

2016

DESARROLLO DE SISTEMA DE TRANSMISIÓN Y DESPLIEGUE DE SEÑALES NEURONALES PARA EL DIAGNÓSTICO MÉDICO REMOTO

FERNÁNDEZ PINO, FELIPE ANDRÉS

<http://hdl.handle.net/11673/23350>

Repositorio Digital USM, UNIVERSIDAD TECNICA FEDERICO SANTA MARIA

UNIVERSIDAD TÉCNICA FEDERICO SANTA
MARÍA
DEPARTAMENTO DE INGENIERÍA ELECTRÓNICA
VALPARAÍSO – CHILE



“DESARROLLO DE SISTEMA DE
TRANSMISIÓN Y DESPLIEGUE DE SEÑALES
NEURONALES PARA EL DIAGNÓSTICO
MÉDICO REMOTO”

FELIPE ANDRÉS FERNÁNDEZ PINO

MEMORIA DE TITULACIÓN PARA OPTAR AL TÍTULO DE INGENIERÍA
CIVIL ELECTRÓNICA MENCIÓN COMPUTADORES

PROFESOR GUÍA: DR. AGUSTÍN GONZÁLEZ V.

PROFESOR CORREFERENTE: DR. MATÍAS ZAÑARTU S.

DICIEMBRE - 2016

Agradecimientos

En el transcurso de mi carrera universitaria, muchas personas están y estuvieron presente en mi vida; brindándome su apoyo, sus consejos, buenas energías e intenciones. Todas ellas fueron recibidas de la mejor forma, y eso me otorgó la fuerza para poder hacer de este difícil, pero bonito período, una bonita experiencia. De manera general, estoy muy agradecido de todos sus actos de afecto.

En primer lugar, quisiera agradecer a mis padres y a toda mi familia que siempre me han apoyado incondicionalmente y han creído en mí, ayudándome a sacar adelante cada uno de los desafíos propuestos. Es sin duda a ellos a quien debo todos mis éxitos, puesto que sin las enseñanzas sobre respeto, valores y educación que me dieron no hubiese llegado a estas instancias.

Agradecer también a todos mis amigos que me han acompañado y han logrado hacer de esta etapa universitaria un periodo más grato y alegre, apoyandome en los estudios y en las instancias de celebración.

Contenido

Agradecimientos	2
Contenido	3
Índice de figuras.....	4
Índice de tablas	6
Resumen.....	7
Abstract	8
Glosario.....	9
Capítulo 1: Introducción	10
Capítulo 2: El electroencefalograma en la actualidad	13
Capítulo 3: Análisis del sistema EEGRemote	17
3.1 Requerimientos del sistema.....	17
3.2 Estimación de la tasa de datos generada por el sensor EEG	18
Capítulo 4: Diseño de la solución	20
4.1 Alternativas de solución	20
4.2 Diseño de la arquitectura del sistema EEGRemote.....	22
4.3 Protocolo de comunicación	27
Capítulo 5: Implementación del sistema EEGRemote	37
Capítulo 6: Evaluación del sistema EEGRemote	40
6.1 Despliegue local con y sin emulador.....	40
6.2 Despliegue remoto con y sin emulador	47
Capítulo 7: Conclusiones y trabajo futuro	56
Referencias.....	58

Índice de figuras

Figura 1-1: Arquitectura del sistema final buscado.	11
Figura 1-2: Sistema de captura, transmisión y despliegue de señales neurológicas propuesto: a) Aplicación cliente-servidor de captura; b) Aplicación cliente-servidor de despliegue.	12
Figura 4-1: Arquitectura solución Sistema de conexión directa sensor-visualizador.	20
Figura 4-2: Arquitectura solución Sistema con servidor de almacenamiento y re- envío en Internet.	21
Figura 4-3: Módulos del sistema.....	23
Figura 4-4: Arquitectura del módulo EEGClient.....	23
Figura 4-5: Arquitectura del módulo EEGServer.	25
Figura 4-6: Arquitectura del módulo EEGViewer.....	27
Figura 4-7: Estructura general de paquetes.....	29
Figura 4-8: Estructura del paquete identificador de canales.....	29
Figura 4-9: Estructura del paquete de muestras capturadas.....	30
Figura 4-10: Estructura del paquete de eventos.....	31
Figura 4-11: Estructura del paquete de solicitud de nueva conexión.	32
Figura 4-12: Estructura del paquete de notificación de nueva fuente.....	33
Figura 4-13: Estructura del paquete de fuentes disponibles.	33
Figura 4-14: Secuencia seguida en la transmisión de tiempo real.	35
Figura 4-15: Secuencia seguida en la transmisión diferida.	36

Figura 5-1: Distribución de electrodos en a) Emotiv EPOC y b) Stat X24.	39
Figura 6-1: Despliegue local con electroencefalógrafo emulado, paso 1: a) Aplicación EEGClient; b) Aplicación EEGServer.....	41
Figura 6-2: Despliegue local con electroencefalógrafo emulado, paso 2: Aplicación EEGClient.....	42
Figura 6-3: Despliegue local con electroencefalógrafo emulado, paso 3: a) Aplicación EEGClient; b) Aplicación EEGServer.....	42
Figura 6-4: Despliegue local con electroencefalógrafo emulado, paso 4: Aplicación EEGClient.....	43
Figura 6-5: Despliegue local con electroencefalógrafo emulado, paso 5: a) Aplicación EEGClient; b) Aplicación EEGServer.....	43
Figura 6-6: Despliegue local usando electroencefalógrafo EPOC, paso 1: a) Aplicación EEGClient; b) Aplicación EEGServer.....	44
Figura 6-7: Despliegue local usando electroencefalógrafo EPOC, paso 2: Aplicación EEGClient.....	45
Figura 6-8: Despliegue local usando electroencefalógrafo EPOC, paso 3: a) Aplicación EEGClient; b) Aplicación EEGServer.....	45
Figura 6-9: Despliegue local usando electroencefalógrafo EPOC, paso 4: Aplicación EEGClient.....	46
Figura 6-10: Despliegue local usando electroencefalógrafo EPOC, paso 5: a) Aplicación EEGClient; b) Aplicación EEGServer.	46
Figura 6-11: Despliegue remoto con electroencefalógrafo emulado, paso 2: a) Aplicación EEGClient; b) Aplicación EEGViewer.....	48
Figura 6-12: Despliegue remoto con electroencefalógrafo emulado, paso 3: Aplicación EEGViewer.....	49

Figura 6-13: Despliegue remoto con electroencefalógrafo emulado, paso 4: a) Aplicación EEGClient; b) Aplicación EEGViewer.....	49
Figura 6-14: Despliegue remoto con electroencefalógrafo emulado, paso 5: a) Aplicación EEGClient; b) Aplicación EEGViewer.....	50
Figura 6-15: Despliegue remoto usando electroencefalógrafo EPOC, paso 2: Aplicación EEGViewer.	51
Figura 6-16: Despliegue remoto usando electroencefalógrafo EPOC, paso 3: Aplicación EEGViewer.	52
Figura 6-17: Despliegue remoto usando electroencefalógrafo EPOC, paso 4: a) Aplicación EEGClient; b) Aplicación EEGViewer.....	52
Figura 6-18: Despliegue remoto usando electroencefalógrafo EPOC, paso 5: a) Aplicación EEGClient; b) Aplicación EEGViewer.....	53

