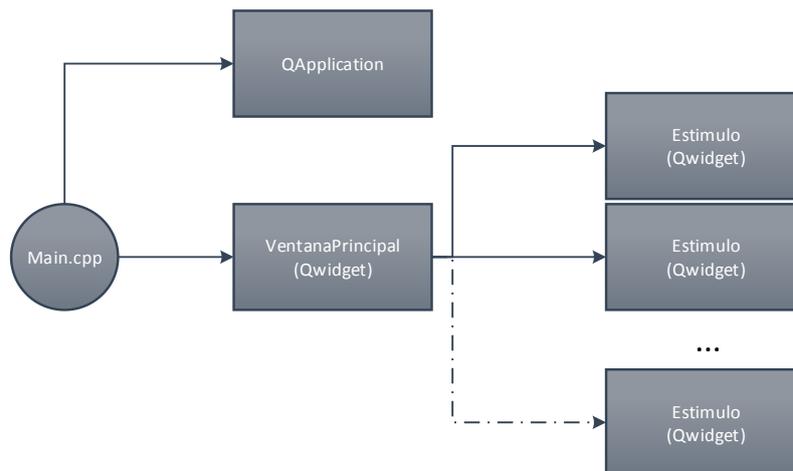


Figura 4-5: Esquema principal de ejecución de la interfaz estimuladora basada en Qt, con la posibilidad de mostrar múltiples estímulos.

QApplication es la clase principal de toda ejecución basada en QT. Sirve para controlar el flujo de la aplicación y algunas de las principales configuraciones. Contiene el bucle principal, dentro del cual todos los eventos de la ventana principal y otras fuentes son procesados y ejecutados. Controla también la inicialización y terminación de la aplicación. Independientemente del número de ventanas, estímulos mostrados... solo existirá un objeto de este tipo en toda aplicación basada en Qt. En este proyecto concreto solo es llamado su método *exec()* para comenzar con la apertura y ejecución de la interfaz.

VentanaPrincipal es un objeto creado para este proyecto dependiente de QWidget. La clase QWidget es la base para el resto de objetos de la interfaz. Pueden realizar todo tipo de tareas, desde recibir información de teclado o ratón hasta representar objetos por pantalla. Un widget que no esté embebido dentro de una clase superior será una ventana. La declaración de este objeto para el presente proyecto es como sigue:

```
// Objeto VentanaPrincipal dependiente de QWidget
class VentanaPrincipal : public QWidget
{
    Q_OBJECT

public slots:
    void timeout();

// Variables del objeto VentanaPrincipal
public:
    Estimulo *estimulo[MAX_ESTIM];           // Estímulos con los que contará la ventana
    int num_estimulos;                       // Número de estímulos totales
    QVBoxLayout layout[MAX_ESTIM];          // Layout para los QWidgets
    QTimer *timer;                          // Variable para el control del tiempo
```

```
// Métodos del objeto VentanaPrincipal
public:
    VentanaPrincipal(QWidget *parent = 0);
    void SetTipoVentana(const char *titulo, const char *fondo, int posx, int posy);
    void setNumeroEstimulos(int estimulos);
    void inicializarEstimulos(int i, int sizex, int sizey, int posx, int posy);
    void setTiempo(int tiempo);

// Método protegido para cerrar la ventana al pulsar ESC
protected:
    void keyPressEvent(QKeyEvent *event);
};
```

Como se puede apreciar, el objeto cuenta con variables (clases) que definirán los estímulos, un objeto de tipo QVBoxLayout cuyo fin será posicionar estos estímulos dentro del widget padre, que será esta ventana principal y una variable para el control del tiempo. Este control del tiempo estará conectado con el slot *timeout* de tal forma que la ventana se pueda cerrar sola automáticamente al llegar a un determinado instante de tiempo. Esta funcionalidad tiene utilidad para el entrenamiento de la interfaz, que se verá más adelante.

El Estimulo es otro objeto creado dependiente de QWidget y contenido dentro de la VentanaPrincipal. Su fin será el definir posición, tamaño y frecuencia de los iconos a mostrar en pantalla. Además, cada uno de los objetos de tipo Estimulo tendrá un sistema gestor de tiempo particular que permitirá que cada uno de los estímulos pueda tener frecuencias de parpadeo independientes. La declaración de este objeto en el código es la siguiente:

```
// Objeto Estimulo dependiente de QWidget
class Estimulo : public QWidget
{
    Q_OBJECT

// Slots que utilizaremos para el control del tiempo
public slots:
    void timeout();

// Variables públicas de la clase
public:
    QTimer *timer;           // Variable para el control del tiempo
    QLabel *label;          // Superficie donde se dibujará el estímulo
    QPixmap fondo;          // Imagen cuando el estímulo está apagado
    QPixmap imagen;         // Imagen cuando el estímulo está encendido
    int counter;             // Contador para modificar el estado del estímulo de encendido a apagado
    float frecuencia;       // Frecuencia asociada al estímulo
    int numero;              // Identificador del estímulo
    int altura;              // Tamaño vertical del estímulo
    int anchura;             // Tamaño horizontal del estímulo
    int pos_x;               // Posición del estímulo en el eje horizontal
```

```

int pos_y; // Posición del estímulo en el eje vertical
bool indicador; // Indicador de si el estímulo es un indicador de posición verde
bool indicadorMostrado; // Comprobación de si el indicador ha sido ya mostrado

// Métodos del objeto Estimulo
public:
    Estimulo(QWidget *parent = 0, int id = 0);
    void setLabel(int posx, int posy, int sizex, int sizey);
    void setImagenPrincipal(QPixmap &ima);
    void setImagenFondo(QPixmap &ima);
    void setPosicionX(int posx);
    void setPosicionY(int posy);
    void setAnchura(int width);
    void setAltura(int height);
    int freqToMilisegundos(float frecuencia);
    void setFrecuencia(float frec);
    void comenzarTimer();
};

```

Como se puede observar del código anterior, este objeto dispone de todos los parámetros necesarios para poder mostrar los estímulos por pantalla. Cada estímulo cuenta, además, con un slot que recibirá una señal cada vez que su *timer* llegue a un tiempo determinado (medio período). Cuando eso suceda se enviará una señal de ejecución hacia el *timeout*. Éste será el encargado de cambiar la imagen mostrada en pantalla entre el estímulo o bien el fondo de pantalla, de forma que se genere el parpadeo a la frecuencia deseada. Existen, además, dos tipos de objetos Estimulo: indicadores o no indicadores. Un estímulo no indicador será la señal continuamente parpadeante en pantalla. Un estímulo indicador será una imagen que solo se mostrará por pantalla en una ocasión, al comienzo de la ejecución de la interfaz. Su fin será el de señalar al usuario en la fase de entrenamiento en qué lugar de la pantalla debe fijar su atención.

La configuración de estos estímulos se realiza mediante archivos externos que es necesario pasar al programa como argumento de entrada. Existe una ligera variación entre los archivos en función de si la interfaz generada tiene que contar con un estímulo indicador o no. Un ejemplo de la forma que han de tener dichos archivos es la siguiente:

Sin estímulo indicador:

```

Numero estímulos: 4
Imágenes/Imagen.jpeg, Imágenes/Fondo.jpeg, 13, 100, 100, 175, 175
Imágenes/Imagen.jpeg, Imágenes/Fondo.jpeg, 10, 675, 100, 175, 175
Imágenes/Imagen.jpeg, Imágenes/Fondo.jpeg, 19, 100, 475, 175, 175
Imágenes/Imagen.jpeg, Imágenes/Fondo.jpeg, 8, 675, 475, 175, 175

```

Con estímulo indicador:

```

Indicador posicion: 742, 655
Numero estímulos: 4
Imágenes/Imagen.jpeg, Imágenes/Fondo.jpeg, 13, 100, 100, 175, 175
Imágenes/Imagen.jpeg, Imágenes/Fondo.jpeg, 10, 675, 100, 175, 175
Imágenes/Imagen.jpeg, Imágenes/Fondo.jpeg, 19, 100, 475, 175, 175
Imágenes/Imagen.jpeg, Imágenes/Fondo.jpeg, 8, 675, 475, 175, 175

```

Como se puede observar, la única diferencia es la primera línea del archivo de configuración, el cual en caso de necesitar estímulo de configuración habrá que indicar su posición. Esta posición vendrá dada por la distancia en píxeles desde la esquina superior izquierda de la pantalla. La configuración para cada uno de los estímulos necesita además conocer la imagen que queremos representar (primer argumento), la imagen de fondo de pantalla (segundo argumento), su frecuencia (tercer argumento) y su tamaño en horizontal y vertical (últimos dos argumentos).

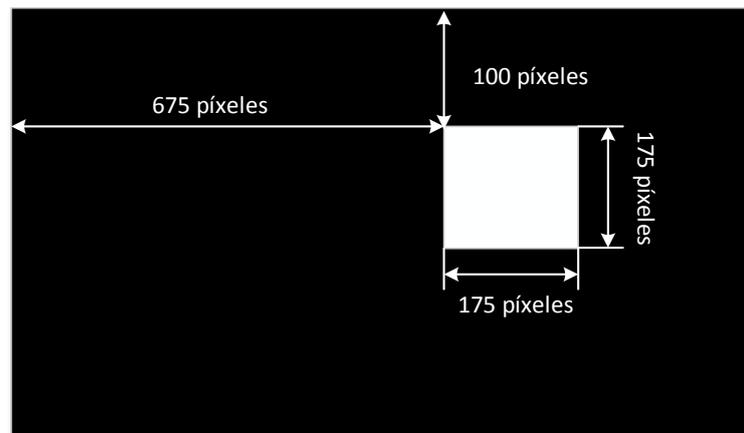


Figura 4-6: Ejemplo de cómo quedaría el segundo estímulo posicionado en pantalla según las indicaciones proporcionadas en el código de ejemplo para casos sin indicador.

4.5. CONFIGURACIÓN DEL ANÁLISIS DE SEÑAL

En el Apartado 4.1 se comentó la necesidad de realizar un primer análisis de señal con el fin de verificar la viabilidad del casco para la detección de SSVEPs. Dicho análisis tenía el fin fundamental de representar el espectro de las señales de la forma más fiel posible sin entrar en consideraciones acerca de la velocidad de análisis y la disponibilidad de información en tiempo real.

La necesidad de realizar el análisis de forma online de forma rápida y sin tamaños de ventana muy grandes serán las condiciones necesarias para la implementación final en el código.

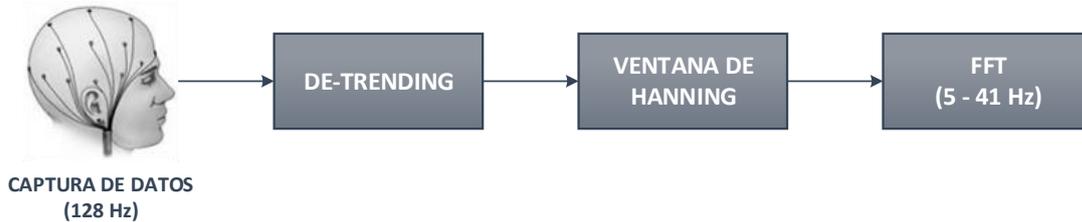


Figura 4-7: Esquema del análisis de señal implementado.

Para que el sistema funcione sin una gran latencia se ha optado por implementar ventanas de dos segundos de duración, con un solape del 50%, eso es, tomando un segundo de la ventana anterior. El desplazamiento entre ventanas será de un segundo, por lo que el programa será capaz de arrojar un resultado del análisis por cada segundo. La primera fase del análisis de señal será la eliminación de tendencias, o *detrending*. Esto permite que las constantes o tendencias lineales no aparezcan como ruido extra de baja frecuencia. Además, al enventanar una señal discreta que presente componentes lineales de la que se pretende obtener su espectro en frecuencia producirá una discontinuidad entre el primer valor de la ventana y el último valor de la misma, que al hacer la FFT (algoritmo para implementar la Transformada de Fourier Discreta o DFT) se transformarán en nuevas componentes frecuenciales que no estaban en la señal originalmente. Al eliminar las tendencias lineales se minimiza la generación de estas nuevas componentes no deseadas fruto de la discontinuidad. Posteriormente se aplicará la ventana de Hanning al conjunto de muestras; nos ayudará a obtener una buena resolución en frecuencia debido a su suficientemente estrecho lóbulo principal con una importancia no muy elevada de sus lóbulos laterales. Hanning es uno de los tipos de ventanas más utilizados para análisis de señal de estas características. Posteriormente solo quedará realizar la DFT, implementada a partir de librerías matemáticas que realicen el algoritmo FFT.

Dadas las pruebas realizadas cuando se implementó el análisis offline mediante el MATLAB, se estimó que el rango útil de frecuencias para este proyecto se podría situar entre los 6 Hz (mínima frecuencia a la que se manifiestan los SSVEP) y los 20 Hz, empleando para ello la suma de la frecuencia fundamental de estimulación con sus armónicos. Por la importancia de éstos y de los adyacentes el rango de frecuencias que se considerará será entre los 5 Hz y los 41 Hz, lo suficiente para abarcar el rango de los 6 a los 20 Hz, sus armónicos y los adyacentes de todos ellos.

4.6. IMPLEMENTACIÓN ONLINE DEL ANÁLISIS DE SEÑAL

En el momento en que se va a realizar un análisis de señal a través del casco Emotiv EPOC, la aplicación lanzará un *fork* que dividirá los procesos de captura de datos y de análisis, de forma que actúen en paralelo y comunicándose como se describió en el apartado 4.2. La lógica que sigue el proceso de análisis se detalla en el siguiente esquema:

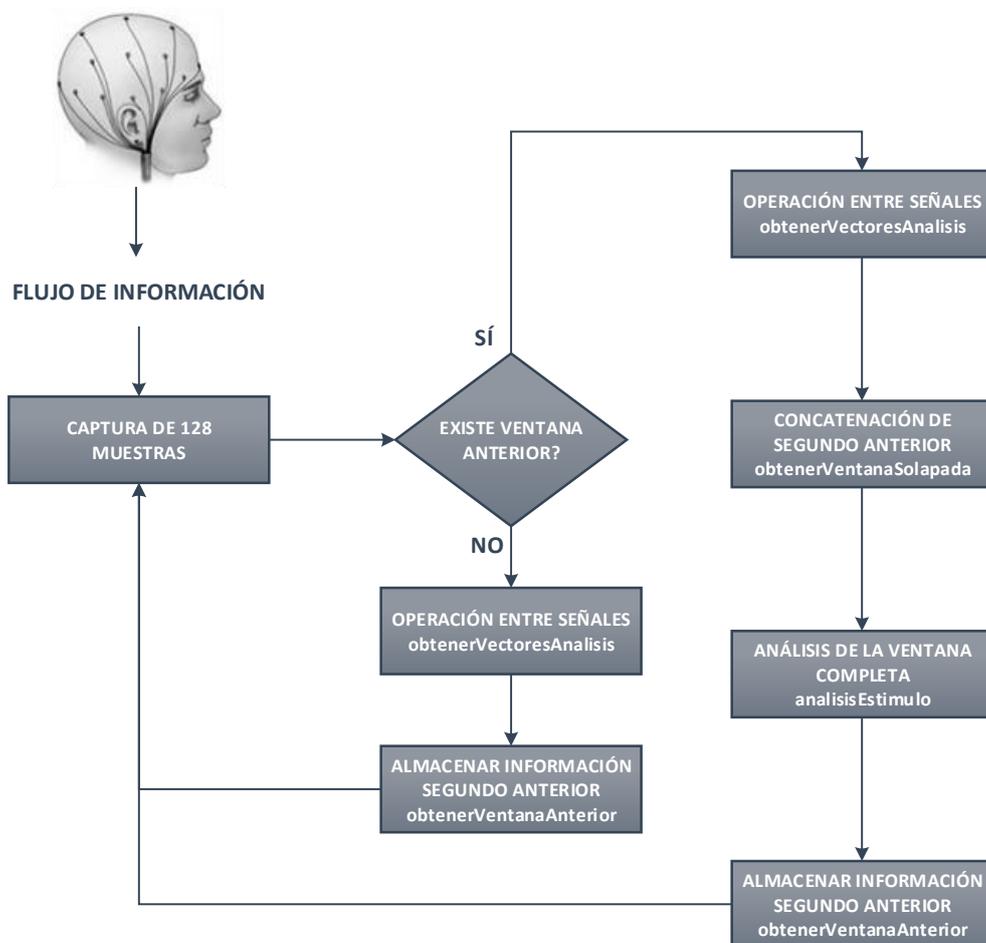


Figura 4-8: Flujo de la rutina encargada de realizar el análisis de señal.

La información que se obtiene del flujo de datos procedente del casco es guardada en una matriz de dimensión 128x22, para almacenar las muestras que generan todos los canales a lo largo de un segundo. La primera fase que se realizará invariablemente será operar entre los distintos canales deseados para obtener la información útil. En este caso la operación que se realiza es $(O1+O2)-(F3+F4)$. La función encargada de ello es `obtenerVectoresAnálisis` (la información de la distribución de estas funciones en los distintos archivos del programa se encuentra en el Apéndice I):

```

/**
 * @brief obtenerVectoresAnalisis: Función encargada de operar entre los distintos canales de grabación del
 * casco para analizar las señales deseadas
 * @param canalesDeseados: Array resultado de la operación entre los distintos canales, por referencia
 * @param matrizDatos: Matriz con los datos de todos los canales obtenidos del casco
 */
void obtenerVectoresAnalisis(double *canalesDeseados, double matrizDatos[MUESTREO][NUM_CANALES])

```

Obtenida la información de los sensores con la que queremos trabajar procedemos a intentar analizarla. Si se trata de las 128 primeras muestras, esto es, el primer segundo de ejecución, no tendremos muestras suficientes para componer una ventana completa de 256 muestras. Por tanto, se procede a almacenar la información hasta que se pueda hacer uso de ella una iteración más tarde. La función encargada de almacenar estas ventanas para su posterior uso es `obtenerVentanaAnterior`:

```

/**
 * @brief obtenerVentanaAnterior: Función encargada de almacenar en la matriz general de datos la
 * ventana anterior para el análisis con solapamiento
 * @param datos: Estructura conetenedora de los datos del análisis
 * @param vectorDatos: Vector con los datos de la ventana anterior
 */
void obtenerVentanaAnterior(datosAnalisis *datos, double vectorDatos[MUESTREO])

```

En la siguiente iteración deberíamos disponer ya de las 256 muestras fruto de los dos segundos de toma de datos. Todas estas muestras constituyen una ventana completa. La primera fase será, de nuevo, operar entre los distintos canales para obtener los resultados de esta nueva iteración llamando, de nuevo, a `obtenerVectoresAnalisis`. El siguiente paso será concatenar la información de esta nueva iteración con la información que guardamos previamente mediante `obtenerVentanaAnterior`. La función encargada de realizar esta tarea es `obtenerVentanaSolapada`:

```

/**
 * @brief obtenerVentanaSolapada: Función encargada de juntar en un mismo array la ventana actual y la
 * parte solapada de la ventana anterior
 * @param datos: Estructura conetenedora de los datos del análisis
 * @param vectorDatos: Vector con los datos de la ventana actual
 */
void obtenerVentanaSolapada(datosAnalisis *datos, double vectorDatos[MUESTREO])

```

Obtenida la ventana completa se de proceder a su análisis como se ha detallado en la figura 4.7. La función encargada de orquestar el análisis y la posterior detección de la

máxima puntuación alcanzada es `analisisEstimulo`. No obstante no es la única función de este tipo implementada en el código; en función de si estamos analizando un *baseline* o si estamos entrenando a la interfaz existen otras funciones dedicadas a ese efecto con pequeñas diferencias entre sí a nivel de código, especialmente en lo referente a cómo tratar la información de salida. En el caso en concreto de que se trate de un análisis de un estímulo normal, la información de salida se muestra por consola y se escribe en un archivo que es pasado a la función:

```
/**
 * @brief analisisEstimulo: Llamada cada ventana de análisis para obtener el procesamiento de señal (con
 * estimulación) asociado a cada ventana de tiempo
 * @param procesado: Estructura conetenedora de los datos del análisis.
 * @param archivoSalida: Archivo en el que se guardarán los resultados obtenidos para facilitar un análisis
 * offline de los mismos.
 */
void analisisEstimulo(datosAnalisis* procesado,FILE *archivoSalida)
```

Una vez que se ha realizado el análisis de la ventana completa el único paso que queda es el de almacenar la información de las 128 muestras que se han obtenido en la última iteración y almacenarlas, de forma que puedan ser usadas por la siguiente iteración, llevando así a cabo el solape entre las distintas ventanas. Para realizar esta operación se volverá a utilizar la función `obtenerVentanaAnterior`.

El análisis de cada ventana implica funciones específicas para realizar la eliminación de las características lineales, realizar el inventariado y obtener su espectro en frecuencia. Para la obtención de este espectro será necesario realizar la DFT, para lo cual se hará uso de la librería GSL (*GNU Scientific Library*). La función que se utilizará para ello es la siguiente:

```
/**
 * Función encargada de realizar la FFT (algoritmo de DFT)
 * @param data: Vector de datos a analizar, longitud 256
 * @param stride: Resolución de los datos de salida. Con stride=1 los resultados se obtienen Hz a Hz
 * @param n: Longitud de la matriz de salida, típicamente de forma 2^x. En este caso n = 256
 */
int gsl_fft_real_radix2_transform (double data[], size_t stride, size_t n)
```

Esta función permite calcular el espectro en frecuencia de un conjunto real de datos. La salida es una secuencia con valores complejos y reales organizada del siguiente modo: si $k < n/2$ la parte real del valor k -ésimo se almacena en la posición k , mientras que la parte imaginaria se almacena en la posición $n-k$. Para los términos en los que $k > n/2$ pueden ser reconstruidos por simetría. Los términos para $k=0$ y $k=n/2$ constituyen un caso especial, ya que ambos serán puramente reales. En este caso, sus valores reales

están almacenados en la posición 0 y $n/2$ respectivamente, mientras que su parte imaginaria al ser igual a cero no se almacena. La siguiente tabla obtenida del manual de las librerías GSL muestra la correspondencia en los datos de salida:

```

complex[0].real = data[0]
complex[0].imag = 0
complex[1].real = data[1]
complex[1].imag = data[n-1]
.....
complex[k].real = data[k]
complex[k].imag = data[n-k]
.....
complex[n/2].real = data[n/2]
complex[n/2].imag = 0
.....
complex[k'].real = data[k]      k' = n - k
complex[k'].imag = -data[n-k]
.....
complex[n-1].real = data[1]
complex[n-1].imag = -data[n-1]

```

Para poder acceder a estos datos de forma correcta tras la llamada a la función de la FFT se ha utilizado el siguiente procedimiento:

```

// Definición de cómo acceder a valores reales e imaginarios del vector devuelto por
gsl_fft_real_radix2_transform
#define REAL(z,i) ((z)[i])
#define IMAG(z,i) ((z)[(MUESTREO+VENTANA)-(i)]) // 256 - (i)

```

Con lo anterior, a partir de REAL e IMAG se podrá acceder a las posiciones de una frecuencia determinada obtenida mediante la función anterior.

Cabe destacar que con la implementación que se ha llevado a cabo, tanto para el uso del casco lector de EEG a través del SDK como en el análisis de señal se ha obtenido un excelente rendimiento, no acumulando el sistema ningún tipo de retardo a la hora de realizar el análisis de señal y mostrando los resultados por consola en tiempo real, incluso con una carga de trabajo relativamente elevada por parte del hardware.

Es importante señalar que el proceso de selección de resultados conoce en todo momento las frecuencias que se están estimulando. Con ello se limita la probabilidad de falsos positivos con frecuencias que ni siquiera están siendo estimuladas. En un primer diseño del programa, el análisis de señal no conocía las frecuencias con las que se estaban realizando la estimulación. En estas circunstancias el comportamiento del casco y los resultados obtenidos por el análisis eran muy poco alentadores debido en parte a la escasa relación señal a ruido obtenido al no tener electrodos emplazados en la

posición idónea para la captura de SSVEP y, además, debido a que los electrodos utilizados para limitar el ruido proveniente de otras partes del cerebro no reducen de forma significativa estas interferencias, tan solo una parte pequeña de ellas.

4.7. SELECCIÓN DE FRECUENCIA RESULTADO

Después de obtener el espectro de la ventana a analizar se procederá a detectar la frecuencia que cuenta con una mayor probabilidad de ser la estimulada. El análisis en esta parte va un poco más allá de observar en qué lugar se observa la mayor amplitud del espectro; la importancia de los adyacentes debido a la no estacionaridad de las señales del cerebro y que estas pueden presentar deriva en frecuencia y fase y la consideración de los armónicos son parte fundamental en este análisis. Esto es especialmente relevante cuando las frecuencias comienzan a ser elevadas, donde la importancia de los armónicos es muy evidente o bien cuando se está estimulando frecuencias muy poco distantes unas de otras, en donde el análisis de adyacentes puede ser contraproducente. Cabe destacar, además, que la amplitud de las señales generadas a cada frecuencia de estimulación, así como la de sus armónicos, adyacentes y la deriva en frecuencia que se puede producir en estas señales es dependiente de cada usuario; distintos usuarios reaccionarán de forma distinta a un mismo estímulo e incluso una misma persona generará señales diferentes respecto un mismo estímulo a lo largo del tiempo. Adicionalmente, habrá estímulos cuyas señales evocadas generen picos más distintivos que otras en las frecuencias deseadas o estímulos que al ser representados simultáneamente en pantalla generen interferencias unos con otros. Por todo ello no se debe perder de vista siempre la importancia de la adaptabilidad del sistema de cara al usuario y el cuidado en el análisis de señal. En este caso, una vez obtenido el espectro en frecuencia de la señal se pondrá especial cuidado en considerar las derivas frecuenciales que se producirán en las señales evocadas así como la importancia de los armónicos.

La expresión correspondiente a la puntuación que se está utilizando es la siguiente:

$$P_f = X(f) + X(2f) + \frac{1}{2} [X(f - 1) + X(f + 1) + X(2f - 1) + X(2f + 1)]$$

Según la expresión anterior, estamos considerando para el cálculo de la puntuación los valores de la transformada en la frecuencia fundamental y en el primer armónico. Para el cálculo de los adyacentes estamos limitando su peso en la puntuación final a la mitad, de forma que se tienen en cuenta pero están ligeramente penalizados. El objetivo de esto es lograr una mayor capacidad de discriminar entre distintas frecuencias muy próximas entre sí. De no ser así es muy posible que dos frecuencias adyacentes seleccionables por el usuario obtuvieran una puntuación prácticamente idéntica. Además, en la anterior suma están involucrados 6 factores, siendo el rango de

frecuencias útiles desde los 5 Hz a los 41 Hz, esto es, 36 valores. Esto implica que para el cálculo de cada frecuencia se tendría en cuenta el 16.7% de los valores totales obtenidos de tener todos el mismo peso en la ecuación anterior, lo cual sería excesivo para el cálculo de un único resultado y podría llevar al escenario comentado en el cual distintas frecuencias obtuviesen valores similares.

Como se mencionó en el apartado anterior, el sistema conoce en todo momento las frecuencias que están siendo estimuladas. Las ventajas en el caso del análisis de señal eran que ello limitaba mucho la probabilidad de que se produjeran falsas detecciones con frecuencias con las que ni siquiera se estaba realizando estimulación. Otra ventaja, también bastante relevante al tratarse de un diseño que tiene que trabajar en tiempo real, es la reducción de cálculos a realizar por parte del sistema; se están considerando únicamente 4 frecuencias como posibles soluciones de entre las 15 totales alternativas que en principio habría que analizar (todas las frecuencias desde los 6Hz hasta los 20Hz, ambas incluidas). Se está reduciendo así el número de cálculos a realizar para obtener el resultado de cada ventana por la cuarta parte.

Si bien al categorizar los posibles resultados únicamente entre las frecuencias con las que se está estimulando al usuario se ha mejorado significativamente la fiabilidad y optimización del sistema, existen también algunas desventajas. La principal es la imposibilidad de detectar la situación en la que el usuario no esté mirando a ninguno de los estímulos de la pantalla y la interfaz sea capaz de darse cuenta de ello. Se ha pensado en la posibilidad de asignar un umbral mínimo para que la puntuación final que obtiene cada frecuencia sea válida como resultado. Sin embargo, esta solución para un valor del umbral no muy alto proporcionaba un altísimo número de no detecciones mientras el usuario estaba mirando algún estímulo en pantalla. Bajando el umbral por debajo de estos límites el sistema dejaba de ser capaz de distinguir cuando el usuario estaba mirando a la pantalla y cuando no estaba siendo estimulado. Por ello, se ha optado por no hacer uso de este umbral ya que los beneficios que aportaba se veían lastrados por una gran bajada de fiabilidad del sistema cuando el usuario se encuentra mirando a alguno de los estímulos.

4.8. ALGORITMO DE SELECCIÓN DE FRECUENCIAS

Debido al distinto comportamiento entre los diferentes usuarios es necesario contar con un sistema de selección de frecuencias que sea flexible y pueda adaptarse a cada uno de ellos. Un mismo usuario, además, puede ser sensible de distinta forma a los estímulos a lo largo del tiempo debido a la no estacionaridad del cerebro. Para adaptarse a estos cambios será necesario desarrollar un sistema que se comuniquen con el usuario, de tal forma que al final de un proceso de entrenamiento se pueda realizar la estimulación de la forma más óptima posible. Este entrenamiento tiene que ser,

además, llevadero y no implicar una gran inversión en tiempo para que pueda realizarse siempre que sea necesario y las circunstancias del usuario cambien hasta el punto de que la configuración anterior del sistema deje de ser fiable.

En general, la tendencia para adaptarse al usuario en la estimulación de estos potenciales se basa en la búsqueda de las frecuencias *top*; esto consiste en realizar un barrido por todas las frecuencias de estimulación posibles y utilizar aquellas en las que se observe una buena respuesta por parte del usuario, esto es, la generación de potenciales evocados distintivos. Sin embargo, este método no tiene en cuenta que puede haber incompatibilidades en el uso simultáneo de varias de estas frecuencias, aunque por separado hayan sido capaces de generar SSVEPs de gran intensidad. Frecuencias muy próximas entre sí puede confundir al sistema que pretende analizar que estímulo está observando el usuario, más si tenemos en cuenta las derivas frecuenciales que pueden producirse en el cerebro al generar los estímulos. Se pueden dar otras circunstancias como que, por ejemplo, un estímulo tenga una frecuencia determinada que distraiga al usuario de tal forma que sea incapaz de generar SSVEPs adecuadamente. Por ello se ha decidido hacer uso de un algoritmo que haga uso de las mejores frecuencias de estimulación para cada usuario pero teniendo en cuenta a la vez como el usuario se comporta cuando hay varios estímulos en pantalla. El algoritmo de selección de frecuencias que se utilizará en el presente proyecto se basará en el protocolo de optimización ACL (*Assisted Closed Loop*) desarrollado por el Grupo de Neurocomputación Biológica (Fernandez-Vargas et al. 2013), con una serie de cambios para adaptarse a la utilización del Emotiv EPOC.

Primera fase: Exploración de frecuencias.

La primera fase del algoritmo consistirá en saber qué frecuencias generan los mejores estímulos para el usuario. Para ello se realizará un barrido de todas las frecuencias utilizables desde los 6Hz a los 20Hz estimulando al usuario. Aquellas frecuencias que obtengan una alta puntuación al analizar los estímulos SSVEP generados y superen cierto umbral serán consideradas como frecuencias utilizables. El umbral se ha elegido basándose en la experimentación y en el análisis de datos obtenidos tras unas simulaciones de prueba, y se ha fijado en un valor de 7. Una vez obtenidas las n frecuencias válidas que generan estímulos evocados, éstas se ordenarán de mayor puntuación a menor puntuación y se les asignará una nueva puntuación en función de su posición: la frecuencia con mayor valor se le asignará puntuación n , a la siguiente $n-1$ y así sucesivamente.

Segunda fase: Búsqueda de los mejores pares.

Obtenidas las frecuencias con las que el usuario reacciona de forma favorable, el siguiente paso es evaluar con qué pares de frecuencias mostradas simultáneamente en pantalla el conjunto usuario-sistema se comporta de forma más fiable. Para ello,

primeramente, habrá que definir un valor de compatibilidad entre distintas frecuencias, cuya expresión es la que sigue:

$$C_{xy}(t) = \alpha (S_x(t) + S_y(t)) + \beta d_{xy}$$

En la ecuación anterior los valores de α y β son 1.5 y 1, respectivamente. El instante t se refiere al número de la iteración para un sujeto determinado. Dicho valor ha sido obtenido a partir de las pruebas realizadas anteriormente por el Grupo de Neurocomputación Biológica. La distancia d_{xy} es, en esencia, la diferencia entre las distintas frecuencias involucradas en la estimulación. Para esta fase su expresión es:

$$d_{xy} = |f_x - f_y|$$

Los valores S_x y S_y son las puntuaciones asignadas a cada una de las frecuencias involucradas y se irán calculando para cada par de frecuencias en función del éxito que se haya obtenido al estimular con ellas al usuario. Estas puntuaciones han sido inicializadas en la primera fase de cara a obtener las primeras compatibilidades y comenzar con la estimulación de pares de frecuencias pero son valores que se deben actualizar a medida que se vaya estimulando al usuario con dos eventos simultáneos. En esta fase en la que existen dos estímulos distintos se pedirá al usuario que se vaya fijando en cada uno de ellos siguiendo una secuencia a lo largo del tiempo. A partir de ello se obtendrán dos parámetros:

- **Tasa de aciertos (SR – Success Rate):** Número de veces en que el sistema es capaz de conocer el estímulo que está siendo observado por el usuario respecto al número total de posibles detecciones.
- **Duración de la detección (T):** Tiempo en segundos en que las detecciones han sido correctas.

Con estos parámetros se obtienen las nuevas puntuaciones de acuerdo a la siguiente fórmula:

$$S_x(t) = S_x(t - 1)(\delta \cdot SR - \gamma \cdot T)$$

Donde el valor de δ y γ serán igual a 3 y 0.02, respectivamente, teniendo un gran peso en la actualización de la fórmula la tasa de aciertos.

En esta segunda fase los pares de frecuencias se obtienen entre aquellas frecuencias que presentan una mejor compatibilidad. Adicionalmente, de cara a evitar el riesgo de que solo se estimulen dos pares de frecuencias durante todas las iteraciones por que presenten una muy buena compatibilidad tras la estimulación se ha implementado un

sistema que impide que dos frecuencias vuelvan a ser usadas para la estimulación si todavía quedan otras frecuencias elegibles sin participar en esta fase. El número total de iteraciones que se llevará a cabo será $3n/4$, siendo n el número de frecuencias que han sido elegidas tras la primera fase. Con todas estas iteraciones será suficiente para evaluar todos los pares de frecuencias de interés.