Índice de tablas

Tabla 4-1: Comparación de alternativas de arquitectura.	21
Tabla 4-2: Lista de posibles eventos o requerimientos.....	31
Tabla 6-1: Secuencia de pasos para las pruebas de despliegue local	40
Tabla 6-2: Secuencia de pasos para las pruebas de despliegue remoto	47
Tabla 6-3: Velocidades de subida y bajada de cada aplicación.....	54
Tabla 6-4: Parámetros del emulador desarrollado	54

Desarrollo de sistema de transmisión y despliegue de señales neuronales para el diagnóstico médico remoto

Memoria para optar al título de Ingeniero Civil Electrónico, mención Computadores

Felipe Andrés Fernández Pino

Profesor Guía: Agustín González

Diciembre 2016

Resumen

En la actualidad uno de los exámenes más utilizados en el área de la neurología es el electroencefalograma (EEG) debido a su bajo costo y su utilidad en el estudio de variados trastornos y enfermedades neurológicas. Sin embargo, diversas enfermedades (entre ellas la Epilepsia) son difíciles de diagnosticar si no se logra realizar el examen en el momento en que se presentan los principales síntomas.

Considerando esta problemática, el objetivo principal del presente trabajo es desarrollar un sistema capaz de capturar señales neurológicas provenientes de un electroencefalógrafo, para luego transmitir y desplegarlas de manera remota con el fin de permitir el diagnóstico neurológico a distancia.

La solución planteada corresponde a un sistema con una arquitectura en base a tres aplicaciones: un cliente capturador de datos para los pacientes, un servidor en Internet y un cliente visualizador de datos para los médicos. La solución también contempla el diseño de un protocolo para la comunicación entre las aplicaciones del sistema. La implementación del sistema diseñado se realiza haciendo uso del *framework* Qt 5.7 para las tres aplicaciones, de la biblioteca gráfica QCustomPlot para el despliegue de los datos en las aplicaciones cliente, y de la biblioteca Emokit para la captura de los datos provenientes del electroencefalógrafo EPOC emotiv.

El resultado obtenido del trabajo corresponde al primer prototipo del sistema que cumple con todos los requerimientos establecidos y las pruebas realizadas.

Palabras claves: Electroencefalograma, Protocolo de comunicación, Diagnóstico neurológico remoto.

Development of transmission and deployment system of neuronal signals for remote medical diagnosis

Report to aim for Electronic Civil Engineer Degree, Computer science mention

Felipe Andrés Fernández Pino

Supervisor: Agustín González

December 2016

Abstract

Nowadays, one of the most used examinations in the neurology area is the Electroencephalography (EEG) due its low cost and utility in the study of a various range of disorders and neurological illness. However, several of this illness, (among them the epilepsy) are difficult to diagnose if is not possible to do an examination in the right moment that is when the main symptoms are presented.

Thus this problem, the main objective of this work is to develop a system capable of capture neurological signals coming from an electroencephalograph and then deploy them in a remote way with the final purpose of a neurological diagnose from distance.

The given solution corresponds to a system with an architecture based on three applications: A data grabber for the patients, an internet server and a data display screen client for the medics. The solution also contemplates the design of a communication protocol among the applications of the system. The implementation of the designed system is done making use of the framework Qt 5.7 for the three applications, of the graphic library QCustomPlot for the deployment of data in the client applications, and of the Emokit library for the capture of data that comes from the electroencephalograph EPOC Emotiv.

The result obtained correspond to a first prototype of the system that fulfills all the requirements a system established.

Keywords: Electroencephalography, communication protocol, Remote Neurological Diagnose.

Glosario

Neurología	Especialidad médica que se ocupa de la prevención, diagnóstico, tratamiento y rehabilitación las enfermedades que involucran al sistema nervioso central, periférico y autónomo.
EEG:	Electroencefalograma. Examen médico en el ámbito neurológico que consiste en el registro de las actividades eléctricas del cerebro producidas por la actividad interneuronal.
Sensor EEG:	Electroencefalógrafo. Instrumento médico con el cuál se realiza el EEG.

Capítulo 1: Introducción

En el contexto de la neurología, uno de los principales exámenes que se utilizan en el diagnóstico neurológico hoy en día es el electroencefalograma (EEG). Esto se debe principalmente a que representa una solución de bajo costo y por la fácil realización ante la necesidad de estudiar, monitorear y diagnosticar enfermedades asociadas a trastornos neurológicos como lo son la Epilepsia^[1], el Alzheimer, el Traumatismo Craneoencefálico (TEC) entre otros.

Sin embargo, en la actualidad el EEG posee ciertas falencias que dificultan la tarea de seguimiento de enfermedades neurológicas debido particularmente a la metodología con la cual se realiza este examen. La elaboración de un EEG implica la utilización de grandes espacios clínicos dedicados para la toma del examen, así como un considerable tiempo de preparación y la manipulación de sofisticados electroencefalógrafos^[2]. Todo esto presenta un problema para los médicos especialistas que están interesados en la toma continua o muy periódica del examen a sus pacientes, o que desean poder contar con los resultados de un EEG aplicado justo en los periodos críticos (donde se manifiestan los principales síntomas de las enfermedades neurológicas), como por ejemplo, durante una convulsión para el caso de los pacientes con Epilepsia.

A partir de las falencias mencionadas, surge la necesidad de contar con una alternativa a los procedimientos tradicionales en la toma de EEG. Esta alternativa debe permitir la realización de un EEG en el lugar del paciente, eliminando así el traslado hacia el centro médico por parte de éste. Por otro lado, la alternativa buscada debe poder ser aplicada sin la necesidad de poseer conocimientos expertos en el tema, por lo que no debe poseer procedimientos de alta complejidad que requiera algún tipo de especialización.

Considerando la necesidad identificada, la solución ideal propuesta radica en un sistema de transmisión de datos vía Internet que le permita a los médicos especialistas obtener en tiempo real los resultados de un EEG aplicado de manera remota. En la

Figura 1-1 se ilustra la arquitectura de esta solución propuesta donde se puede observar que incorpora la posibilidad de utilizar una aplicación móvil o una aplicación de escritorio, tanto para la transmisión de los datos como para la visualización de los mismos.