Tercera fase: Búsqueda de los mejores pares de pares.

En esta última fase se mostrarán los cuatro estímulos por pantalla. Primeramente se crearán pares de frecuencias según los resultados alcanzados en la parte anterior, de forma que los pares de frecuencias con mejor compatibilidad sean agrupados y se consideren de aquí en adelante como un conjunto indivisible. Esto reducirá el número de sets a $n/2$. La idea será representar por pantalla pares de sets que constituirán los 4 estímulos evocados de cara a evaluar la máxima compatibilidad entre todos los pares de sets de dos frecuencias. De esta forma, primeramente se habrán creado sets de dos frecuencias con la mejor compatibilidad entre ellos para, posteriormente, evaluar la compatibilidad entre todos los sets creados con el objetivo de encontrar las 4 mejores frecuencias de estimulación para el usuario concreto. La fase de estimulación y las fórmulas utilizadas serán las mismas que en la segunda fase, con diferencia del cálculo de distancia que, en este caso, involucrará cuatro frecuencias:

$$d_{xy} = \sum_{i=1}^4 \sum_{j=1}^4 |f_i - f_j|$$

Nuevamente, cuando se hayan realizado $3m/4$ estimulaciones, siendo $m=n/2$ al estar utilizando conjuntos de dos frecuencias y haber dividido, por tanto, entre dos el número de alternativas utilizables, terminará la fase y se elegirán dos sets de frecuencias como las frecuencias resultado del análisis. Se han explorado dos métodos para obtener estos dos sets de frecuencias:

- **Criterio de compatibilidad:** Al igual que en las iteraciones anteriores del algoritmo, las frecuencias elegidas serán aquellas que cumplan el criterio de máxima compatibilidad. Esto implica que las cuatro frecuencias elegidas serán las frecuencias que forman los dos pares de frecuencias que presentan una mejor compatibilidad entre sí dadas las fórmulas expresadas anteriormente.
- **Criterio de máxima puntuación:** Según este criterio, las cuatro frecuencias finales elegidas en la última iteración se obtendrán en función de los dos pares de frecuencias que presentan una mayor puntuación.

El criterio de compatibilidad ha sido utilizado anteriormente cuando se ha implementado este algoritmo (Fernandez-Vargas et al. 2013). Sin embargo, en estas ocasiones el sistema no conocía de antemano la frecuencia de las señales que estaban siendo estimuladas, teniendo una mayor importancia a la hora de realizar el algoritmo el concepto de distancia entre frecuencias. Frecuencias próximas entre sí pueden derivar en la aparición de picos en frecuencias intermedias, sus armónicos o, en otras frecuencias que no forman parte de la estimulación. En esta ocasión, como el sistema conoce siempre que tipos de estímulos están siendo utilizados y solo sus frecuencias involucradas serán analizadas, la distancia entre frecuencias reduce su importancia en favor de la tasa de aciertos y la duración de detección. Sigue teniendo peso en el algoritmo en iteraciones anteriores por que se ha observado que utilizando pares de frecuencias con escasa distancia entre sí la efectividad se reduce apreciablemente. Sin embargo, se ha considerado que una vez que ha sido evaluada esta compatibilidad en iteraciones anteriores, en la elección final de frecuencias tiene un mayor peso la tasa de aciertos, teniendo por tanto sentido en este escenario el uso de criterio de máxima puntuación.

4.9. SELECCIÓN DE FRECUENCIAS PREFIJADAS

Si bien es interesante el uso de un algoritmo que se adapte al usuario, también puede ser de utilidad analizar un conjunto de frecuencias comunes para todos los usuarios siguiendo una serie de criterios generales y dejando de lado las particularidades propias de cada persona. Este set de frecuencias será elegido en función de las siguientes reglas, con el objetivo de proporcionar señales evocadas lo más distintivas posibles:

- Evitar el uso de frecuencias múltiplos de otras frecuencias.
- Uso de frecuencias fácilmente representables por el estimulador.
- Distancia entre frecuencias superior a 1 Hz.

De cara a satisfacer los tres criterios de forma simultánea de la mejor forma posible, las frecuencias elegidas son 6 Hz, 9 Hz, 15 Hz y 20 Hz. Todas estas frecuencias presentan un excelente comportamiento al ser mostradas por pantalla con un error máximo de 0.16Hz al representar los 15 Hz mediante Qt. La distancia entre ellas es lo suficientemente elevadas para que no haya problemas al compartir valores adyacentes y, además, en el conjunto de frecuencias seleccionadas ninguna frecuencia es armónico de otra.

5. EXPERIMENTACIÓN

5.1. ESTRUCTURA DE LAS PRUEBAS

Para comprobar la validez de la interfaz se probará su desempeño con distintos voluntarios dentro del laboratorio del Grupo de Neurocomputación Biológica, en un equipo único destinado a realizar estas pruebas, de cara a que las condiciones ambientales no varíen entre sujetos. Se trata de un ordenador con un procesador Intel Core i7 CPU 950 de a 3.07 GHz con 4 núcleos reales e *hyperthreading*, 8GB de memoria RAM y pantalla Ancor Communications Inc de 27 pulgadas, a una resolución de 1920x1080 y 60Hz de refresco de pantalla. El sistema operativo es un Ubuntu 12.04 LTS de 32 bits y la tarjeta gráfica es una NVIDIA GeForce GT 220.

El experimento está dividido en una serie de fases que se detallan a continuación:

Fase 1: Grabación del baseline

Para hacer el resto de análisis es necesario tener una grabación de la actividad de fondo del cerebro de los usuarios que sirva como referencia. Todos los sujetos tendrán que someterse a una grabación de dicha actividad. Para ello deberán observar 60 segundos a la pantalla del monitor en negro, esto es, sin ningún estímulo visual. Esto permitirá conocer de antemano la actividad del cerebro de cara a poder distinguir los potenciales estimulados con mayor facilidad. Se pedirá a los sujetos que eviten parpadear en la medida de lo posible y que limiten los movimientos musculares, especialmente los faciales ya que introducirán un gran nivel de ruido en las grabaciones. Esto incluye también evitar hablar, realizar gestos, muecas...

Fase 2: Entrenamiento de la interfaz

Después de obtener el *baseline* se comenzará a estimular al sujeto según el algoritmo ACL. La primera fase del mismo permitirá conocer si el individuo que esté realizando el experimento es capaz de generar SSVEPs y si, en caso de hacerlo, es sensible a un número de frecuencias lo suficientemente elevado como para poder controlar los cuatro grados de libertad de los que contará la interfaz. Posteriormente, se buscarán las mejores cuatro frecuencias para el usuario concreto. En esta fase, por tanto, obtenemos:

- Frecuencias top.
- Frecuencias dado el algoritmo con criterio de compatibilidad.
- Frecuencias dado el algoritmo con criterio de máxima puntuación.

La primera fase del algoritmo consiste en mostrar un único evento parpadeante por pantalla, cuya frecuencia varía desde los 6Hz a los 20Hz en pasos de 1Hz de forma ascendente. Para cada frecuencia se mostrará un icono cuadrado blanco de 200x200 píxeles encargado de realizar la estimulación gráfica, apareciendo y desapareciendo de la pantalla a la frecuencia deseada. El usuario deberá mirar fijamente dicho icono durante 10 segundos evitando parpadear y moverse. Pasados esos 10 segundos el usuario contará con 2 segundos de descanso para parpadear y prepararse para el siguiente icono a una frecuencia de 1Hz mayor. Adicionalmente debajo del icono a observar aparecerá un indicativo de color verde que señalará la posición en la que el usuario debe posar su mirada. Dicho indicador verde solo estará visible durante el primer segundo de la estimulación; después desaparecerá para no interferir en la atención del usuario al icono blanco parpadeante. En este primer barrido de frecuencias la presencia de este indicador no es necesaria dado que solo existe un único evento en la pantalla. Aun así, de cara a acostumbrar al usuario a siguientes fases del algoritmo en las que dicho indicativo es necesario, se ha decidido mostrar el icono en esta fase inicial.

Obtenidas las frecuencias de estimulación útiles para el usuario, se procederá a la ejecución de las siguientes fases del algoritmo ACL como se ha especificado en apartados anteriores. En este caso, cuando se muestren varios eventos parpadeantes por pantalla, el indicador verde será el encargado de instruir al usuario hacia qué parte de la pantalla mirar en cada momento.

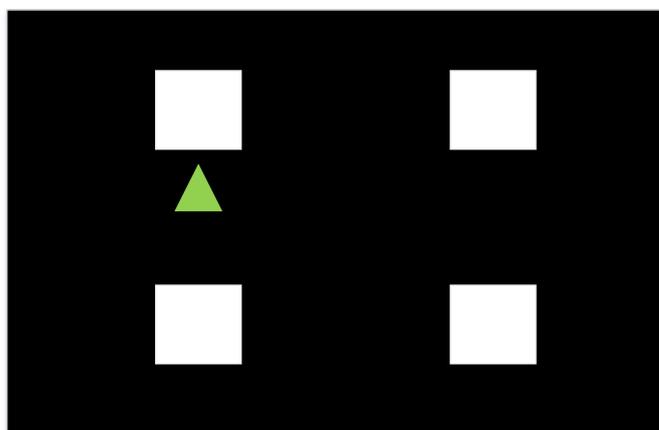


Figura 5-1: Representación de la última fase del algoritmo ACL, con los cuatro estímulos en pantalla y el indicador verde.

Para cada par de frecuencias elegidas por el algoritmo en cada iteración el usuario tendrá que mirar dos veces a cada uno de los iconos que parpadean, siguiendo las indicaciones de la flecha indicativa verde. El proceso es muy similar al de la fase anterior: deberá mirar al icono que señala el indicador durante 10 segundos sin parpadear, seguidos de otros 2 segundos de descanso antes de volver a fijar la mirada en el icono que indique nuevamente el marcador. Una vez que haya mirado dos veces a cada uno

de los iconos (4 observaciones por cada par de frecuencias, siguiendo las indicaciones del marcador verde), el sistema dará un descanso de 10 segundos al usuario antes de comenzar con el análisis de la siguiente iteración con el nuevo set de frecuencias.

Si se trata de la última fase del algoritmo en la cual se muestran 4 estímulos parpadeantes por pantalla, el procedimiento será similar a lo mencionado anteriormente con la salvedad de que, al realizar el indicador verde un barrido por todos los iconos estimulantes dos veces, cada set de frecuencias necesitará de 8 observaciones por parte del usuario, dos por cada uno de los iconos que aparecen en pantalla.

Fase 3: Fase de estimulación y experimentación

En esta fase se comprobará la eficacia de la interfaz una vez ha sido entrenada para el usuario concreto. Para ello, se estimulará al usuario con cuatro objetos parpadeantes en pantalla simultáneamente y se le pedirá que observe a cada uno de ellos durante 15 segundos. Como el análisis de señal de la interfaz permite obtener un resultado por segundo, se evaluará la fiabilidad como la medida de predicciones correctas entre el total de predicciones posibles. Se evaluarán cuatro sets de frecuencias de cuatro estímulos cada una, atendiendo a los siguientes criterios:

- **Frecuencias prefijadas:** Grupo de frecuencias de control que será idéntico para todos los usuarios.
- **Frecuencias top:** En este caso los cuatro estímulos oscilarán a las frecuencias a las que el usuario haya sido más sensible en la primera fase del entrenamiento, sin atender a ningún otro criterio.
- **Algoritmo con criterio de compatibilidad:** Set de frecuencias obtenidas al finalizar el algoritmo, eligiendo en la última iteración los pares de sets de frecuencias que presenten mejor compatibilidad entre ellos.
- **Algoritmo con criterio de puntuación:** Set de frecuencias obtenidas al finalizar el algoritmo, eligiendo en la última iteración los pares de sets de frecuencias que hayan obtenido una mayor puntuación, esto es, una mayor tasa de éxitos.

La duración de las pruebas para cada usuario se estima que será siempre inferior a una hora. La parte de entrenamiento mediante el algoritmo ACL durará típicamente entre unos 25-30 minutos considerando los tiempos de descanso necesarios. Hay que tener en cuenta que la duración del algoritmo depende del número de frecuencias a las que los usuarios sean sensibles, por lo tanto la duración es estimada. Los tiempos de descanso durante el entrenamiento también varían en función de las personas, dado que es importante que los usuarios sean capaces de mantener cierta capacidad de concentración durante todo el proceso de entrenamiento. Atendiendo a las necesidades particulares de la persona que esté realizando el experimento, en ocasiones habrá que prolongar los tiempos de descanso para que la experiencia no sea incómoda y ello se traduzca en un entrenamiento ineficaz. La parte de experimentación no deberá prolongarse más allá de 10 minutos en total.

5.2. PREVISIÓN DE RESULTADOS DE LAS PRUEBAS

Una vez que se ha grabado el *baseline* y se ha realizado el entrenamiento de la interfaz para un usuario concreto se obtienen los cuatro sets de soluciones. Se espera que el uso de frecuencias prefijadas obtenga, en general, los peores resultados a la hora de evaluar la efectividad de la interfaz. Esto es debido a que la casuística propia de cada usuario se obvia. Pueden darse casos puntuales en los que usuarios determinados obtengan un buen rendimiento con la interfaz configurada con esta solución debido a que sean justamente estimulables a las frecuencias prefijadas. Sin embargo hay que tener presente que, pese a que esta situación pueda darse, debe ser de forma esporádica y estadísticamente poco relevante.

La mayor importancia en estas pruebas se centra en el análisis de los resultados de las frecuencias *top* y frecuencias prefijadas respecto a los resultados obtenidos al realizar todo el algoritmo ACL completo. Sigue siendo común el uso de las frecuencias prefijadas cuando se realizan experimentos basados en SSVEP en los laboratorios de todo el mundo. Una aproximación más lógica es el uso de las frecuencias *top* dado que implica el estudio de qué estímulos son mejores para cada usuario concreto y el uso de dichos estímulos en la fase de experimentación final. Sin embargo, con el uso del algoritmo se espera obtener un mejor resultado que con el uso de las frecuencias *top* por el hecho de combinar este primer estudio con el análisis posterior de compatibilidad entre distintas frecuencias. Esto se ha demostrado previamente en un casco de mayor sensibilidad y con estimulación mediante LEDs (Fernandez-Vargas 2013). Queda por determinar, por tanto, si el algoritmo mejora en desempeño el uso directo de las frecuencias *top* y prefijadas con el casco Emotiv EPOC y la estimulación en pantalla y cuál de las posibles soluciones del algoritmo arroja mejores resultados. Se espera que el algoritmo con criterio de puntuación sea capaz de obtener el mejor desempeño por su configuración específica para el tipo de análisis que se está desempeñando con este casco.

5.3. RESULTADOS DEL EXPERIMENTO

Se van a realizar una serie de pruebas con la interfaz cerebro-máquina diseñada de cara a evaluar distintos aspectos:

- **Fiabilidad de la interfaz:** Medida de la eficacia de la interfaz y del impacto del algoritmo ACL, frecuencias *top* y prefijadas en el rendimiento del sistema.

- **Separación entre estímulos:** Se ha probado a alterar la distancia entre los distintos estímulos que se muestran en pantalla.
- **Morfología de los estímulos:** Se ha comprobado cuál es el desempeño de la fiabilidad del sistema si en vez de utilizar cuadros blancos como figuras se utilizan figuras de tipo tablero de ajedrez.
- **Reproducibilidad de resultados:** Si el algoritmo arroja resultados similares a lo largo del tiempo significará que tiene una gran estabilidad a la hora de ofrecer resultados. Todo ello junto con la capacidad de hacer frente a la no estacionariedad del cerebro de los usuarios servirá, en cierta medida, para validar los resultados.

5.3.1. Fiabilidad de la interfaz

Las pruebas anteriormente descritas en el apartado 5.1 han sido realizadas a un número total de 17 individuos, dos de los cuales mostraron poca sensibilidad a la estimulación con parpadeo. Pese a ello, a esas personas también se les realizó la fase de estimulación de cara a comprobar si en estos casos extremos el uso de las frecuencias *top* conseguía mejorar en resultados a las frecuencias prefijadas. Antes de las pruebas, los voluntarios confirmaron que no tenían fotosensibilidad ni predisposición a la epilepsia.