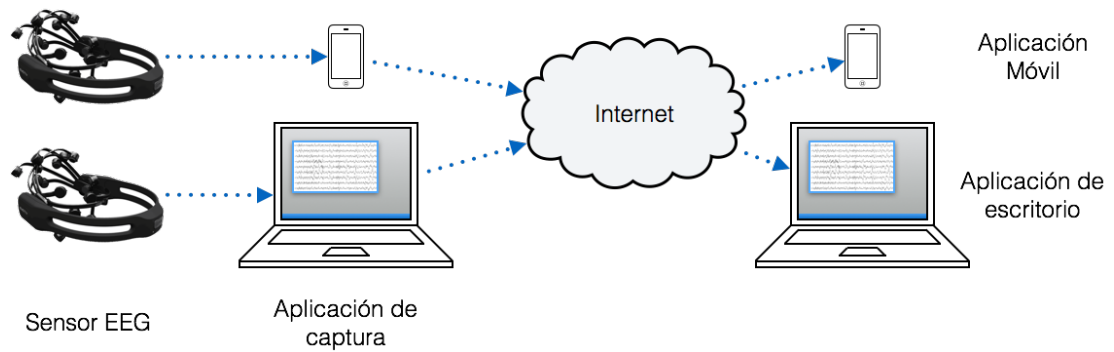
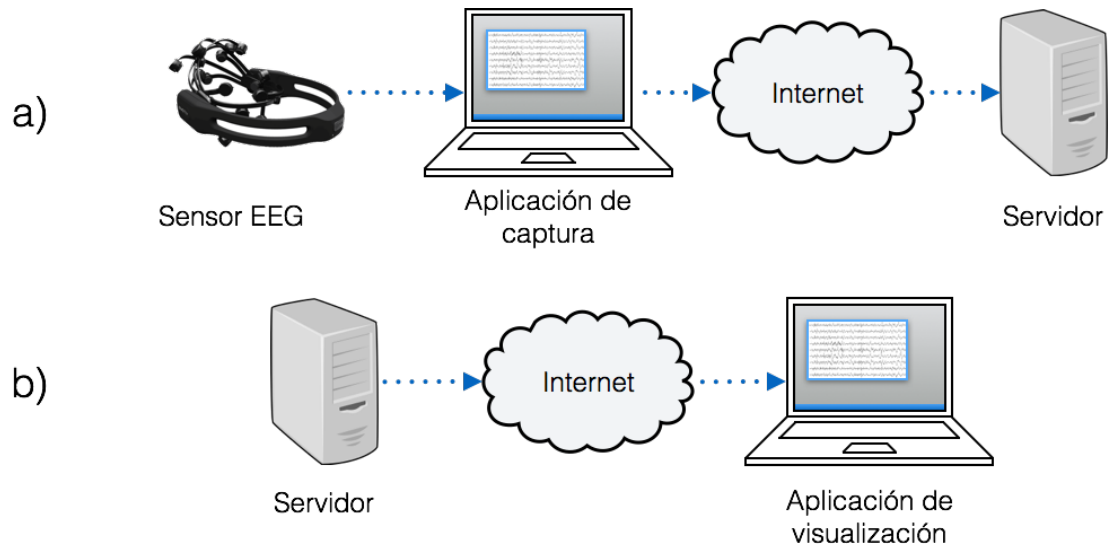


Figura 1-1: Arquitectura del sistema final buscado.

Si bien se presenta la solución ideal, en el presente trabajo se desarrollará la primera iteración encaminada a la solución final, es decir, un primer prototipo que cumpla un primer conjunto de características de la solución final propuesta. Esta primera solución contempla el desarrollo de las aplicaciones de transmisión y visualización en su versión de computador, y al sistema completo se le denomina con un nombre artificial (para facilitar su identificación): Sistema “EEGRemote”.

En resumen; el objetivo principal de este trabajo es el diseño, desarrollo e implementación del primer prototipo del sistema que se ajuste a la primera iteración de la solución antes mencionada. Este objetivo general se desglosa en dos objetivos específicos, por una parte, está el desarrollo de una primera aplicación cliente-servidor. El lado cliente de dicha aplicación se encarga de recibir los datos correspondientes a un sensor EEG (provenientes de un electroencefalógrafo conectado al computador “cliente”) y de transmitirlos vía Internet a un servidor dedicado donde se almacena esta información temporalmente. Por otro lado, está el desarrollo de una segunda aplicación cliente-servidor en la cual el lado cliente de dicha aplicación se encarga de acceder a los datos del servidor (común tanto para la aplicación de captura y la aplicación de visualización) en Internet y de desplegar adecuadamente dichos datos para permitir el

estudio y diagnóstico neurológico de un médico especialista. En la Figura 1-2 se ilustra la arquitectura de la solución contemplada del presente trabajo de memoria.



*Figura 1-2: Sistema de captura, transmisión y despliegue de señales neurológicas propuesto:
a) Aplicación cliente-servidor de captura; b) Aplicación cliente-servidor de despliegue.*

Capítulo 2: El electroencefalograma en la actualidad

Hoy en día, el desarrollo tecnológico en conjunto al avance científico en el área médica, ha permitido el desarrollo de métodos no invasivos en el diagnóstico y control de enfermedades en pacientes con trastornos neurofisiológicos y físicos. Un área de gran relevancia se encuentra en el terreno de la salud neurológica, en donde el análisis de información y diagnóstico de enfermedades es de gran importancia en el tratamiento temprano de pacientes sin la necesidad de requerir a la utilización de metodologías invasivas que puedan poner en riesgo sus vidas.

Uno de los exámenes popularmente utilizados en el área de la neurología es el EEG, el cual registra la actividad cerebral de un individuo, posibilitando diagnósticos *in situ* guiado por un médico especialista en el área. A pesar de las ventajas que esto conlleva, se generan ciertos problemas relacionados a la obtención de reservas de horas médicas para acceder a dichos análisis, en donde el paciente debe acercarse a un centro médico especialista, solicitar el examen, esperar un cupo disponible y nuevamente acercarse al centro para hacer efectiva la solicitud. Esto genera bastantes inconvenientes para la identificación de ciertos trastornos, tales como la Epilepsia, en donde lo ideal sería contar con un EEG al momento de manifestarse los síntomas que evidencian la enfermedad. Un estudio realizado en la Universidad Estatal de Campinas de Brasil ^[2] refleja el impacto que tiene el EEG en el diagnóstico y tratamiento de pacientes con trastornos neurológicos, donde se determinó que, dentro de una muestra de 94 electroencefalogramas, más de la mitad de los EEG fueron realizados después de 24 horas del ingreso de los pacientes (53,3%). Además, reveló que en el 72,2% de los casos el examen implicó un cambio en el tratamiento de los pacientes, reflejando así la importancia que posee el EEG en el diagnóstico de los trastornos neurológicos.

Los tipos de EEG que se aplican en la actualidad corresponden a cuatro: El EEG basal, el EEG en provación de sueño, el EEG de sueño y el EEG de muerte cerebral. Los primeros tres tipos de EEG se utilizan comúnmente para el diagnóstico de enfermedades neurológicas mientras que el EEG de muerte cerebral es utilizado como

parte de un protocolo denominado “protocolo de muerte cerebral” para dar paso al trasplante de organos cuando existe muerte cerebral (bajo consentimiento familiar). Para la realización de un EEG basal se requiere de una breve preparación que consiste en la limpieza del cuero cabelludo para el posicionamiento del electroencefalógrafo. En el caso de un EEG de privación de sueño primero debe transcurrir un periodo de 24 horas en el cual se mantiene despierto al paciente, luego en la consulta se realiza el EEG mientras que el paciente duerme, facilitando así la aparición de trazados fisiológicos de las distintas fases del sueño. Finalmente el EEG de sueño consiste en un EEG convencional en el que el paciente se recuesta en una cama y es grabado en video su actividad al mismo tiempo que se aplica el EEG. La duración es de al menos 3 horas pudiendo extenderse a toda una noche. Su finalidad es tener constancia visual y eléctrica en pacientes de difícil diagnóstico ^[3].

Existen diversos electroencefalógrafos que estan adecuados para ser utilizados en la aplicación de cada tipo de EEG. Los más comunes estan compuestos por varios electrodos que se conectan por cable a una estación en donde se procesan los datos y se pueden visualizar; estos están orientados al uso en consultas donde se aplican los EEG privados de sueño y los EEG de sueño. También existen electroencefalógrafos inalámbricos que le otorgan mayor libertad al paciente dentro de la consulta, sin embargo no poseen un rango de distancia muy amplio. Por otro lado existen electroencefalógrafos que incluyen un computador portátil que permite realizar el exámen en espacios fuera de la consulta o incluso existen algunos que permiten monitorear continuamente al paciente por duraciones largas para el posterior estudio del examen. Finalmente también se pueden hallar sistemas que le permiten al médico recibir y monitorear vía Internet a un paciente que se realiza el examen en una consulta ^[4].

El uso de tecnologías de la información ofrece en este plano gran versatilidad en la manipulación y almacenamiento de datos que, con la instrumentación adecuada, permite rastrear y guardar las señales percibidas en tiempo real por un electroencefalógrafo remoto en un paciente que requiera un estudio detallado de su

condición. Cabe destacar que el avance en tecnologías inalámbricas, que a través de servidores, nubes de datos e Internet permite almacenar de manera confiable la información recibida, pudiendo hacer efectivo su procesamiento en ausencia de un médico especialista y de forma remota, o en caso de identificar condiciones anómalas que presenten riesgos mayores en la vida del paciente, enviar una señal de alarma a algún centro de salud.

Teniendo en cuenta el conjunto de características buscadas en la solución al problema planteado, se hace evidente la necesidad de un instrumento de medición de la actividad interneuronal en el cerebro que permita el registro de esta información de manera rápida, inalámbrica y confiable. Dados los avances tecnológicos actuales se pueden hallar variadas alternativas de estos instrumentos con diferencias considerables en el costo y el número de canales (electrodos) que posee cada uno, reflejando así la necesidad de desarrollar un sistema capaz de registrar los datos obtenidos por cualquiera de los sensores EEG disponibles.

El desarrollo de *software* ha sido una herramienta indispensable para la manipulación de herramientas tecnológicas en conjunto a la visualización de información generada por sensores e instrumentos electrónicos especializados. Esto conlleva al desarrollo de un *Interfaz Hombre Máquina (HMI)* en donde un ingeniero o técnico especialista es capaz de estructurar un programa que permita maximizar el desempeño de la información obtenida por la instrumentación utilizada acorde a las necesidades de un usuario específico.

Los *Kits de Desarrollo de Software (SDK)* son herramientas que le permiten al desarrollador crear aplicaciones orientadas para el uso en sistemas concretos. Estos *Kits* consisten en una *Interfaz de Programación de Aplicaciones (API)* creada para permitir el uso de ciertos lenguajes de programación que puede también incluir *hardware* para comunicarse con ciertos sistemas embebidos. En este plano, cabe destacar que las *API's* representan la capacidad de comunicación entre componentes de un software, en otras palabras, consiste en un conjunto de llamadas a ciertas bibliotecas que ofrecen acceso a ciertos servicios desde los procesos y conseguir abstracción en la programación. Entre

las funciones más utilizadas se destacan el dibujo de ventanas e iconos de pantalla. La gama de *SDK* es variada y obedece a las necesidades buscadas por el usuario, las que pueden ser generalizadas o especializadas, ofreciendo alternativas de código abierto o comerciales para la generación de aplicaciones interactivas de software. Aun así, es necesario considerar los sistemas operativos que el desarrollador utiliza para la creación del programa, acotando las alternativas a utilizar.

Dentro de los *SDK* más populares se destaca el uso de *XCode*, exclusivo para sistema operativo macOS, *Visual Studio* que opera en Sistema operativo Windows y *QT* que puede operar conjuntamente en ambos sistemas e incluso Linux. Estos sistemas ofrecen entornos de desarrollo interactivo que permiten un plano más cómodo de programación de acuerdo a los lenguajes empleados para la creación del software, contando además con el apoyo global de experiencia de desarrolladores alrededor del mundo.

Capítulo 3: Análisis del sistema EEGRemote

En este capítulo, se describe detalladamente un análisis de alto nivel de las características que el sistema EEGRemote debe cumplir para solucionar de manera adecuada el problema identificado.

3.1 Requerimientos del sistema

En base a la necesidad de poder realizar un diagnóstico neurológico de manera remota se identifican los siguientes requerimientos funcionales principales que aseguren una primera solución a dicha necesidad:

- El sistema debe ser independiente de la ubicación:

El sistema EEGRemote debe funcionar con conexiones a través de Internet, para así abarcar todas las ubicaciones en las cuales se disponga de una conexión a la red con una velocidad (de subida y bajada) suficiente para el sistema.

- El sistema debe concentrar la dependencia del electroencefalógrafo:

El sistema EEGRemote debe poseer un único módulo que concentre todas las funcionalidades y características que sean dependientes del electroencefalógrafo utilizado y ofrecer al resto del sistema la información capturada en una interfaz independiente del sensor EEG utilizado. De esta manera se facilita la tarea de modificar el sistema para añadir compatibilidad a distintos electroencefalógrafos.

- El sistema debe permitir múltiples transmisiones en paralelo:

El sistema EEGRemote debe ser desarrollado pensando en la posibilidad de que varios pacientes y varios médicos lo utilicen simultáneamente. Esto conlleva la necesidad de obtener una solución altamente escalable.

- El sistema debe otorgar herramientas para visualizar adecuadamente la información:

La aplicación encargada de la visualización de los datos (aplicación utilizada por los médicos) debe tener las herramientas básicas para la visualización de la información como, por ejemplo, la posibilidad de hacer acercamientos (*zoom*) o la posibilidad de moverse a través de la escala de tiempo entre otras.

- El sistema debe permitir observar la información en tiempo real:

Los datos capturados por el electroencefalógrafo deben poder ser visibles por el médico especialista (de manera remota) en tiempo real y con el menor retardo posible.

- El sistema debe permitir observar información registrada anteriormente:

Se debe contar con la posibilidad de recibir la información registrada de los usuarios (pacientes) en el pasado.