Se han realizado los experimentos sobre 17 personas, dos de las cuales no mostraron suficiente sensibilidad a la estimulación y no generaron señales específicas para un número mínimo de frecuencias. La siguiente tabla muestra el desarrollo del entrenamiento que se ha realizado sobre todos los individuos en base al *success rate* de cada fase de pruebas, las frecuencias de estimulación utilizables para cada usuario obtenidas a partir de la primera fase del algoritmo ACL ordenadas de mejor a peor y el conjunto de frecuencias solución adaptada a cada usuario que ha proporcionado finalmente un mejor resultado de acuerdo al criterio de la fiabilidad:

| ID Usuario | Nº frecuencias válidas | SR segunda fase | SR tercera fase | SR total | Frecuencias válidas | Frecuencias mejor resultado |
|------------|------------------------|-----------------|-----------------|----------|---------------------------|-----------------------------|
| 1 | 8 | 0.883 | 0.654 | 0.807 | 6,7,10,8,9,19,11,17 | 6,7,10,19 |
| 2 | 5 | 0.833 | - | 0.833 | 6,7,10,8,9 | 6,7,8,10 |
| 3 | 7 | 0.75 | 0.531 | 0.687 | 7,19,8,18,13,20,17 | 7,8,18,19 |
| 4 | 2 | - | - | - | 7, 13 | 7,8,13,20 |
| 5 | 2 | - | - | - | 20,16 | 12,16,19,20 |
| 6 | 9 | 0.612 | 0.304 | 0.509 | 20,13,9,12,11,16,17,14,15 | 9,12,13,20 |
| 7 | 8 | 0.654 | 0.442 | 0.587 | 6,7,20,8,15,13,19,18 | 6,7,19,20 |
| 8 | 5 | 0.642 | - | 0.642 | 6,7,8,19,20 | 6,7,19,20 |
| 9 | 7 | 0.855 | 0.619 | 0.787 | 6,7,20,8,19,18,15 | 6,7,19,20 |
| 10 | 7 | 0.855 | 0.668 | 0.802 | 6,7,10,8,11,9,13 | 6,7,10,11 |
| 11 | 4 | - | - | - | 6,7,17,8 | 6,7,8,17 |
| 12 | 4 | - | - | - | 6,7,8,11 | 6,7,8,11 |
| 13 | 7 | 0.745 | 0.519 | 0.680 | 6,7,10,13,8,18,19 | 6,7,10,13 |
| 14 | 6 | 0.812 | 0.569 | 0.731 | 6,7,10,8,9,11 | 6,7,8,10 |
| 15 | 7 | 0.67 | 0.375 | 0.586 | 6,7,10,18,8,13,19 | 7,8,10,19 |
| 16 | 8 | 0.529 | 0.296 | 0.451 | 6,16,7,20,17,9,15,11 | 6,9,15,20 |
| 17 | 9 | 0.78 | 0.6 | 0.719 | 6,7,10,8,13,11,17,19,18 | 6,9,15,20 |

Tabla 5-1: Resultados del entrenamiento, siendo SR el success rate, esto es, número de aciertos respecto al número total de detecciones posibles en la prueba. La SR total se refiere a la suma de todos los intentos promediada.

El número de frecuencias válidas se ha obtenido a partir de la primera fase de la prueba, en la cual se fija un umbral que la SNR resultado debe superar para considerar que se ha producido una estimulación significativa. Se puede apreciar como, por ejemplo, en casos como el de los usuarios número 6 y 7, un mayor número de frecuencias válidas no implica un buen desempeño en el resto de fases. Esto pone de manifiesto la importancia de la interferencia entre estímulos; un usuario sensible a muchos de los estímulos de forma aislada muestra un comportamiento mucho peor cuando tiene más de un estímulo dentro de su campo visual. Otros usuarios, como en el caso del 11 y 12, solo han logrado superar la primera fase con 4 frecuencias válidas, con lo que no hay utilidad en realidad el algoritmo ACL posterior. Como se comentó anteriormente, existen dos usuarios, concretamente el 4 y 5, a los que fue imposible detectar el número mínimo de frecuencias necesarias para realizar la fase de entrenamiento y estimulación. Para el caso de estas personas se ha forzado una simulación con las 4 frecuencias que hayan obtenido una mayor puntuación en la primera fase, superasen o no el umbral. El objetivo de esta simulación es ver si en su caso el uso de las frecuencias *top* mejora al de las frecuencias prefijadas. Los casos más

curiosos nos los encontramos en los usuarios 16 y 17; para estos usuarios, pese a obtener un alto número de frecuencias con SSVEPs detectables, las frecuencias que mejores resultados han arrojado son las frecuencias prefijadas. Para el caso del usuario número 16 todas las frecuencias prefijadas están dentro del conjunto de frecuencias válidas. La solución arrojada por el algoritmo ACL en ese caso ha sido (20, 6, 7, 16), variando solo el conjunto de frecuencias (7, 16) por el (9, 15). Esta discrepancia puede deberse a que las frecuencias 7 y 16 cuentan con una posición inicial muy ventajosa al comienzo del algoritmo ACL al haber sido unas de las frecuencias que mayor puntuación habían obtenido en la primera fase. El caso del usuario 17 tiene otra explicación; en la primera fase del entrenamiento se excluyen automáticamente las frecuencias que son armónicas/subarmónicas unas de otras en favor de aquellas que han obtenido una mayor puntuación. Así, en ese caso, podemos ver como las frecuencias 6,18 y 10, que presentan relaciones de armonía con 6,9 y 20, forman parte de los resultados posibles. Sin embargo, el hecho de que se elijan unas de esas frecuencias en función de otras por su mayor puntuación en la primera fase no implica que tengan un mejor desempeño entre ellas una vez que se muestran simultáneamente en pantalla. Es otra de las pruebas de que un análisis concienzudo de compatibilidad entre frecuencias es muy necesario, más allá de la magnitud de los estímulos evocados individualmente. Adicionalmente, el hecho de que dentro del conjunto de frecuencias con mejor resultado esté una que no ha formado parte del conjunto de frecuencias elegibles (los 15 Hz) se explica teniendo en cuenta que el decisor que detecta a qué frecuencia está mirando el usuario conoce todas las frecuencias de estimulación; en este caso una frecuencia que no ha pasado del umbral mínimo en la primera fase puede tener un buen desempeño final pues el decisor podrá saber que el usuario está mirando a dicho estímulo por descarte, dada la baja puntuación de las otras tres alternativas a las que no está mirando.

Lo anterior constituye una serie de excepciones al algoritmo implementado que merece la pena considerar, pues pueden ayudar a mejorar el algoritmo en futuras implementaciones.

Por otra parte, el resto de resultados entran dentro de lo esperado. Se puede observar, además, como en la tercera fase del algoritmo ACL se produce una bajada en la SR o puntuación final respecto de la fase anterior. Esto se puede deber a que estamos estimulando al usuario con 4 frecuencias simultáneamente en lugar de con 2, incrementando así la dificultad para la identificación de la elección del usuario. No se deben olvidar las causas inherentes al propio sujeto, que puede encontrar dificultad para generar las señales deseadas debido a la mayor interferencia producida por el resto de estímulos.

Los resultados medidos en porcentaje de fiabilidad por parte del sistema para cada uno de los usuarios y según el tipo de solución se resumen en la siguiente tabla:

| ID Usuario | Top | Compatib. | Puntuación | Prefijadas | Media |
|------------|--------|-----------|------------|------------|-----------------|
| 1 | 70% | 65% | 80% | 66.7% | 70.425% |
| 2 | 63.33% | 63.33% | 63.33% | 53.33% | 60.83% |
| 3 | 63.33% | 58.33% | 63.33% | 45% | 57.4975% |
| 4* | 45% | | | 31.66% | 38.33% |
| 5* | 58.33% | | | 46.66% | 52.495% |
| 6 | 45% | 58.33% | 58.33% | 41.66% | 50.83% |
| 7 | 48.33% | 41.66% | 50% | 53.33% | 48.33% |
| 8 | 46.66% | 48.33% | 48.33% | 36.33% | 44.9125 |
| 9 | 71.66% | 76.66% | 76.66% | 53.33% | 69.5775% |
| 10 | 48.33% | 50% | 65% | 65% | 57.0825% |
| 11** | 70% | | | 65% | 67.5% |
| 12** | 58.33% | | | 40% | 49.165% |
| 13 | 60% | 50% | 60% | 28.33% | 49.5825% |
| 14 | 73.33% | 60% | 73.33% | 65% | 67.915% |
| 15 | 25% | 31.66% | 41.66% | 31.66% | 32.495% |
| 16 | 63.33% | 58.33% | 63.33% | 78.33% | 65.83% |
| 17 | 70% | 40% | 73.33% | 78.33% | 65.415% |

* Individuos con dificultad para generar SSVEP.

** Individuos con solo 4 frecuencias por encima del umbral.

Tabla 5-2: Resultados de la experimentación. Las filas finales son los resultados obtenidos como media de todos los resultados particulares.

Analizando los resultados obtenidos para cada usuario se puede apreciar como la fiabilidad obtenida mediante el algoritmo ACL bajo el criterio de máxima puntuación suele ser el que obtiene los mejores resultados. La virtud que se puede observar del algoritmo con este criterio es que siempre es capaz de igualar o superar la máxima fiabilidad obtenida por el resto de métodos; para los usuarios 3, 13 y 16 el criterio de puntuación arroja los mismos resultados que las frecuencias *top* y son los mejores resultados, mientras que para los usuarios 6, 8 y 9 el criterio de puntuación arroja los mismos resultados que se han obtenido mediante puntuación. En el resto de los casos el algoritmo ACL mediante máxima puntuación es el que obtiene mejores resultados sin ningún tipo de empate, salvo en los dos últimos casos discutidos con anterioridad.

La media, así como las frecuencias de mejor resultado de la tabla 5.1 son un indicativo claro de las particularidades de cada usuario; hay usuarios para los cuales los SSVEP serán mucho más distintivos que para otros usuarios, independientemente del set de frecuencias solución que se elija. Las razones para esto pueden ser muy variadas, desde dificultad a la hora de capturar la señal hasta a razones biológicas, como que el cerebro no reaccione de igual manera frente a todos los estímulos visuales, que se produzcan atenuaciones de forma no uniforme a distintas frecuencias... Por otro lado, todos los

usuarios obtienen unas frecuencias determinadas que se adaptan a sus circunstancias y, además, obtienen unos resultados diferentes para cada conjunto de soluciones consideradas. Con esto queda claro, no solo que cada usuario se comporta de forma diferente a la estimulación si no que, un mismo usuario puede reaccionar de forma muy diferente a la estimulación visual en función de la frecuencia de parpadeo. Un caso interesante es el del usuario número 16; se trata de un usuario que, según la tabla 5-1 obtiene un gran número de frecuencias válidas y sin embargo luego su desempeño en las siguientes fases de la prueba es bastante pobre debido a que se trata de un usuario para el cual múltiples estímulos en pantalla de forma simultánea interfieren entre sí de forma muy fuerte.

En general los resultados obtenidos mediante el algoritmo ACL bajo el criterio de puntuación son bastante buenos y ofrecen un desempeño razonable, teniendo en consideración las características del casco y la interfaz de estimulación. Sobre todo es importante también el apreciar como el hacer uso de un algoritmo adaptativo al usuario sirve para incrementar de forma notable la fiabilidad del sistema en prácticamente todos los usuarios, salvo los dos últimos como ya se ha comentado con anterioridad. Queda, por tanto, demostrada la individualidad de cada usuario a la hora de percibir los estímulos visuales, así como las peculiaridades dentro de cada usuario para distintas frecuencias y, además, la conveniencia de utilizar métodos para adaptarnos a estas particularidades. Así vemos como, en general, el uso de frecuencias *top* mejora la fiabilidad del sistema sobre las frecuencias prefijadas y, adicionalmente, el algoritmo ACL con criterio de máxima puntuación adaptado al Emotiv EPOC mejora el desempeño proporcionado por las frecuencias *top* y el resto.

5.3.2. Separación entre estímulos

La distancia entre los iconos a la hora de realizar los experimentos que se ha utilizado hasta el momento ha sido de 550 píxeles horizontales y 350 píxeles en vertical, dada la resolución panorámica de la pantalla. Se ha realizado una reducción homogénea de la distancia a la mitad de sus valores, quedando en 275 píxeles horizontales y 175 píxeles en vertical. El objetivo de este desplazamiento es comprobar cómo afecta la distancia a la fiabilidad del sistema. Es previsible que una menor distancia entre estímulos pueda incrementar la interferencia entre estímulos adyacentes y baje la fiabilidad del sistema. Para realizar este experimento los usuarios han tenido que entrenar previamente la interfaz mediante el algoritmo ACL. Posteriormente, con el sistema entrenado y utilizando las frecuencias que mejor se hayan adaptado al usuario, se ha procedido a realizar las dos fases de experimentación con las diferentes distancias entre objetos. Se ha realizado el experimento con 5 personas con los siguientes resultados:

| ID Usuario | Iconos separados | Iconos juntos |
|--------------|------------------|---------------|
| 1 | 93% | 83.33% |
| 2 | 48.33% | 38.33% |
| 10 | 35% | 33.33% |
| 12 | 35% | 26.66% |
| 17 | 73.33% | 70% |
| Media | 56.932% | 50.33% |

Tabla 5-3: Resultados para distinta distancia entre iconos.

En este caso se ha dado el caso de que los sujetos que han formado parte de estas pruebas han obtenido unas medias un poco inferiores al caso general con más sujetos del apartado anterior. Independientemente de esto, se puede observar como existe una clara correlación entre la distancia de los iconos y la fiabilidad del sistema; en todos los casos analizados existe un peor rendimiento cuando juntamos los iconos respecto a su posición inicial. No obstante no afecta a todos los usuarios por igual y, en algunos casos, es un efecto poco perceptible en el desempeño final del sistema.

5.3.3. Morfología de los estímulos

Hasta ahora todos los entrenamientos y experimentaciones se han utilizado imágenes planas cuadradas de 200 píxeles de ancho de color blanco. Hay que tener en consideración que para realizar experimentos basados en SSVEP se suelen utilizar dos alternativas, como ya se comentó en el apartado 3.3: imágenes planas o imágenes de tipo tablero de ajedrez. El fin de utilizar imágenes de tipo tablero de ajedrez es limitar los efectos que se producen al mostrar imágenes planas en una pantalla debido a la frecuencia de refresco. Dichos efectos incluyen situaciones molestas y que pueden distraer a los usuarios, como el hecho de que al mostrar un parpadeo en un momento dado el icono se vea cortado debido a que la pantalla no ha podido refrescar todo el icono correctamente. El uso de estas imágenes tiene también sus desventajas, siendo una de las más importantes el hecho de que al ser imágenes formadas por múltiples cuadrados más pequeños el usuario no es capaz de fijar la atención en todo el objeto global o en uno de los cuadros pequeños en particular. No existe una posición clara acerca de qué tipo de imágenes ofrecen un mejor resultado final y, por ello, se ha decidido comprobar qué convendría más con el uso de este casco de captura de señales y la metodología del análisis que se está aplicando. Este experimento necesita que la interfaz haya sido previamente entrenada mediante el algoritmo ACL. Este entrenamiento se ha realizado con imágenes planas dado que no se ha considerado relevante el uso de uno u otro tipo de imágenes en esta fase, puesto que las frecuencias

a las que un usuario es más sensible deberán ser las mismas. Realizado el entrenamiento, la fase de experimentación consistirá en una primera estimulación con imágenes planas seguida por una estimulación con imágenes de tipo tablero de ajedrez. Posteriormente se comparará el desempeño obtenido en ambas estimulaciones. Los resultados para los 5 usuarios que han tomado parte en estas pruebas son los siguientes:

| ID Usuario | Imágenes planas | Tablero de ajedrez |
|------------|-----------------|--------------------|
| 1 | 93% | 66.66% |
| 2 | 48.33% | 28.33% |
| 10 | 35% | 41.66% |
| 12 | 35% | 33.33% |
| 17 | 73.33% | 43.33% |
| Media | 56.932% | 42.66% |

Tabla 5-4: Resultados mostrando imágenes planas contra imágenes tipo tablero de ajedrez.

Para este experimento las imágenes de tipo tablero de ajedrez han supuesto un muy peor desempeño del sistema en casi todos los usuarios. Para el usuario 3, no obstante, el uso de este tipo de imágenes ha supuesto una ligera mejora en el uso de la interfaz con respecto a las imágenes planas. Pese a tratarse de un caso aislado, conviene tenerlo en consideración; es posible que según el usuario llegue a ser muy determinante qué tipo de figuras se representan en pantalla y que éste sea muy sensible a los artefactos que se muestran en la misma por efecto de la frecuencia de refresco. Estos resultados pueden verse condicionados por el casco que está siendo utilizado; con el uso de cascos con electrodos posicionados explícitamente para la captura idónea de SSVEP el uso de unos u otros estímulos suele ofrecer unos rendimientos similares, dejando abierto a discusión el dilema de qué tipo de estimulación por pantalla ofrece mejores resultados en casos generales (Allison et al. 2008).

Respecto al uso de una paleta de colores más amplia a la hora de mostrar los estímulos se ha optado finalmente por considerar únicamente el color blanco en todos los experimentos realizados dado que, dada la naturaleza del ojo humano, el color blanco puede generar señales más distintivas al estimular todos los conos de forma simultánea y ser, por tanto, un color más perceptible (Cao, Wan, and Mak 2012).

5.3.4. Reproducibilidad de los resultados

Otro de los experimentos derivados del desarrollo de la interfaz cerebro-máquina es la validación del algoritmo ACL. Si bien su utilidad ha quedado contrastada en el experimento principal, puede ser de gran interés el realizar un segundo análisis a algunas de las personas que formaron parte de dicha primera prueba. El objetivo de ello es comprobar la constancia de los resultados con el fin de validar los mismos. Si los resultados son similares significará que el algoritmo es capaz de adaptarse a cada usuario y que, además, es capaz de proporcionar resultados constantes más allá de las variaciones puntuales que se puedan estar produciendo en el cerebro que esté generando las señales evocadas.