- El sistema debe ser eficiente en el uso de los recursos de ancho de banda:

Dado que el sistema EEGRemote se debe desarrollar pensando en la futura migración a una aplicación móvil, gana importancia el ancho de banda que la aplicación necesita para funcionar, ya que a diferencia de la mayoría de los planes de Internet domiciliario, el Internet para dispositivos móviles es más limitado en la velocidad de transmisión y en la cantidad de datos. Es por esto que se debe buscar obtener una solución que utilice el ancho de banda lo más eficiente posible.

3.2 Estimación de la tasa de datos generada por el sensor EEG

Teniendo en consideración el último de los requerimientos mencionados (aquel que se refiere al uso de ancho de banda) se hace evidente la necesidad de poder contar con una base teórica que detalle cuál es la tasa de datos que genera un

electroencefalógrafo en función de sus características, para así poder contrastar esta velocidad de transmisión con el uso de ancho de banda del sistema, y conocer la eficiencia del mismo.

Considere un electroencefalógrafo con las siguientes características:

1. Número de canales: n [canal]
2. Frecuencia de muestreo: f [Hz]
3. Resolución (bits por canal): r [bits/muestra]

Luego la tasa de datos mínima del electroencefalógrafo descrito corresponde a:

$$bw = n \times f \times r \text{ [bits/segundo]}$$

Cabe destacar que esta tasa de datos corresponde básicamente a un óptimo inalcanzable ya que en toda transmisión debe existir información adicional a los datos que se desean transmitir.

Capítulo 4: Diseño de la solución

En este capítulo se plantean dos alternativas de solución al problema identificado y se analizan y comparan en base a los requisitos mencionados en el Capítulo 3:. Luego, se detalla la arquitectura general de la alternativa seleccionada para posteriormente identificar el diseño interno de cada módulo principal del sistema.

4.1 Alternativas de solución

Se consideró las siguientes dos alternativas de solución:

A. Sistema de conexión directa sensor-visualizador:

Esta alternativa considera solo una conexión del tipo cliente-servidor en donde el computador que recibe directamente la información desde el sensor EEG (computador capturador) actúa como servidor, mientras que el computador que permite observar los datos recibidos (computador visualizador) actúa como cliente. En la Figura 4-1 se observa la arquitectura de esta alternativa.

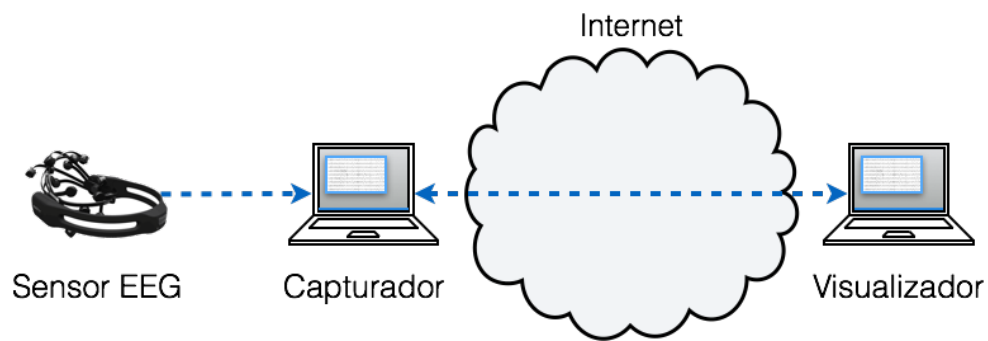


Figura 4-1: Arquitectura solución Sistema de conexión directa sensor-visualizador.

B. Sistema con servidor de almacenamiento y re-envío en Internet:

Esta alternativa contempla la incorporación de un servidor dedicado alojado en Internet por lo que el sistema estaría compuesto por dos conexiones: la primera es entre el computador capturador y el servidor dedicado y la segunda es entre el servidor dedicado y el computador visualizador. En la Figura 4-2 se observa la arquitectura de esta segunda alternativa.

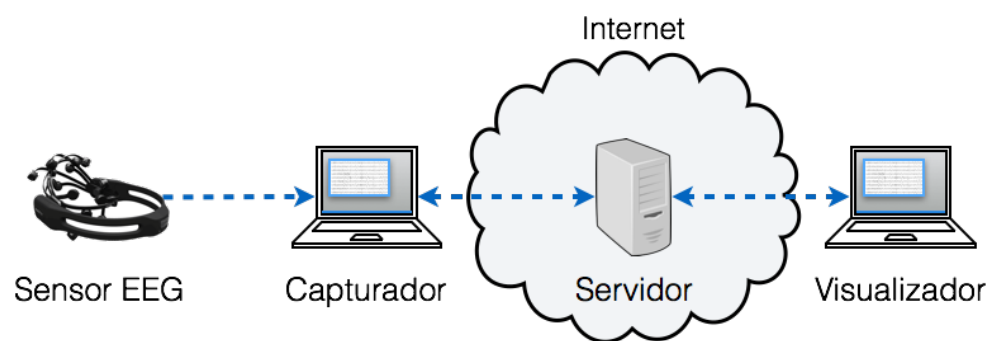


Figura 4-2: Arquitectura solución Sistema con servidor de almacenamiento y re-envío en Internet.

A continuación se analizan las ventajas y desventajas de ambas opciones:

Tabla 4-1: Comparación de alternativas de arquitectura.

	Solución A	Solución B
Ventajas	<ul style="list-style-type: none"> • El desarrollo es más simple pues las aplicaciones se orientan a una única conexión. • Es más robusto a fallas, dado que si existe un error en la transmisión de un paciente, el resto de transmisiones de otros pacientes no se ven afectadas. 	<ul style="list-style-type: none"> • Permite retener información en la “nube” para así poder obtenerlos cuándo y dónde se desee. • Centraliza las transmisiones en un punto en común. • Permite a un visualizador conocer todas las transmisiones a las cuales se puede conectar. • Opera incluso con clientes ubicados en redes con IPs privadas.
Desventajas	<ul style="list-style-type: none"> • El médico especialista debería conocer información específica del computador de su paciente (dirección IP, información que por lo general es dinámica) para poder conectarse a él. • Si el paciente no está conectado no se pueden visualizar datos pasados. 	<ul style="list-style-type: none"> • Tiene un costo mayor al requerir un servidor dedicado. • Mayor uso de ancho de banda del sistema completo. • Tiene un punto de falla crítico: si el servidor no funciona el sistema completo no funciona.

Realizando un análisis en donde se determina nivel de cumplimiento que posee cada alternativa de los requerimientos establecidos en el Capítulo 3:, se concluye lo siguiente:

- La alternativa A no cumple en ningún grado el requerimiento que describe la característica que permite a un médico observar información registrada en el pasado de un paciente (una vez que el paciente ya no se encuentra disponible en el sistema).
- Por otro lado la alternativa B presenta un uso mayor del ancho de banda del sistema completo, pero dado que el punto crítico del sistema en donde se necesita minimizar el uso de este recurso corresponde al computador captador, se considera que esta alternativa cumple de todas formas el requerimiento de eficiencia.
- Ambas alternativas satisfacen completamente el resto de requerimientos.

Finalmente en base a los resultados presentados se elige como alternativa de solución la alternativa B, la cual será referida como EEGRemote (Electroencefalograma remoto).

4.2 Diseño de la arquitectura del sistema EEGRemote

En esta sección se detalla el diseño de la arquitectura del sistema completo, así como el diseño de cada módulo principal del sistema.

Se comienza identificando cada módulo de la arquitectura ilustrada en la Figura 4-1 asignándole un nombre identificador. En la Figura 4-3 se pueden observar dichos módulos junto a sus nombres. Cabe destacar que en la ilustración se menciona el Sensor EEG por motivos de claridad, pero éste no forma parte del sistema en sí.

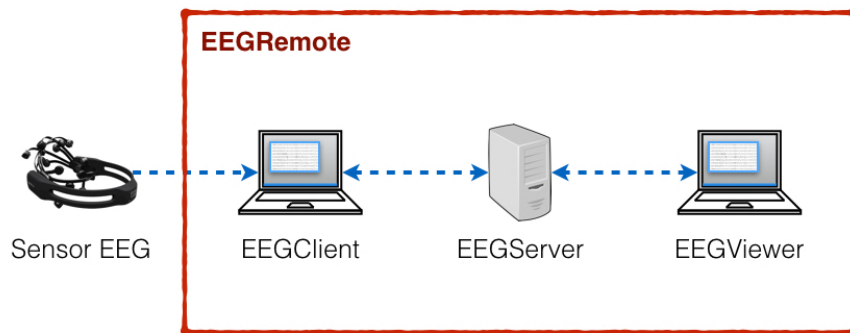


Figura 4-3: Módulos del sistema.

4.2.1 Diseño de la aplicación EEGClient

La aplicación EEGClient corresponde a la aplicación que se ejecuta en el computador capturador, es decir, el computador que recibe los datos provenientes del electroencefalógrafo y luego los transmite hacia el servidor del sistema. Dado que en el uso de esta aplicación se interactúa directamente con el usuario (que en este caso corresponde al paciente al cual se le aplica el examen), esta aplicación debe poseer una interfaz gráfica de fácil uso.

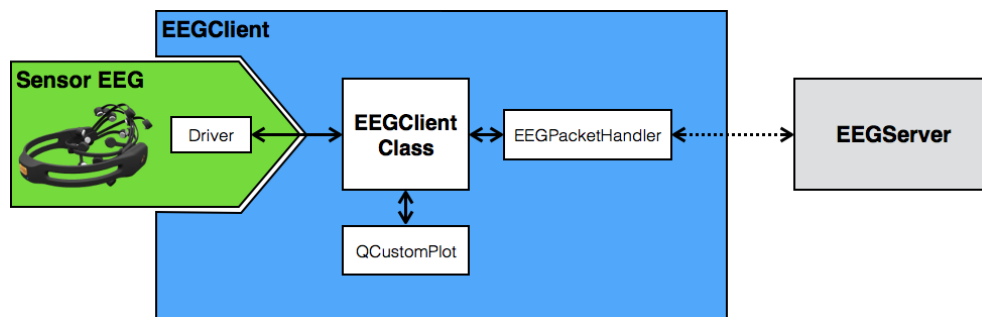


Figura 4-4: Arquitectura del módulo EEGClient.

La arquitectura de esta aplicación y el flujo de datos se ilustran en la Figura 4-4. En dicha figura se añaden el sensor EEG y el módulo EEGServer para facilitar la comprensión del flujo de datos.

Este módulo está compuesto por tres clases importantes:

- *EEGClient Class*: Clase principal de la aplicación que se encarga de la interacción con el usuario y haciendo uso de la biblioteca QCustomPlot ^[5] (mencionada en el Capítulo 5:) muestra gráficamente los datos obtenidos desde el sensor EEG.
- *Driver*: Clase que adecua los datos del sensor EEG a un estándar común para lograr la independencia del resto del sistema respecto del electroencefalógrafo. En particular, para el caso del electroencefalógrafo Emotiv EPOC ^[6], esta clase hace uso de la biblioteca *open source* Emokit ^[7].
- *EEGPacketHandler*: Clase compartida por los tres módulos principales del sistema que se encarga de la transmisión y recepción de paquetes pertenecientes al protocolo de comunicación del sistema detallado en la sección 4.3.

Adicionalmente a las clases mencionadas, este módulo cuenta con una cuarta clase denominada EEGDevEmulator encargada de generar datos aleatorios para así emular el funcionamiento de un sensor EEG. El propósito principal de esta clase es facilitar y agilizar el proceso de desarrollo del sistema.

4.2.2 Diseño de la aplicación EEGServer

La aplicación EEGServer corresponde a la aplicación que debe ser ejecutada en el servidor dedicado. Ésta no posee interfaz gráfica dado que es ejecutada en trasfondo del servidor por tratarse de un proceso “demonio”.

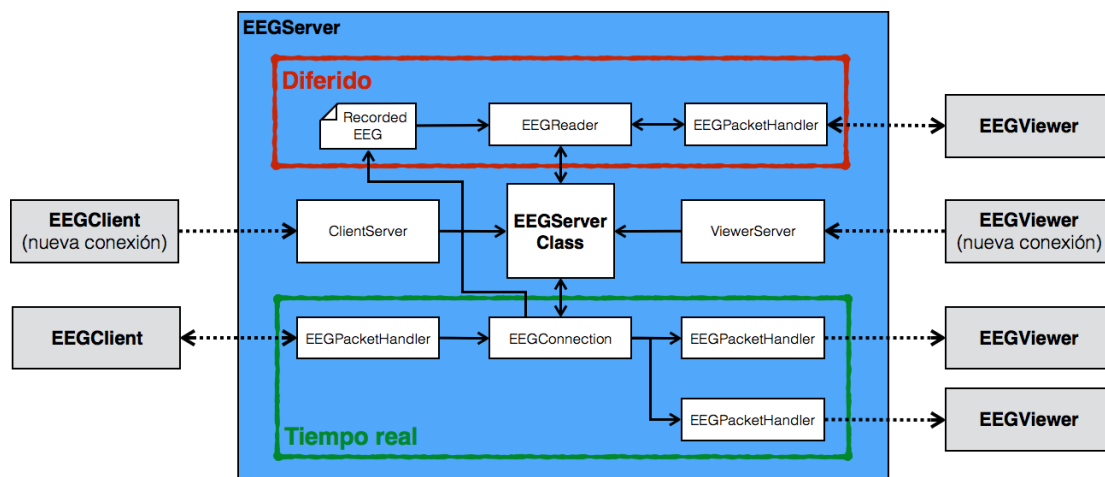


Figura 4-5: Arquitectura del módulo EEGServer.