Para realizar esta parte del experimento se ha contado con 5 de los usuarios que formaron parte del experimento principal. Se les ha pedido que repitan el entrenamiento de la interfaz unos 15 días después de haber realizado las primeras pruebas. Los resultados han sido los siguientes:

| ID Usuario | PRIMERA | | | SEGUNDA | | |
|------------|------------|----------|------------|------------|----------|------------|
| | Nº válidas | Top | Puntuación | Nº válidas | Top | Puntuación |
| 1 | 8 | 6,7,10,8 | 6,7,10,19 | 8 | 6,7,10,8 | 6,7,8,10 |
| 2 | 5 | 6,7,10,8 | 6,7,8,10 | 5 | 8,6,10,7 | 6,7,8,10 |
| 10 | 7 | 6,7,10,8 | 6,7,10,11 | 6 | 6,7,10,8 | 6,8,9,10 |
| 12 | 4 | 6,7,8,11 | 6,7,8,11 | 4 | 6,7,8,13 | 6,7,8,13 |
| 17 | 9 | 6,7,10,8 | 6,7,10,13 | 9 | 6,7,20,8 | 6,7,10,13 |

Tabla 5-5: Resultados mostrando las 4 frecuencias elegidas en primer y segundo entrenamiento de la interfaz, además del número de frecuencias válidas o utilizables en base a los resultados de la primera fase de las pruebas.

De los resultados obtenidos de la tabla anterior se pueden desprender varias conclusiones:

- Parece existir una relación entre los resultados obtenidos en una primera prueba para cada usuario y, posteriormente, en el segundo experimento. Como mucho siempre se produce un cambio de una única frecuencia entre las dos sesiones.
- Tanto en este experimento como en el principal, siempre se observa una predilección por parte de la interfaz respecto a las frecuencias 6Hz, 7Hz, 8Hz y 10Hz. En general se aprecia una tendencia por las bajas frecuencias salvo en caso

puntuales. Una de las razones para ello puede ser la mayor facilidad del casco para capturar señales SSVEP de baja frecuencia.

Estos resultados parecen indicar que el algoritmo se comporta de forma constante en el tiempo. No obstante, quizás no sea una prueba muy eficaz para evaluar este aspecto debido al hecho de que este casco limita mucho la elección de las posibles frecuencias; el hecho de que el rango disponible vaya desde los 6Hz a los 20Hz y que, además, se excluyan en los posibles resultados a frecuencias múltiplos unas de otras limita mucho las opciones de posibles frecuencias resultado. En el mejor de los casos, los usuarios dispondrán como máximo de 10 frecuencias *top* y 10 alternativas para desarrollar el algoritmo. Pocos son los usuarios que llegan a tener todas las alternativas disponibles e incluso, en ese caso, suele existir una preferencia debido al casco de captura de datos por las bajas frecuencias. Por tanto, los resultados tienden a ser similares para todos los usuarios y, para un mismo usuario, hay una gran probabilidad de que todas o casi todas las soluciones se repitan en sucesivos experimentos. Aunque hay indicios de que las mejores frecuencias se mantienen constantes para cada individuo, es necesario realizar un estudio más exhaustivo para determinar este punto.

5.4. CONCLUSIONES Y MEJORAS

El objetivo inicial del proyecto era la creación de una interfaz cerebro-máquina desde el punto de vista de la optimización de la usabilidad con un coste reducido. Para ello se ha utilizado el casco de propósito general Emotiv Eloc. La primera parte del proyecto ha consistido en comprobar si este casco en concreto era capaz de captar con la suficiente intensidad las señales evocadas. Para ello se hizo uso de una primera interfaz gráfica y se analizaron los resultados de forma *offline*. El resultado fue satisfactorio pero con matices; la relación señal a ruido proporcionada por el casco no era muy elevada. Para poder paliar estos efectos se puso especial cuidado en diseñar la interfaz gráfica e incorporar la categorización de resultados al realizar el análisis de señal, esto es, que en el cálculo de puntuaciones solo se consideren las frecuencias que están formando parte de la estimulación y se obvian aquellas frecuencias que se sabe que no están siendo estimuladas. Todo ello combinado con el uso del algoritmo ACL de entrenamiento de la interfaz se ha traducido en unos rendimientos en todos los casos por encima del 50% de fiabilidad, llegando hasta un 62% de media utilizando las soluciones del algoritmo ACL por puntuación. Para algunos usuarios la fiabilidad del sistema se sitúa por encima del 70%, siendo en estos casos el sistema prácticamente utilizable. Los resultados son agrisados; no se esperaba con este casco un gran rendimiento dado que no está capacitado para capturar SSVEP de forma eficiente y el hecho de conseguir esos porcentajes se considera un éxito pero, por desgracia, la fiabilidad del sistema no es lo

suficientemente alta como para que esta interfaz cerebro-máquina sea utilizada de forma comercial por usuarios que necesiten de ella.

El incluir el algoritmo ACL de ciclo cerrado como parte del proceso de entrenamiento ha sido un acierto. Cabe recordar que se trata de una técnica innovadora dado que en general el entrenamiento de estas interfaces se basa en la búsqueda de las frecuencias *top* únicamente, esto es, las frecuencias que generan SSVEPs más distintivos en el usuario. Al incluir en el entrenamiento la posibilidad de que se evalúe el comportamiento de múltiples estímulos en pantalla de forma simultánea ha permitido optimizar los resultados, como se ha visto anteriormente. Adicionalmente, el algoritmo ACL había sido utilizado con anterioridad únicamente en los laboratorios del Grupo de Neurocomputación Biológica con estimulación LED, el casco de licra Aegis Array y un amplificador mucho más potente. El hecho de que la misma filosofía haya funcionado con el Emotiv Epoc con estimulación implementada por software en un monitor habla acerca de la capacidad de adaptación del algoritmo, al cual solo ha habido que hacerle unas pequeñas modificaciones, más que por el casco en sí, por los cambios introducidos en el análisis de señal en los cuales el sistema conoce en todo momento las señales que están formando parte en la estimulación. Esto último constituye otra de las mejoras implementadas sobre la que hacer hincapié: la categorización del análisis de señal en cuatro únicas soluciones posibles gracias a conocer las cuatro frecuencias de las señales estimuladas ha ayudado a mejorar las prestaciones del casco, que mostraba resultados erráticos en las pruebas realizadas durante el desarrollo del proyecto antes de introducir esta serie de mejoras.

La solución más inmediata a los problemas de falta de fiabilidad que se pueden observar pese a todas las mejoras introducidas en el proyecto es el uso de otro casco que permita capturar los SSVEP de forma más eficiente. Esto es posible sin comprometer la comodidad de uso, tan solo utilizando un casco que permita utilizar electrodos en las posiciones Oz y POz. Con ello se lograría capturar las señales de forma eficaz mediante el electrodo en Oz y restar el ruido proveniente de otras zonas del cerebro mediante el POz. Otra solución es la mejora de la calidad de los electrodos, sin que ello implique el uso de métodos más invasivos. Ello serviría para mejorar la relación señal a ruido de las señales deseadas y mejorar la fiabilidad de la interfaz.

Otro apartado susceptible de mejora es el algoritmo ACL. El uso de este algoritmo en este caso constituye una mejora sobre el extendido uso de las frecuencias prefijadas o *top* para realizar interfaces basadas en SSVEP, demostrándose así que el poner un gran énfasis en la adaptación del sistema al usuario es uno de los caminos a seguir. El uso del ciclo cerrado ha proporcionado una fiabilidad extra al sistema al ir adaptándose a la reacción del usuario cuando se muestran múltiples eventos parpadeantes dentro de su campo visual. Además, se ha demostrado como el uso de este algoritmo es capaz de proporcionar resultados muy similares a lo largo del tiempo, lo cual puede llegar a permitir una simplificación del entrenamiento de los usuarios si ya han realizado un primer uso con anterioridad. Sin embargo, se trata de un algoritmo todavía en desarrollo

y que ha tenido que ser especialmente adaptado para el uso de este casco. Además de reajustar las variables existentes puede ser de interés que incluya nuevos parámetros a la hora de realizar el análisis de frecuencias óptimas: tipo de figuras a mostrar, tamaño de las mismas, distancia entre ellas... todo ello intentando evitar que el entrenamiento se convierta en un proceso excesivamente engorroso de cara a mantener la accesibilidad de la interfaz. Por otro lado, el hecho de que el análisis de señal conozca las frecuencias a las que se está realizando la estimulación, limitando así el número de detecciones erróneas, es otra innovación que ha sido introducida en este proyecto respecto a otros proyectos realizados en el Grupo de Neurocomputación Biológica. Sin embargo, sería deseable que la interfaz pudiera conocer si el usuario está observando alguno de los estímulos visuales o si no se encuentra mirando a la pantalla. El uso de un umbral de detección fue descartado por no suponer una mejora apreciable. Sin embargo, si se combina este umbral con un casco o electrodos que proporcionen una mejor calidad de señal, podría llegar a ser una alternativa factible.

Otra de las mejoras que se podrían implementar es el uso de sistemas híbridos; combinar distintas tecnologías dentro de la interfaz puede desembocar en una mejora en la fiabilidad y velocidad del sistema. Por ejemplo, en este caso el uso de tecnologías de tipo *eye gaze* podría servir como validación para los resultados proporcionados por el análisis de señal y, además, proporcionar información sobre si el usuario está mirando o no la pantalla. Finalmente, se da la circunstancia de que un porcentaje de usuarios de entre el 15% y el 30% (Vidaurre and Blankertz 2010) son incapaces de interactuar de forma efectiva con una interfaz cerebro-máquina y de generar las señales necesarias, fenómeno conocido como BCI-Illiteracy. Las mejoras de los algoritmos de adaptación al individuo serán de vital importancia para poder construir interfaces accesibles de una forma universal.

6. MÓDULOS DE INTERÉS AÑADIDOS

6.1. CONTROL DE ROBOT

Uno de los usos de las interfaces cerebro-máquina es el control y movimiento de distintos dispositivos. De cara a facilitar la comunicación de la interfaz creada con algún dispositivo concreto, ya sea una silla, un brazo robótico... se ha creado un módulo de

comunicación con dispositivos externos. El programa llamará a la rutina encargada de gestionar el dispositivo externo. Dicha rutina tiene que tener el nombre de ProgramaRobot. La llamada se produce cuando el programa principal va a proceder a analizar los estímulos del usuario en tiempo real. ProgramaRobot debe recibir los siguientes argumentos:

- Identificador de la cola de intercambio de datos.
- El conjunto de frecuencias con el que se está estimulando al usuario.

Con estos argumentos el dispositivo tendrá acceso a la sección de memoria donde el análisis de señal irá escribiendo los resultados al ritmo de uno por segundo. Además, conocerá cuales son todas las frecuencias disponibles y podrá codificarlas según lo necesite en distintos movimientos o acciones de la máquina.

Una vez que el programa de gestión del dispositivo está inicializado con los argumentos anteriores, deberá mantenerse escuchando la cola cuyo identificador ha sido proporcionado. En cuanto reciba un dato por parte del programa principal realizará las acciones correspondientes y volverá a ponerse en modo escucha.

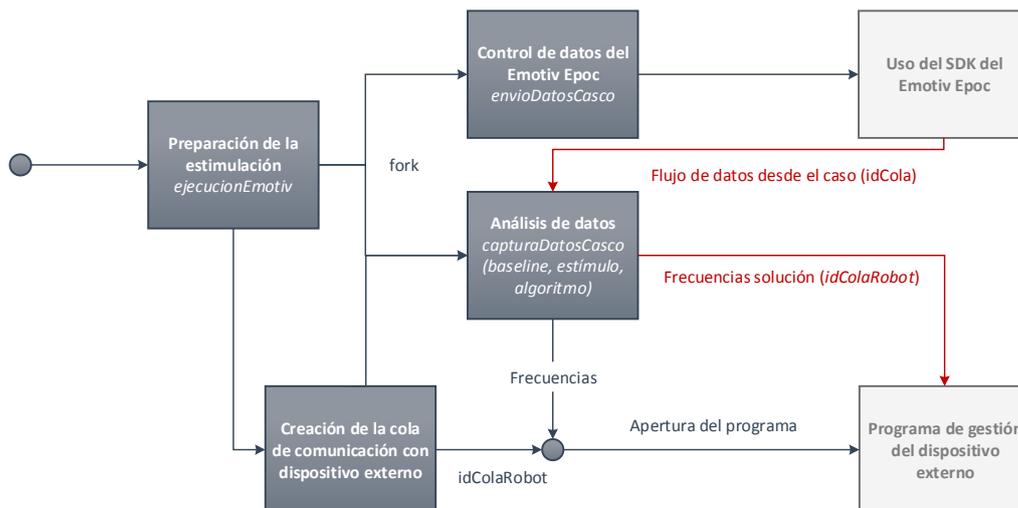


Figura 6-1: Esquema de comunicación con el programa de control del dispositivo.

6.2. GESTIÓN DE USUARIOS Y FICHEROS

Una de las claves del programa se encuentra en la gestión de usuarios. Cada usuario está identificado por una ID en base a la cual se genera una carpeta propia. De esta forma todos los datos de los usuarios quedan almacenados para posteriores series de experimentación o para comprobar errores que hayan podido suceder durante la ejecución o el entrenamiento. Dicha carpeta contendrá datos propios de cada usuario:

- **AnalisisAlgoritmo_FASE1.txt:** Este archivo contiene los resultados obtenidos en la primera fase del algoritmo, esto es, las puntuaciones para el barrido de frecuencias cuando el sujeto está sometido a una única estimulación.
- **Baseline.txt:** Este fichero almacena un vector de frecuencias que contiene los resultados obtenidos para la grabación del *baseline*.
- **BrutosAlgoritmo.txt, brutosBaseline.txt y brutosEstimulos.txt:** Estos archivos contienen los datos brutos grabados directamente desde el casco, sin ningún tipo de procesado.
- **configAnalisis.cfg:** Archivo de configuración que será utilizado por el análisis de señal. Contiene las cuatro frecuencias que están siendo estimuladas.
- **Estimulos.txt:** Este fichero contiene los vectores de frecuencias obtenidos tras el análisis de la señal cuando el sujeto está sometido a estimulación. El programa dibujará un vector por cada ventana de análisis.
- **estimulosUsuario.cfg:** Este fichero de configuración contiene toda la información necesaria para la configuración de la interfaz gráfica en función de las frecuencias obtenidas tras la realización del algoritmo ACL.

Existen, además, otros archivos de configuración general que son necesarios para el funcionamiento del programa. Éstos tienen que ver con la selección de las frecuencias prefijadas que serán útiles para todos los usuarios indistintamente y se encuentran en la carpeta raíz en la que esté instalado el programa:

- **baseline.cfg:** Este archivo de configuración contiene toda la información necesaria para que la interfaz gráfica muestre toda la pantalla en negro para la realización del *baseline*.
- **estimulosPrefijados.cfg:** Este fichero de configuración contiene toda la información necesaria para la configuración de la interfaz gráfica de acuerdo a las frecuencias prefijadas para todos los usuarios.
- **frecuenciasPrefijados.cfg:** Archivo de configuración que será utilizado por el análisis de señal. Contiene las cuatro frecuencias que están siendo estimuladas que serán, en este caso, las prefijadas.

Además, el programa cuenta con dos carpetas con información necesaria para su correcto funcionamiento:

- **Directorio Imágenes:** Almacena todas las imágenes que vayan a ser utilizadas por la interfaz gráfica: estímulos cuadrados, de tipo tablero de ajedrez, flecha indicativa verde....
- **Directorio Algoritmo:** Contiene los ficheros de configuración necesarios para llevar a cabo la ejecución del algoritmo.
 - o **Secuencias:** Cuando se realizan las fases del algoritmo con múltiples estímulos en pantalla, debe haber un indicador verde que señale la posición de la pantalla a la que debe mirar el usuario. En estos archivos se detalla la secuencia que seguirá el indicador verde por la pantalla.
 - o **Archivos efímeros:** Estos serán archivos que son generados puntualmente para realizar alguna tarea concreta. Por ejemplo, los archivos que el programa genera automáticamente de configuración de la interfaz gráfica para realizar la estimulación durante el algoritmo ACL son guardados aquí y borrados después de su uso.

El esquema general de dependencia entre directorios y archivos se encuentra en el Apéndice 2 – Instalación y librerías.

6.3. WEBCAM EN LA INTERFAZ DE ESTIMULACIÓN

Se ha considerado de interés incluir, durante la fase de estimulación, la posibilidad de mostrar por pantalla la señal de vídeo procedente de una webcam junto a los estímulos. Esto puede servir para que, por ejemplo, personas con discapacidad puedan operar un brazo robótico sin la necesidad de apartar la vista de la pantalla gracias a una cámara que esté grabando la posición del brazo en cada instante. El objetivo, en cualquier caso, es proporcionar un *feedback* visual al usuario de cómo está operando el dispositivo sin que sea necesario apartar la vista de la pantalla, con la consiguiente pérdida de concentración y con el riesgo de enviar al dispositivo señales no deseadas durante ese intervalo de tiempo en el que el usuario no está sometido a la estimulación visual.

Para ello se hará uso de OpenCV. Estas librerías fueron descartadas como alternativa para la generación de estímulos visuales dadas sus limitaciones. Sin embargo, proporcionan muchas utilidades para la captación y procesado de imágenes, lo que hace que su uso sea ideal para mostrar por pantalla la información procedente de una cámara web. Tanto Qt como OpenCV tienen flexibilidad suficiente para ser usadas de forma conjunta en una misma interfaz, lo que nos permite incrustar objetos CameraWidget y CvCapture dentro de la ventana principal encargada de la estimulación basada en Qt.