En la Figura 4-5 se observa la arquitectura del módulo EEGServer. Se ha encerrado en un cuadro rojo los componentes que permiten la transmisión diferida de los datos, mientras que los componentes que permiten la transmisión en tiempo real se han encerrado en un cuadro verde.

Para el caso de la transmisión en tiempo real los datos son recibidos desde la aplicación EEGClient por el módulo EEGPacketHandler, luego el módulo EEGConnection guarda los datos entrantes en el archivo almacenado (Recorded EEG), a continuación otro módulo EEGPacketHandler se encarga de retransmitir los datos vía Internet para finalmente ser recibidos por la aplicación EEGViewer. En el caso de la transmisión diferida los datos son leídos desde un archivo almacenado en el servidor (Recorded EEG) por el módulo EEGReader, luego un módulo EEGPacketHandler recibe estos datos y lo retransmite vía Internet para que finalmente la aplicación EEGViewer los reciba.

Este módulo contiene los siguientes submódulos:

- *EEGServer Class*: Clase principal que se encarga de la interacción con las conexiones entrantes y de comprobar el conjunto los datos guardados en el servidor (transmisiones pasadas).

- *ClientServer* y *ViewerServer*: Corresponden a dos manejadores de conexión, uno para la conexión con los módulos EEGClient y el otro para la conexión con los módulos EEGViewer respectivamente. Estos manejadores de conexión se encargan de recibir las conexiones entrantes de los otros módulos para luego otorgar a la clase EEGServer Class el manejo de cada conexión.
- *EEGConnection*: Clase que se encarga de la retransmisión de los datos provenientes de una aplicación EEGClient conectada. Le permite a la clase EEGServer Class anclar múltiples visores para la retransmisión múltiple (duplicación de datos).
- *EEGReader*: Clase encargada de la lectura y transmisión diferida de los datos correspondientes a transmisiones pasadas. Ofrece la posibilidad de controlar la lectura de los datos gracias a los comandos recibidos desde la aplicación de visualización (utilizada por el médico).
- *EEGPacketHandler*: Clase ya mencionada en la subsección 4.2.1. Se encarga de la transmisión y recepción de paquetes pertenecientes al protocolo de comunicación del sistema detallado en la sección 4.3.

4.2.3 Diseño de la aplicación EEGViewer

La aplicación EEGViewer corresponde a la aplicación que se ejecuta en el computador visualizador, es decir, el computador que recibe los datos desde el servidor del sistema y los grafica adecuadamente para permitir el diagnóstico médico. Dado que en el uso de esta aplicación, al igual que el módulo EEGClient, se interactúa directamente con un usuario (que en este caso corresponde al médico especialista que

estudia los datos transmitidos por el sistema), esta aplicación debe poseer una interfaz gráfica de fácil uso.

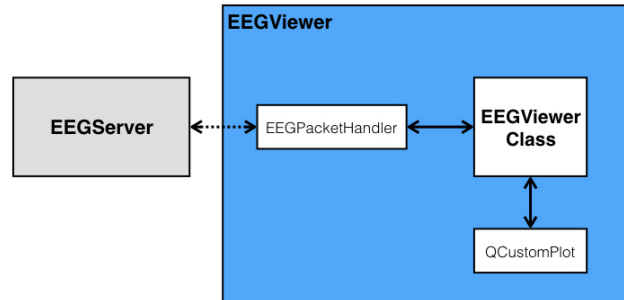


Figura 4-6: Arquitectura del módulo EEGViewer.

La arquitectura de esta aplicación se ilustra en la Figura 4-7. En dicha figura se añaden el módulo EEGServer para facilitar la comprensión del flujo de datos.

Éste módulo está compuesto por dos clases importantes:

- *EEGViewer Class*: Clase principal de la aplicación que se encarga de la interacción con el usuario y haciendo uso de la biblioteca QCustomPlot (mencionada en el Capítulo 5:) muestra gráficamente los datos de la transmisión provenientes de la aplicación EEGServer.
- *EEGPacketHandler*: Clase ya mencionada en la subsección 4.2.1. Se encarga de la transmisión y recepción de paquetes pertenecientes al protocolo de comunicación del sistema detallado en la sección 4.3.

4.3 Protocolo de comunicación

El diseño del protocolo de comunicación contempla parte importante del sistema, por ello, a la hora de definirlo se deben tomar en cuenta todas las consideraciones necesarias para poder cumplir adecuadamente cada uno de los requisitos que fueron definidos para el diseño del protocolo y que corresponden a los siguientes:

- El protocolo debe ser independiente de un electroencefalógrafo particular:
Dado que la cantidad de información a transmitir es dependiente de los parámetros del electroencefalógrafo (número de canales y resolución) se debe diseñar un protocolo que permita la transmisión de paquetes de tamaño variable y así permitir enviar todos los canales capturados en un solo paquete.
- El protocolo debe contemplar la transmisión de comandos y eventos:
El protocolo debe ser diseñado tomando en consideración la necesidad de transmitir identificadores de eventos o comandos que ayuden a la sincronización entre las aplicaciones y el servidor.
- El protocolo debe permitir la transmisión de una lista de identificadores variable:
Considerando que entre las características esperables del sistema es que el médico pueda obtener información de alguno de los pacientes conectados, o de una transmisión diferida, se hace necesario contar con un protocolo que permita la transmisión de una lista de identificadores que le notifique al médico todas las fuentes de información disponibles.

A continuación se presenta el protocolo que cumple con todos los requisitos descritos anteriormente. En la Figura 4-7 se observa la estructura general de los paquetes diseñados. Se considera un *Byte* que identifica el tipo de paquete, y a continuación está el *payload* que es de tamaño variable dependiendo del tipo de paquete.

General structure of packets	
Packet Type	Payload
1 Byte	Variable

Figura 4-7: Estructura general de paquetes.

Los distintos tipos de paquetes se presentan a continuación:

a) Paquete identificador de canales:

Este tipo de paquete permite transmitir la información correspondiente al número de canales y al nombre de cada canal registrado por el electroencefalógrafo. En la Figura 4-8 se observa la estructura de este tipo de paquete. El largo total de este paquete corresponde a $L = 3 + 3 \times n$ [Bytes], donde n es el número de canales del electroencefalógrafo utilizado. El paquete incluye también *DEV_ID* que corresponde a un identificador del modelo del electroencefalógrafo utilizado, mientras que *CH_1* ... *CH_n* corresponden a los nombres de cada canal (de tres Bytes cada uno debido a que no contienen más de 3 caracteres).