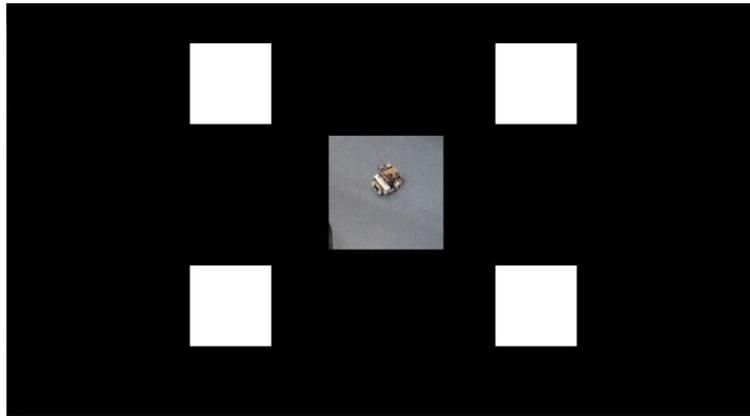


Figura 6-2: Ventana de estimulación con los cuatro estímulos y la grabación procedente de la cámara web.

Como se ve en la figura anterior, la disposición de los objetos en una misma ventana permite al usuario monitorizar el movimiento del dispositivo, un pequeño robot en este caso, con un ligero movimiento de ojos.

Se ha apreciado, no obstante, que la inclusión de la información de la cámara web genera una carga de trabajo extra lo suficientemente elevada como para alterar la frecuencia de parpadeo de los estímulos. Para evitar esto se ha limitado la frecuencia de refresco de estas imágenes a una captura cada medio segundo; con esto se logra que las variaciones de frecuencia en los estímulos debido al procesamiento extra no sean apreciables y que el usuario pueda mantener monitorizado al dispositivo en tiempo real.

REFERENCIAS

- Allison, B, I Sugiarto, B Graimann, and A Gräser. 2008. "Display Optimization in SSVEP BCIs." *Proceedings of the Computer-Human Interaction*.
http://hmi.ewi.utwente.nl/chi2008/chi2008_files/allison.pdf.
- Amiri, Setare, Reza Fazel-rezai, and Vahid Asadpour. 2013. "A Review of Hybrid Brain-Computer Interface Systems" 2013.
- Amiri, Setare, Ahmed Rabbi, Leila Azinfar, and Reza Fazel-rezai. 2013. "A Review of P300 , SSVEP , and Hybrid P300 / SSVEP Brain- Computer Interface Systems."
- Arafat, Ibrahim. 2013. "Brain Computer Interface : Past , Present & Future," 1–6.
- Cao, T, F Wan, and PU Mak. 2012. "Flashing Color on the Performance of SSVEP-Based Brain-Computer Interfaces." *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, 1819–22.
http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6346304.
- Capilla, Almudena, Paula Pazo-Alvarez, Alvaro Darriba, Pablo Campo, and Joachim Gross. 2011. "Steady-State Visual Evoked Potentials Can Be Explained by Temporal Superposition of Transient Event-Related Responses." *PloS One* 6 (1): e14543. doi:10.1371/journal.pone.0014543.
- Chumerin, Nikolay, Nikolay V Manyakov, Marijn Van Vliet, Adrien Combaz, Marc M Van Hulle, Nikolayv Manyakov, Arne Robben, and Marc Vanhulle. 2012. "Processing and Decoding Steady-State Visual Evoked Potentials for Brain-Computer Interfaces." *Digital Image and Signal Processing for Measurement Systems*, 1–33.
- Diez, Pablo F, Sandra M Torres Müller, Vicente a Mut, Eric Laciari, Enrique Avila, Teodiano Freire Bastos-Filho, and Mário Sarcinelli-Filho. 2013. "Commanding a Robotic Wheelchair with a High-Frequency Steady-State Visual Evoked Potential Based Brain-Computer Interface." *Medical Engineering & Physics* 35 (8). Institute of Physics and Engineering in Medicine: 1155–64.
doi:10.1016/j.medengphy.2012.12.005.
- Elshout, JA. 2009. "Review of Brain-Computer Interfaces Based on the P300 Evoked Potential." <http://igitur-archive.library.uu.nl/student-theses/2009-0323-200602/UUindex.html>.
- Emotiv. 2014. "Emotiv EPOC & Testbench Specifications."
[http://www.emotiv.com/product-specs/Emotiv EPOC Specifications 2014.pdf](http://www.emotiv.com/product-specs/Emotiv_EPOC_Specifications_2014.pdf).

- Fernandez-Vargas, J. 2013. "Assisted Closed-Loop Optimization of SSVEP-BCI Efficiency." *Frontiers in Neural Circuits*.
<http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3580891/>.
- Gao, Shangkai, Yijun Wang, Xiaorong Gao, and Bo Hong. 2014. "Visual and Auditory Brain-Computer Interfaces." *IEEE Transactions on Bio-Medical Engineering* 61 (5): 1436–47. doi:10.1109/TBME.2014.2300164.
- Guger, Christoph, Shahab Daban, Eric Sellers, Clemens Holzner, Gunther Krausz, Roberta Carabalona, Furio Gramatica, and Guenter Edlinger. 2009. "How Many People Are Able to Control a P300-Based Brain-Computer Interface (BCI)?" *Neuroscience Letters* 462 (1): 94–98. doi:10.1016/j.neulet.2009.06.045.
- Herrmann, Christoph S. 2001. "Human EEG Responses to 1-100 Hz Flicker: Resonance Phenomena in Visual Cortex and Their Potential Correlation to Cognitive Phenomena." *Experimental Brain Research* 137 (3-4): 346–53.
- Höhne, J, E Holz, P Staiger-Sälzer, and KR Müller. 2014. "Motor Imagery for Severely Motor-Impaired Patients: Evidence for Brain-Computer Interfacing as Superior Control Solution." *PloS One*.
<http://dx.plos.org/10.1371/journal.pone.0104854.g005>.
- Hornero, Roberto, Rebeca Corralejo, and Daniel Álvarez. 2012. "Brain-Computer Interface (bCI) Aplicado Al Entrenamiento Cognitivo Y Control Domótico Para Prevenir Los Efectos Del Envejecimiento." *Lychnos*, 29–34.
- Konrad, Peter, and Todd Shanks. 2010. "Implantable Brain Computer Interface: Challenges to Neurotechnology Translation." *Neurobiology of Disease* 38 (3). Elsevier Inc.: 369–75. doi:10.1016/j.nbd.2009.12.007.
- Krepki, R, B Blankertz, G Curio, and KR Müller. 2007. "The Berlin Brain-Computer Interface (BBCI)—towards a New Communication Channel for Online Control in Gaming Applications." *Multimedia Tools and Applications*.
<http://link.springer.com/article/10.1007/s11042-006-0094-3>.
- Lebedev, Mikhail. 2014. "Brain-Machine Interfaces: An Overview." *Translational Neuroscience* 5 (1): 99–110. doi:10.2478/s13380-014-0212-z.
- Leeb, Robert, Reinhold Scherer, Claudia Keinrath, Gert Pfurtscheller, Felix Y Lee, Horst Bischof, and Mel Slater. 2007. "Combining BCI and Virtual Reality: Scouting Virtual Worlds." *Toward Brain-Computer Interfacing*, no. 23: 393–407.
- Maynard, Edwin M, Craig T Nordhausen, and Richard A Normann. 1997. "The Utah Intracortical Electrode Array : A Recording Structure for Potential Brain-Computer Interfaces" 102: 228–39.

- McFarland, Dennis J, and Jonathan R Wolpaw. 2011. "Brain-Computer Interfaces for Communication and Control." *Communications of the ACM*. doi:10.1145/1941487.1941506.
- Nicolas-Alonso, LF, and J Gomez-Gil. 2012. "Brain Computer Interfaces, a Review." *Sensors*. <http://www.mdpi.com/1424-8220/12/2/1211/pdf>.
- Ortner, R, C Guger, and R Prueckl. 2010. "Ssvep Based Brain-Computer Interface for Robot Control." *Computers Helping People with Special Needs*. http://link.springer.com/chapter/10.1007/978-3-642-14100-3_14.
- Pearlmutter, Barak A, Tomas E Ward, Christopher Soraghan, and Charles Markham. 2008. "Hemodynamics for Brain-Computer Interfaces [," no. January: 87–94.
- Ruiz, Sergio, Korhan Buyukturkoglu, Mohit Rana, Niels Birbaumer, and Ranganatha Sitaram. 2014. "Real-Time fMRI Brain Computer Interfaces: Self-Regulation of Single Brain Regions to Networks." *Biological Psychology* 95 (January). Elsevier B.V.: 4–20. doi:10.1016/j.biopsycho.2013.04.010.
- Schalk, Gerwin, and Eric C. Leuthardt. 2011. "Brain-Computer Interfaces Using Electrographic Signals." *IEEE Reviews in Biomedical Engineering* 4: 140–54. doi:10.1109/RBME.2011.2172408.
- Sulzer, J, S Haller, F Scharnowski, N Weiskopf, N Birbaumer, M L Blefari, A B Bruehl, et al. 2013. "Real-Time fMRI Neurofeedback: Progress and Challenges." *NeuroImage* 76: 386–99. doi:10.1016/j.neuroimage.2013.03.033.
- Vidaurre, Carmen, and Benjamin Blankertz. 2010. "Towards a Cure for BCI Illiteracy." *Brain Topography* 23 (2): 194–98.
- Volosyak, I., H. Cecotti, and A. Gräser. 2009. "Impact of Frequency Selection on LCD Screens for SSVEP Based Brain-Computer Interfaces." In *Proc. 4th Int. IEEE/EMBS Conference on Neural Engineering NER 09*, 706–13. Antalya, Turkey.
- Volosyak, Ivan, Diana Valbuena, Thorsten Lüth, Tatsiana Malechka, and Axel Gräser. 2011. "BCI Demographics II: How Many (and What Kinds Of) People Can Use a High-Frequency SSVEP BCI?" *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 19 (3): 232–39.
- Wikipedia. 2015. "Graphical User Interface." *Wikipedia*. http://en.wikipedia.org/wiki/Graphical_user_interf.

GLOSARIO

| | |
|---|--|
| ALS – Esclerosis alteral amiotrófica | Enfermedad degenerativa de tipo neuromuscular. Provoca una parálisis muscular progresiva. |
| ACL – Assisted Closed Loop | Entrenamiento de ciclo cerrado que permite a la interfaz adaptarse a las frecuencias de estimulación propicias para un usuario concreto y cómo unas interfieren con otras cuando se existen varios estímulos simultáneamente. |
| Análisis offline | Se trata del análisis de los datos de forma posterior a su captura. |
| Análisis online | Análisis de los datos tomados en el momento de su captura o de forma inmediatamente posterior. |
| Armónicos | Componentes de una señal cuya frecuencia es un múltiplo de la fundamental. El primer armónico de una frecuencia dada es el doble del valor de dicha frecuencia. |
| Baseline | Se entiende como <i>baseline</i> en el contexto de este proyecto al estado del individuo en una situación de reposo o no estimulación visual para generación de señales evocadas. |
| BCI | Interfaz cerebro-máquina. Canal de comunicación que permite a un usuario el control de un dispositivo a partir de las señales cerebrales que genere. |
| Biometría | Estudio de métodos automáticos para el reconocimiento distintivo de humanos basados en sus rasgos o conductas. |
| Biónica/o/s | Aparatos o máquinas que funcionan de acuerdo a principios observados en seres vivos. La biónica sería la ciencia que estudia el desarrollo de estos aparatos. |
| Cola | En programación, una cola es una estructura de datos en el cual solo podemos acceder al primer y al último elemento que la componen. Puede utilizarse para enviar información entre distintos procesos; uno de los procesos es el que introduce los últimos datos y otro de los procesos el que lee los primeros datos introducidos previamente. |
| Desplazamiento de Ventanas | En procesado de señal, el desplazamiento indica el número de muestras que separan el comienzo de una ventana y el comienzo de la siguiente ventana de análisis. |
| Electrodo | Conductor eléctrico utilizado para hacer contacto con una parte no metálica del circuito. En el caso de los cascos de electroencefalografía, los electrodos sirven de contacto entre el circuito y la cabeza del usuario. |
| Electroencefalografía | Exploración neurofisiológica que se basa en el registro de la actividad bioeléctrica cerebral. |

| | |
|---|--|
| Enventanado de señal | Enventanar consiste en limitar una señal continua en el tiempo para su análisis. A dicha señal acotada en el tiempo se suele aplicar una función para limitar los efectos y distorsiones derivados de haber acotado una señal continua. |
| Estacionario / estacionaridad | Se dice que un sistema es estacionario cuando las características del mismo no varían con el tiempo. Por el contrario, un sistema no estacionario será aquel que varíe con el tiempo. |
| Estímulo / Estímulos sensoriales | Un estímulo es una señal externa o interna capaz de provocar una reacción en un sistema. En el caso de los estímulos sensoriales se tratará de estímulos percibidos por los sentidos que provocan una reacción en el organismo. |
| Eye tracking | Técnica para evaluar el punto de fijación de la mirada o el movimiento del ojo en relación al resto de la cabeza. |
| Feedback | Realimentación que en un sistema provoca que cierta proporción de la salida del mismo sea redirigido a su entrada, con el objetivo de controlar su comportamiento. |
| FIFO | Siglas de <i>First In First Out</i> . Concepto utilizado en estructuras de datos y teoría de colas que se refiere a que el primer dato en entrar a una estructura debe ser el primero en salir de la misma. |
| Filtrado de señal | Procesamiento matemático sobre la señal que permite la atenuación de componentes no deseados de la misma. |
| Fork | En programación en Linux es una instrucción que permite que un proceso genere otro proceso hijo, permitiendo que un único programa cuente con varios procesos para atender de forma simultánea distintas tareas. |
| Frecuencia | Magnitud que mide el número de repeticiones de un evento por unidad de tiempo de cualquier fenómeno periódico. |
| ID | Identificador o etiqueta que sirve para designar algo inequívocamente. Las colas de comunicación entre procesos tienen una ID que permite diferenciarlas del resto de estructuras que esté gestionando el sistema operativo. |
| Latencia | Retardos temporales. |
| Lateralidad | Preferencia que muestran la mayoría de los seres humanos por un lado de su propio cuerpo. |
| LED | Diodo emisor de luz. |
| Librerías de programación | Las librerías o bibliotecas constituyen un conjunto de implementaciones funcionales codificadas en un lenguaje de programación determinado y que proporcionan herramientas y funciones para la funcionalidad requerida. Las librerías gráficas proporcionan utilidades para representar objetos en pantalla. |
| Linux | Sistema operativo abierto, gratuito y disponible para que cualquier persona pueda usarlo, modificarlo, estudiarlo y redistribuirlo. |

- MATLAB** Programa de cálculo numérico diseñado para trabajar con matrices.
- Memoria compartida** Recurso utilizado en programación. Es el tipo de memoria que puede ser accedida por múltiples procesos, por ejemplo, para comunicarse entre ellos.
- Occipital** La parte occipital del cráneo constituye el límite posterior del mismo.
- Oxihemoglobina** La hemoglobina es una heteroproteína de la sangre que transporta el oxígeno desde los órganos respiratorios hasta los tejidos y el dióxido de carbono en recorrido inverso. La oxihemoglobina es la hemoglobina oxigenada.
- Refresco de pantalla** Frecuencia a la que las imágenes que se muestran por una pantalla cambian de una a la siguiente con el objetivo de crear la ilusión de una imagen en movimiento.
- Ruido** En el análisis y procesamiento de señales se considera ruido a todos aquellos componentes no deseados dentro de la señal objetivo para la realización de un determinado análisis.
- SDK** Se trata de un kit de desarrollo software que engloba un conjunto de herramientas que permite al programador crear aplicaciones para un sistema o dispositivo concreto.
- Solapamiento de ventanas** En el procesado de señal, el solapamiento de ventanas se refiere al número de muestras que una ventana y la anterior comparten. Una ventana de análisis puede comenzar antes de que la ventana anterior haya llegado a su fin. Ese conjunto de muestras que estarán registradas y se analizarán en ambas ventanas constituye el solapamiento.
- SSVEP** Potenciales evocados gracias a estimulación visual y que se caracterizan por tener una frecuencia similar a la frecuencia de oscilación del estímulo. Se generan en la parte occipital del cerebro.
- Subarmónicos** Componentes de una señal cuya frecuencia es un divisor de la fundamental.
- Timer y timeout** En programación un *timer* es aquella función que permite el control del paso del tiempo. *Timeout* es una señal generada cuando un *timer* llega a un valor determinado, a modo de alarma.
- Transformada de Fourier** Transformación matemática que permite transformar señales entre el dominio del tiempo y el dominio de la frecuencia. Es reversible.
- Tuberías** Las tuberías son un recurso de programación que permiten conectar una salida de un proceso con la entrada de otro proceso. Suelen ser de un solo sentido, siendo uno de los procesos el que envía la información y otro el que la recibe.

ANEXO I – PROGRAMA PRINCIPAL

El programa principal encargado de llevar la orquestación de la ejecución está compuesto por los siguientes archivos:

- **Main.cpp:** Este archivo solo contiene la función iniciadora del programa, y las sucesivas llamadas al resto de funciones según las necesidades del usuario. Además será la encargada de crear y gestionar los procesos que llamen a la interfaz gráfica estimuladora.
- **Interfaz.cpp e Interfaz.h:** En estos archivos se encuentran las definiciones y código de las funciones de las distintas opciones que se proporcionan al usuario. También están escritos los métodos de gestión de información del casco, gestión de colas de información entre procesos y llamada a las funciones de análisis de señal.
- **Preprocesado.cpp y preprocesado.h:** En estos archivos se encuentran todas las funciones relacionadas con el procesamiento de señal: inventanados, eliminación de características lineales, transformada de Fourier, operaciones entre canales... También se encuentran las funciones matemáticas necesarias para llevar a cabo el algoritmo ACL.

El siguiente esquema resume el funcionamiento de main.cpp:

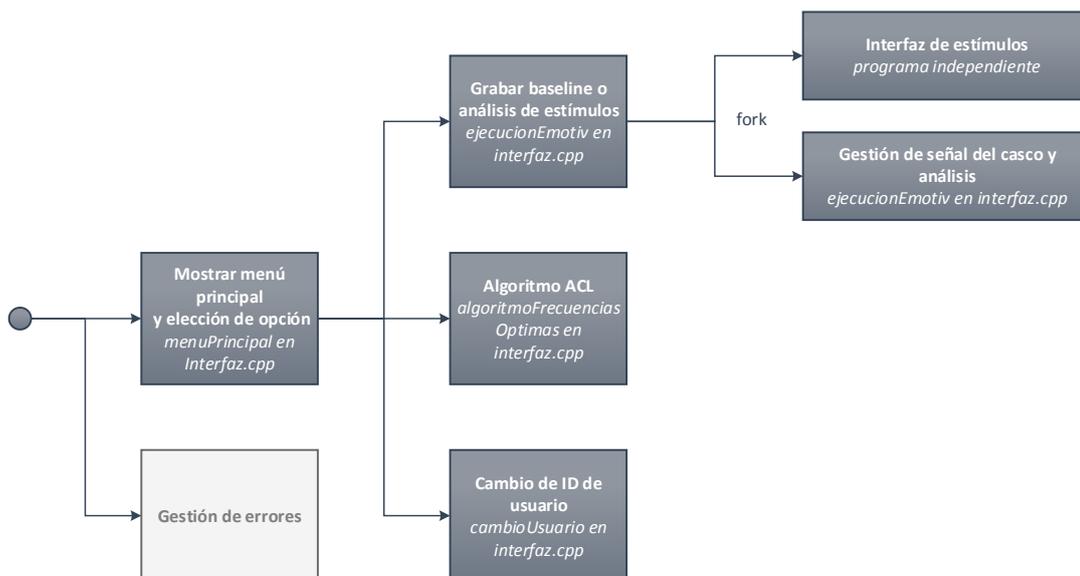


Figura A1-1: Esquema general de funcionamiento y dependencia de funciones en main.cpp

El siguiente esquema resume el funcionamiento de `interfaz.cpp`:

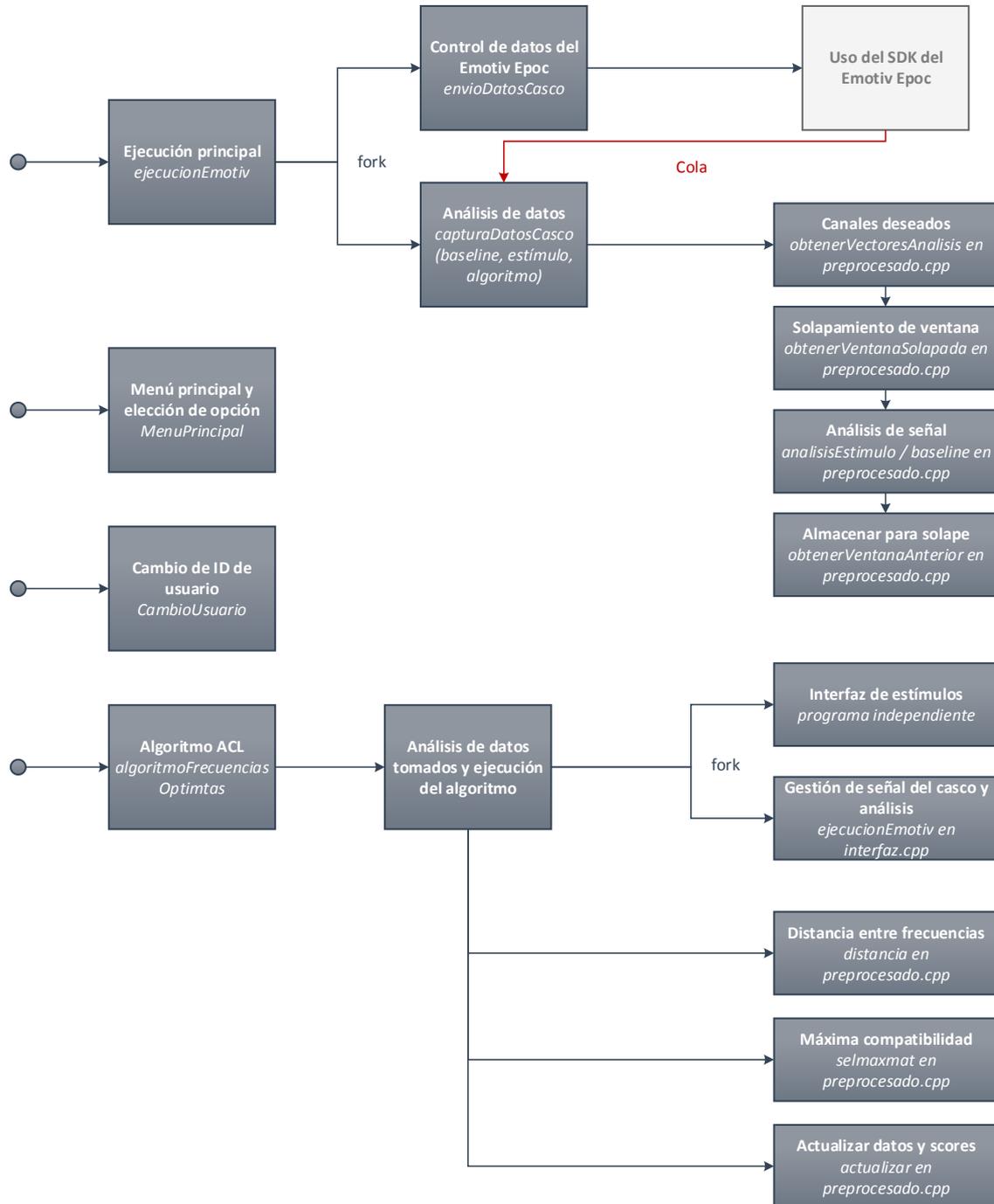


Figura A1-2: Esquema general de funcionamiento y dependencia de funciones en `interfaz.cpp`.

En `preprocesado.cpp` no existe gran dependencia entre funciones dado que todas ellas sirven, en general, para propósitos muy particulares y suelen ser accedidas desde otros módulos, típicamente desde funciones propias de `interfaz.cpp`. Solo para la parte de procesamiento de señal puede ser útil dejar claro cuál será el flujo del programa:

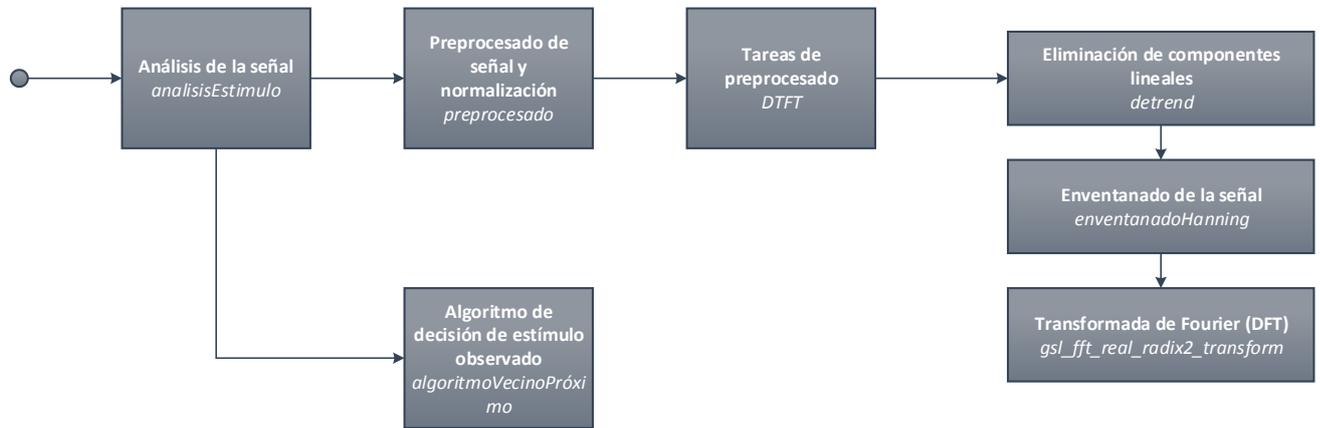


Figura A1-3: Esquema general de la dependencia entre funciones para la realización del procesamiento de señal cuando se está realizando estimulación. Para el caso del baseline el esquema sería idéntico obviando la parte del algoritmo de decisión de estímulo observado.

ANEXO II – INSTALACIÓN Y LIBRERÍAS

Tanto el programa principal como la interfaz gráfica necesitan de una serie de librerías específicas para su correcto funcionamiento. Estas librerías deben estar correctamente instalada en el ordenador en el que se va a hacer uso de la interfaz.

- **Librerías para el control del casco:** Son las librerías proporcionadas por el SDK del Emotiv Epoc. Nos permitirán acceder a los datos brutos proporcionados por el casco. Una serie de carpetas son creadas en el sistema al instalar la edición del software de Emotiv para desarrolladores. La ruta de las librerías para desarrolladores estarán en la subcarpeta *lib* en el directorio en el que se ha realizado la instalación. Las librerías necesarias son *ledk* y *ledk_utils*.
- **Librerías Qt:** Para que la interfaz gráfica funcione correctamente, una serie de librerías de la biblioteca Qt serán necesarias para la correcta estimulación. En sistemas operativos de tipo Ubuntu estas librerías pueden ser instaladas mediante consola accediendo al repositorio web en el que se encuentran:

```
1. wget
   http://download.qt-project.org/official_releases/qt/5.0/5.0.2/qt-
   linux-opensource-5.0.2-x86-offline.run
2. chmod +x qt-linux-opensource-5.0.2-x86-offline.run
3. ./qt-linux-opensource-5.0.2-x86-offline.run
```

- **Librerías GSL:** Para realizar ciertas operaciones matemáticas el programa hace uso del conjunto del GNU Scientific Library. La forma más inmediata de hacerlo es mediante consola a través del Advanced Packaging Tool proporcionado por Ubuntu:

```
1. sudo apt-get install libgsl0ldb1
2. sudo apt-get install gsl-bin libgsl0-dev
3. sudo apt-get install gsl-doc-info gsl-doc-pdf gsl-ref-html gsl-ref-
   psdoc
```

- **Otras dependencias:** En función de qué tipo de configuración se haya realizado al instalar el sistema operativo es posible que falten algunas dependencias importantes que impidan realizar la compilación e instalación de los archivos que constituyen el programa. Un compilador de C++, así como algunas librerías

OpenGL y utilidades generales serán necesarias. Para asegurarse de que el equipo dispone de todo lo necesario se puede ejecutar mediante consola lo siguiente:

1. `sudo apt-get install build-essential`
2. `sudo apt-get install mesa-common-dev`
3. `sudo apt-get install libglu1-mesa-dev -y`

Si se quiere utilizar el programa con la funcionalidad extra de la cámara web será necesario realizar la instalación de las librerías y utilidades proporcionadas por OpenCV. Al estar utilizando funcionalidades básicas no es necesario instalar las últimas versiones de las librerías que incluyan nuevas funciones más específicas. No se recomienda, por tanto, hacerlo dado que este puede ser un proceso engorroso. La forma más sencilla consiste en instalar las librerías proporcionadas en los repositorios de Ubuntu, aunque no estén actualizadas. Para ello:

1. `sudo apt-get install libopencv-dev`

Para realizar la compilación correcta de este proyecto se proporcionan ficheros .pro preparados para ser ejecutados por consola mediante el comando *qmake*. Qmake es un programa que permite la creación automática de ficheros de tipo Makefile y que es proporcionado a partir de la instalación de las librerías Qt. El uso de qmake es universal y genera ficheros de instalación adaptándose a la plataforma y sistema operativo específico en el que se está corriendo el programa. El contenido de los archivos .pro puede ser relevante pues en él se especifican algunas rutas de librerías que deben estar instaladas en el sistema además de las librerías específicas mencionadas anteriormente:

- **AnalisisSSVEP_EE.pro:** Archivo de instalación del programa principal. Será necesario modificar la ruta de las librerías del Emotiv Epoc en función de la instalación que se haya realizado en el sistema:

```
# Álvaro Morán García
# Interfaz EEG basada en Emotiv Epoc
# Proyecto de Fin de Carrera
#
# Grupo de Neurocomputación biológica
# Universidad Autónoma de Madrid

QT += core gui
greaterThan(QT_MAJOR_VERSION, 4): QT += widgets

CONFIG += console
```

```

CONFIG -= app_bundle
CONFIG += -arch x86

TEMPLATE = app
TARGET = AnalisisSSVEP_EE

# Archivos fuente
SOURCES += main.cpp \
interfazEEG.cpp \
preprocesado.cpp

# Cabeceras
HEADERS += \
EmoStateDLL.h \
edkErrorCode.h \
edk.h \
interfazEEG.h \
preprocesado.h

# Rutas para librerías necesarias (modificar según el directorio de instalación)
LIBS += -L$$/home/Usuario/EmotivResearch_1.0.0.5/lib \
# Librerías para Emotiv
-ledk \
-ledk_utils \
-Incurses \
# Librerías para GSL
-lgsl

```

InterfazGrafica.pro: Archivo de instalación de la interfaz gráfica sin webcam:

```

# Álvaro Morán García
# Interfaz de estimulación para SSVEP
# Proyecto de Fin de Carrera
#
# Grupo de Neurocomputación biológica
# Universidad Autónoma de Madrid

QT += core gui
greaterThan(QT_MAJOR_VERSION, 4): QT += widgets

TARGET = InterfazGrafica
TEMPLATE = app

# Archivos fuente
SOURCES += main.cpp \
ventanaprincipal.cpp \
estimulo.cpp \
interfaz.cpp

# Cabeceras
HEADERS += ventanaprincipal.h \
estimulo.h \

```

```
interfaz.h
```

InterfazGrafica.pro: Archivo de instalación de la interfaz gráfica con webcam:

```
# Álvaro Morán García
# Interfaz de estimulación para SSVEP
# Proyecto de Fin de Carrera
#
# Grupo de Neurocomputación biológica
# Universidad Autónoma de Madrid

QT += core gui
greaterThan(QT_MAJOR_VERSION, 4): QT += widgets

TARGET = InterfazGrafica
TEMPLATE = app

# Archivos fuente
SOURCES += main.cpp \
    ventanaprincipal.cpp \
    estimulo.cpp \
    interfaz.cpp \
    camerawidget.cpp

# Cabeceras
HEADERS += ventanaprincipal.h \
    estimulo.h \
    interfaz.h \
    camerawidget.h

# Librerías OpenCV para la Webcam
LIBS += /usr/local/lib/libopencv_highgui.so \
    /usr/local/lib/libopencv_core.so
```

La instalación de la interfaz gráfica y del programa principal debe realizarse en carpetas independientes. Para ello, en cada una de las carpetas se puede realizar la instalación escribiendo vía consola:

1. | qmake
2. | make

Una vez se hayan creado los ejecutables, deberán moverse hacia una misma ruta y disponer de los archivos necesarios para la ejecución especificados en el apartado 6.2. El esquema general de dependencia de ficheros y rutas es el siguiente, incluyendo aquellos archivos que serán necesarios para la ejecución del programa:

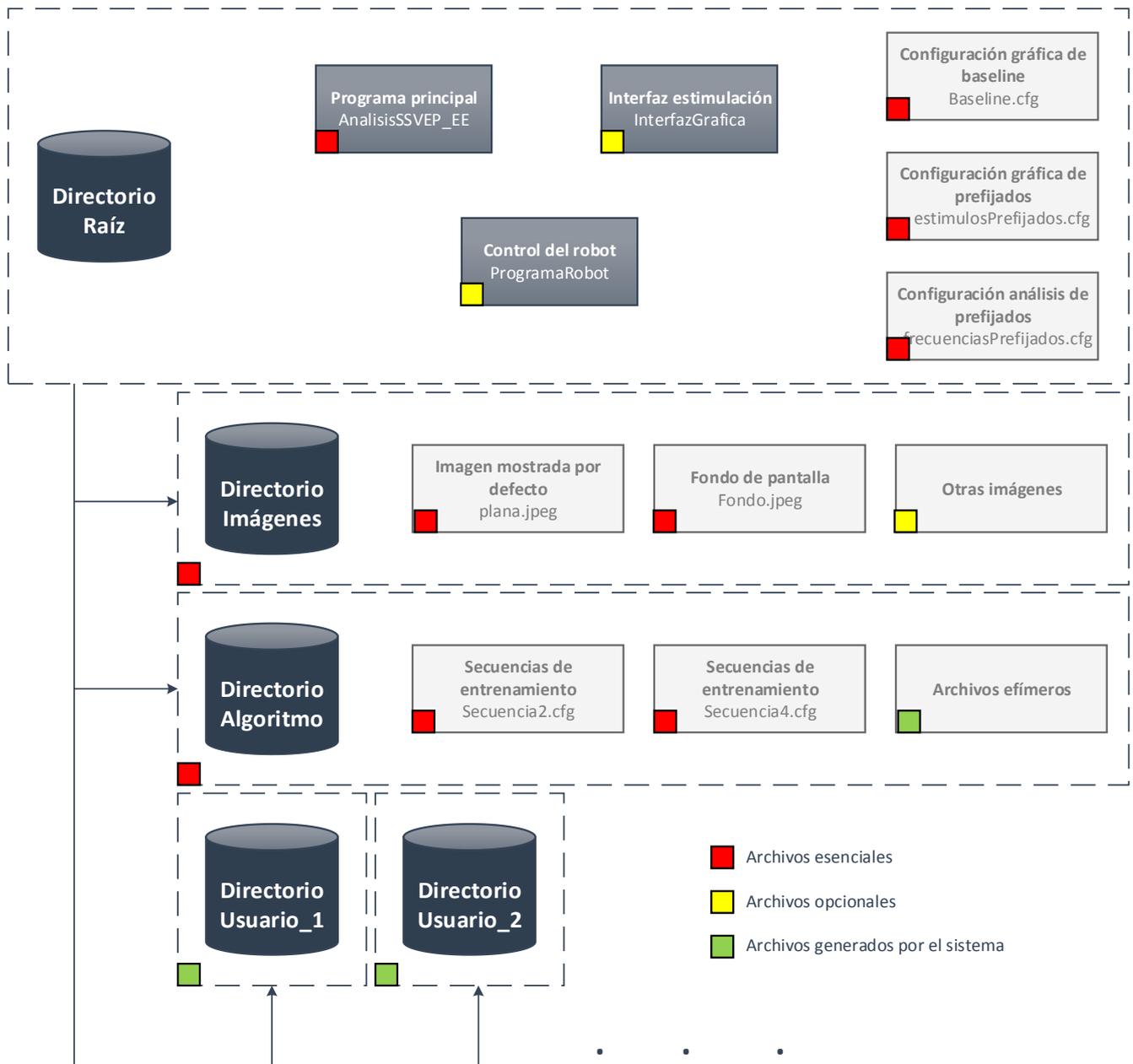


Figura A2-1: Esquema general de organización de los archivos y directorios.

ANEXO III – GUÍA DE USUARIO

Realizada correctamente la instalación se podrá iniciar el programa mediante consola de la siguiente forma:

```
1. | ./AnalisisSSVEP_EE
```

Una vez se inicie el programa se mostrará su interfaz por consola, que permitirá al usuario elegir entre diversas opciones:

```
-----  
MENU PRINCIPAL - ID Usuario: 1  
-----  
Opción 1: Análisis de datos (baseline)  
Opción 2: Análisis de datos (estímulo)  
Opción 3: Búsqueda de frecuencias óptimas  
Opción 4: Análisis de datos con frecuencias prefijadas  
Opción 5: Introducir o cambiar ID de usuario  
Opción 0: Salir  
-----  
Seleccione una: |
```

Figura A3-1: Menú principal del programa.

En la cabecera del menú principal se mostrará siempre la ID del usuario que vaya a realizar el experimento. Las distintas opciones ofrecidas son las siguientes:

- **Opción 1:** Realizar el análisis del *baseline*. Esto supondrá que el usuario tendrá que mantenerse observando la pantalla sin ningún tipo de estímulo durante un período largo de tiempo, recomendado entre 30 y 60 segundos. Esta es la primera opción a realizar por un usuario que va a comenzar una sesión.
- **Opción 2:** Realizar el análisis con estimulación. Esta opción mostrará en pantalla los 4 estímulos en pantalla y se realizará el análisis de los datos que se estén obteniendo desde el casco. Para poder acceder a esta opción debe haber grabado previamente un *baseline* y haber sido entrenado el sistema.
- **Opción 3:** Búsqueda de frecuencias óptimas. Esto constituye el entrenamiento del sistema; se realizará el algoritmo ACL de cara a que la interfaz se adapte lo mejor posible al usuario. Esto es un requisito indispensable para poder acceder

a la opción 2. Para realizar el entrenamiento es necesario que el *baseline* esté grabado de antemano.

- **Opción 4:** Análisis de los datos con estimulación haciendo uso de frecuencias prefijadas. Para poder acceder a esta opción es necesario haber grabado previamente el *baseline*.
- **Opción 5:** Este apartado nos permitirá cambiar la ID de usuario, de cara a que los datos de cada uno de los usuarios no sean sobrescritos cada vez que se inicia una nueva sesión en el sistema.

Para realizar una experimentación con un nuevo usuario lo primero será siempre realizar la grabación del *baseline*. El siguiente paso recomendado será acceder a la opción 3 para que el sistema sea entrenado de acuerdo a las características propias de la persona. Además, en esta opción se permite configurar como queremos que se vea la interfaz estimuladora, definiendo tamaños de objetos parpadeantes, distancia entre ellos...

```
[INFO] Configuración de posición de los estímulos

Introduzca la separación de los estímulos respecto del marco izquierdo de la pantalla (pixels): 100
Introduzca la separación de los estímulos respecto del marco superior de la pantalla (pixels): 100
Introduzca la separación horizontal entre estímulos (pixels): 100
Introduzca la separación vertical entre estímulos (pixels): 100
Introduzca el tamaño del lado del estímulo (cuadrados): 100
```

Figura A3-2: Opciones de configuración de la interfaz gráfica cuando se está comenzando a entrenar el sistema.

Definidas las características de la estimulación gráfica comenzará el proceso de entrenamiento del sistema. Una vez se haya terminado ya se podrá acceder a la opción de estimulación del sistema (opción 2 del menú principal).

La opción 4 del menú con estímulos prefijados no necesita de entrenamiento previo por parte del usuario. No obstante, el archivo del que se toman los datos que configuran la interfaz gráfica está ya generado y es posible que no se ajuste a las características de la pantalla. Es conveniente modificar este archivo de nombre `estimulosPrefijados.cfg` para que se ajuste a las características de la estimulación deseada, de ser necesario.

PRESUPUESTO

| | Coste |
|--|----------------|
| Hardware | |
| Ordenador | 600 € |
| Pantalla | 150 € |
| Casco de EEG – Emotiv Epoc | 320 € |
| Fotodiodo y acondicionamiento | 20 € |
| Software | |
| SDK para el Emotiv Epoc | 250 € |
| Gestión y monitorización de sensores (fotodiodo) | 50 € |
| Licencia de Emotiv para uso educativo | 1440 € |
| Ejecución material y honorarios | |
| Subtotal de ejecución material (hardware y software) | 2830 € |
| Gastos generales (16% del subtotal) | 453 € |
| Beneficio industrial (6% del subtotal) | 170 € |
| Honorarios (700 horas a 15€/hora) | 10.500 € |
| Material fungible | |
| Agua ionizada | 10 € |
| Electrodos para el casco captador (x2) | 100 € |
| Fotocopias | 10 € |
| Gastos de impresión | 120 € |
| Encuadernaciones | 50 € |
| Totales | |
| Subtotal del presupuesto | 13953 € |
| I.V.A. Aplicable (21%) | 2931 € |
| Total presupuesto | 16884 € |

Madrid, Enero de 2015
El ingeniero jefe de proyecto

Fdo.: Álvaro Morán García
Ingeniero de Telecomunicación

PLIEGO DE CONDICIONES

Este documento contiene las condiciones legales que guiarán la realización, en este proyecto, de una Interfaz cerebro-máquina controlada mediante registros de EEG. En lo que sigue, se supondrá que el proyecto ha sido encargado por una empresa cliente a una empresa consultora con la finalidad de realizar dicho sistema. Dicha empresa ha debido desarrollar una línea de investigación con objeto de elaborar el proyecto. Esta línea de investigación, junto con el posterior desarrollo de los programas está amparada por las condiciones particulares del siguiente pliego.

Supuesto que la utilización industrial de los métodos recogidos en el presente proyecto ha sido decidida por parte de la empresa cliente o de otras, la obra a realizar se regulará por las siguientes:

Condiciones generales

1. La modalidad de contratación será el concurso. La adjudicación se hará, por tanto, a la proposición más favorable sin atender exclusivamente al valor económico, dependiendo de las mayores garantías ofrecidas. La empresa que somete el proyecto a concurso se reserva el derecho a declararlo desierto.

2. El montaje y mecanización completa de los equipos que intervengan será realizado totalmente por la empresa licitadora.

3. En la oferta, se hará constar el precio total por el que se compromete a realizar la obra y el tanto por ciento de baja que supone este precio en relación con un importe límite si este se hubiera fijado.

4. La obra se realizará bajo la dirección técnica de un Ingeniero Superior de Telecomunicación, auxiliado por el número de Ingenieros Técnicos y Programadores que se estime preciso para el desarrollo de la misma.

5. Aparte del Ingeniero Director, el contratista tendrá derecho a contratar al resto del personal, pudiendo ceder esta prerrogativa a favor del Ingeniero Director, quien no estará obligado a aceptarla.

6. El contratista tiene derecho a sacar copias a su costa de los planos, pliego de condiciones y presupuestos. El Ingeniero autor del proyecto autorizará con su firma las copias solicitadas por el contratista después de confrontarlas.

7. Se abonará al contratista la obra que realmente ejecute con sujeción al proyecto que sirvió de base para la contratación, a las modificaciones autorizadas por la superioridad o a las órdenes que con arreglo a sus facultades le hayan comunicado por escrito al Ingeniero Director de obras siempre que dicha obra se haya ajustado a los preceptos de los pliegos de condiciones,

con arreglo a los cuales, se harán las modificaciones y la valoración de las diversas unidades sin que el importe total pueda exceder de los presupuestos aprobados. Por consiguiente, el número de unidades que se consignan en el proyecto o en el presupuesto, no podrá servirle de fundamento para entablar reclamaciones de ninguna clase, salvo en los casos de rescisión.

8. Tanto en las certificaciones de obras como en la liquidación final, se abonarán los trabajos realizados por el contratista a los precios de ejecución material que figuran en el presupuesto para cada unidad de la obra.

9. Si excepcionalmente se hubiera ejecutado algún trabajo que no se ajustase a las condiciones de la contrata pero que sin embargo es admisible a juicio del Ingeniero Director de obras, se dará conocimiento a la Dirección, proponiendo a la vez la rebaja de precios que el Ingeniero estime justa y si la Dirección resolviera aceptar la obra, quedará el contratista obligado a conformarse con la rebaja acordada.

10. Cuando se juzgue necesario emplear materiales o ejecutar obras que no figuren en el presupuesto de la contrata, se evaluará su importe a los precios asignados a otras obras o materiales análogos si los hubiere y cuando no, se discutirán entre el Ingeniero Director y el contratista, sometiéndolos a la aprobación de la Dirección. Los nuevos precios convenidos por uno u otro procedimiento, se sujetarán siempre al establecido en el punto anterior.

11. Cuando el contratista, con autorización del Ingeniero Director de obras, emplee materiales de calidad más elevada o de mayores dimensiones de lo estipulado en el proyecto, o sustituya una clase de fabricación por otra que tenga asignado mayor precio o ejecute con mayores dimensiones cualquier otra parte de las obras, o en general, introduzca en ellas cualquier modificación que sea beneficiosa a juicio del Ingeniero Director de obras, no tendrá derecho sin embargo, sino a lo que le correspondería si hubiera realizado la obra con estricta sujeción a lo proyectado y contratado.

12. Las cantidades calculadas para obras accesorias, aunque figuren por partida alzada en el presupuesto final (general), no serán abonadas sino a los precios de la contrata, según las condiciones de la misma y los proyectos particulares que para ellas se formen, o en su defecto, por lo que resulte de su medición final.

13. El contratista queda obligado a abonar al Ingeniero autor del proyecto y director de obras así como a los Ingenieros Técnicos, el importe de sus respectivos honorarios facultativos por formación del proyecto, dirección técnica y administración en su caso, con arreglo a las tarifas y honorarios vigentes.

14. Concluida la ejecución de la obra, será reconocida por el Ingeniero Director que a tal efecto designe la empresa.

15. La garantía definitiva será del 4% del presupuesto y la provisional del 2%.

16. La forma de pago será por certificaciones mensuales de la obra ejecutada, de acuerdo con los precios del presupuesto, deducida la baja si la hubiera.

17. La fecha de comienzo de las obras será a partir de los 15 días naturales del replanteo oficial de las mismas y la definitiva, al año de haber ejecutado la provisional, procediéndose si no existe reclamación alguna, a la reclamación de la fianza.

18. Si el contratista al efectuar el replanteo, observase algún error en el proyecto, deberá comunicarlo en el plazo de quince días al Ingeniero Director de obras, pues transcurrido ese plazo será responsable de la exactitud del proyecto.

19. El contratista está obligado a designar una persona responsable que se entenderá con el Ingeniero Director de obras, o con el delegado que éste designe, para todo relacionado con ella. Al ser el Ingeniero Director de obras el que interpreta el proyecto, el contratista deberá consultarle cualquier duda que surja en su realización.

20. Durante la realización de la obra, se girarán visitas de inspección por personal facultativo de la empresa cliente, para hacer las comprobaciones que se crean oportunas. Es obligación del contratista, la conservación de la obra ya ejecutada hasta la recepción de la misma, por lo que el deterioro parcial o total de ella, aunque sea por agentes atmosféricos u otras causas, deberá ser reparado o reconstruido por su cuenta.

21. El contratista, deberá realizar la obra en el plazo mencionado a partir de la fecha del contrato, incurriendo en multa, por retraso de la ejecución siempre que éste no sea debido a causas de fuerza mayor. A la terminación de la obra, se hará una recepción provisional previo reconocimiento y examen por la dirección técnica, el depositario de efectos, el interventor y el jefe de servicio o un representante, estampando su conformidad el contratista.

22. Hecha la recepción provisional, se certificará al contratista el resto de la obra, reservándose la administración el importe de los gastos de conservación de la misma hasta su recepción definitiva y la fianza durante el tiempo señalado como plazo de garantía. La recepción definitiva se hará en las mismas condiciones que la provisional, extendiéndose el acta correspondiente. El Director Técnico propondrá a la Junta Económica la devolución de la fianza al contratista de acuerdo con las condiciones económicas legales establecidas.

23. Las tarifas para la determinación de honorarios, reguladas por orden de la Presidencia del Gobierno el 19 de Octubre de 1961, se aplicarán sobre el denominado en la actualidad "Presupuesto de Ejecución de Contrata" y anteriormente llamado "Presupuesto de Ejecución Material" que hoy designa otro concepto.

Condiciones particulares

La empresa consultora, que ha desarrollado el presente proyecto, lo entregará a la empresa cliente bajo las condiciones generales ya formuladas, debiendo añadirse las siguientes condiciones particulares:

1. La propiedad intelectual de los procesos descritos y analizados en el presente trabajo, pertenece por entero a la empresa consultora representada por el Ingeniero Director del Proyecto.

2. La empresa consultora se reserva el derecho a la utilización total o parcial de los resultados de la investigación realizada para desarrollar el siguiente proyecto, bien para su

publicación o bien para su uso en trabajos o proyectos posteriores, para la misma empresa cliente o para otra.

3. Cualquier tipo de reproducción aparte de las reseñadas en las condiciones generales, bien sea para uso particular de la empresa cliente, o para cualquier otra aplicación, contará con autorización expresa y por escrito del Ingeniero Director del Proyecto, que actuará en representación de la empresa consultora.

4. En la autorización se ha de hacer constar la aplicación a que se destinan sus reproducciones así como su cantidad.

5. En todas las reproducciones se indicará su procedencia, explicitando el nombre del proyecto, nombre del Ingeniero Director y de la empresa consultora.

6. Si el proyecto pasa la etapa de desarrollo, cualquier modificación que se realice sobre él, deberá ser notificada al Ingeniero Director del Proyecto y a criterio de éste, la empresa consultora decidirá aceptar o no la modificación propuesta.

7. Si la modificación se acepta, la empresa consultora se hará responsable al mismo nivel que el proyecto inicial del que resulta el añadirla.

8. Si la modificación no es aceptada, por el contrario, la empresa consultora declinará toda responsabilidad que se derive de la aplicación o influencia de la misma.

9. Si la empresa cliente decide desarrollar industrialmente uno o varios productos en los que resulte parcial o totalmente aplicable el estudio de este proyecto, deberá comunicarlo a la empresa consultora.

10. La empresa consultora no se responsabiliza de los efectos laterales que se puedan producir en el momento en que se utilice la herramienta objeto del presente proyecto para la realización de otras aplicaciones.

11. La empresa consultora tendrá prioridad respecto a otras en la elaboración de los proyectos auxiliares que fuese necesario desarrollar para dicha aplicación industrial, siempre que no haga explícita renuncia a este hecho. En este caso, deberá autorizar expresamente los proyectos presentados por otros.

12. El Ingeniero Director del presente proyecto, será el responsable de la dirección de la aplicación industrial siempre que la empresa consultora lo estime oportuno. En caso contrario, la persona designada deberá contar con la autorización del mismo, quien delegará en él las responsabilidades que ostente.