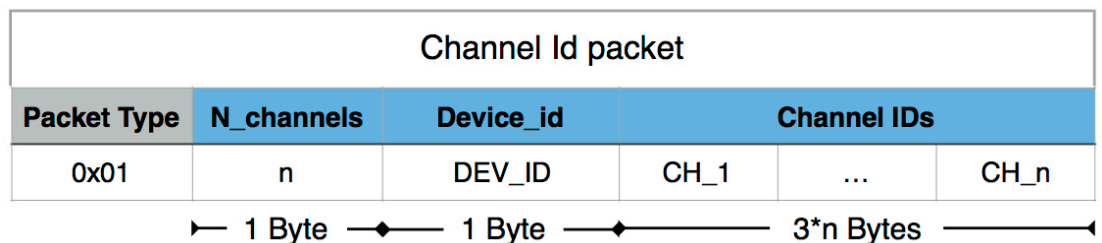


Figura 4-8: Estructura del paquete identificador de canales.

b) Paquete de muestras capturadas:

Este tipo de paquete permite transmitir la información correspondiente a las muestras capturadas por el electroencefalógrafo. En una transmisión

correcta este paquete no debiese ser recibido antes de recibir el paquete identificador de canales. En la Figura 4-9 se observa la estructura de este tipo de paquete. El largo total de este paquete corresponde a $L = 5 + 2 \times n$ [Bytes], donde n es el número de canales del electroencefalógrafo utilizado y se extrae a partir de la información entregada por el canal identificador de canales. En el paquete también se transmite t que representa el tiempo de medición de las muestras, cabe mencionar que este tiempo es calculado en la aplicación EEGClient. Por otro lado, $CH_1 \dots CH_n$ corresponden a las muestras tomadas por el sensor EEG y cada una se entrega en dos Bytes. Es importante aclarar que este valor de dos Bytes debe ser fácilmente modificable en la implementación del protocolo dado que puede ser necesario incrementar el tamaño del paquete en función de la resolución de nuevos electroencefalógrafos.

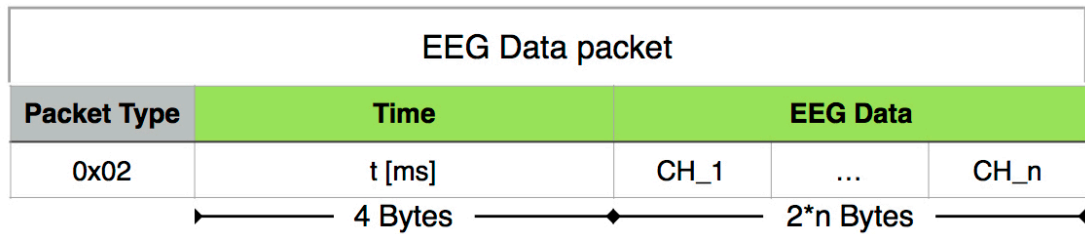


Figura 4-9: Estructura del paquete de muestras capturadas.

c) Paquete de eventos (y requerimientos):

Este tipo de paquete permite notificar la ocurrencia de un evento o la solicitud de un requerimiento dentro del listado de la Tabla 4-2. Facilita la sincronización de las aplicaciones y permite transmitir comandos. En la Figura 4-10 se observa la estructura de este tipo de paquete. El largo total del paquete es fijo y corresponde a $L = 3$ [Bytes]. En este paquete sólo se transmite el valor de e que es un identificador del evento o requerimiento según los valores presentes en la Tabla 4-2.

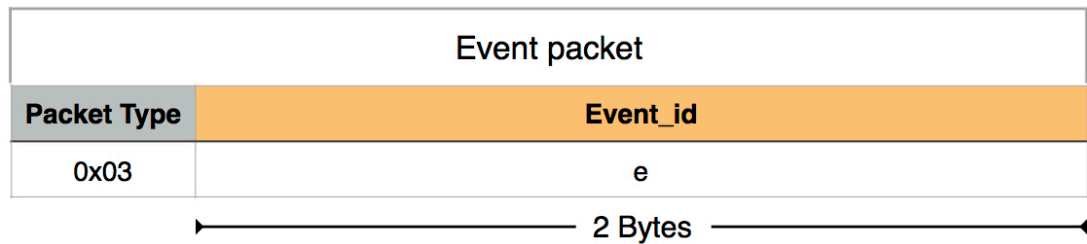


Figura 4-10: Estructura del paquete de eventos.

Tabla 4-2: Lista de posibles eventos o requerimientos.

ID	Significado	Breve explicación
0x00	Conexión aceptada	Evento enviado por el servidor confirmado una conexión entrante como aceptada.
0x01	Conexión rechazada	Evento enviado por el servidor rechazando una conexión entrante.
0x02	Ping (solicitud)	Es utilizado para comprobar que se está conectado a cualquiera de las aplicaciones del sistema.
0x03	Ping (confirmación)	Es utilizado para confirmar una solicitud de Ping.
0x04	Iniciar captura	Permite a la aplicación EEGViewer iniciar/reanudar una transmisión diferida.
0x05	Detener captura	Permite a la aplicación EEGViewer detener/pausar una transmisión diferida.
0x06	Paquete identificador de canales recibido	Confirma la recepción de un paquete identificador de canales. Se utiliza para la sincronización.
0x07	Sesión finalizada	Notifica a la aplicación EEGViewer que se ha alcanzado el final de una transmisión diferida.

0x08	Cliente desconectado	Notifica a la aplicación EEGViewer que el cliente que transmitía en tiempo real se ha desconectado del sistema.
0x09	Error de archivo	Notifica a la aplicación EEGViewer que la transmisión diferida solicitada no se pudo hallar.

d) Paquete de solicitud de nueva conexión:

Este tipo de paquete le permite a la aplicación EEGViewer notificar al servidor que se desea inicial una nueva transmisión y cuál de las fuentes de información disponibles se desea visualizar. En la Figura 4-11 se observa la estructura de este tipo de paquete. El largo total de este paquete es fijo y corresponde a $L = 3$ [Bytes]. En este paquete sólo se transmite *SRC_ID* que corresponde al identificador de la fuente de la cual se desea visualizar su información.

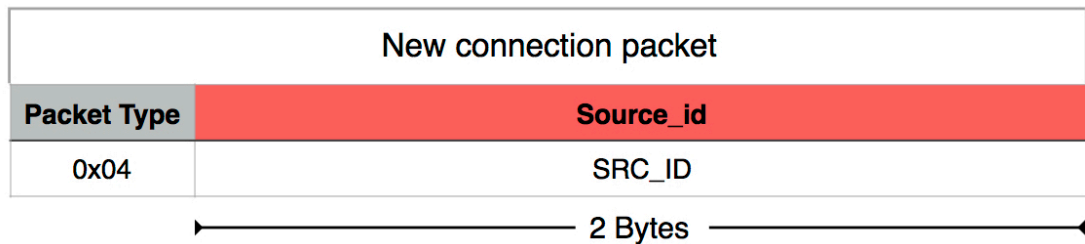


Figura 4-11: Estructura del paquete de solicitud de nueva conexión.

e) Paquete de notificación de nueva fuente:

Este tipo de paquete le permite a la aplicación EEGClient notificar al servidor que se desea conectar y registrar como una nueva fuente para la transmisión en tiempo real. En el paquete se adjunta un nombre (codificado en ASCII) ingresado por el paciente para que el médico pueda identificarlo fácilmente en la lista de fuentes disponibles. En la Figura 4-12 se observa

la estructura de este tipo de paquete. El largo total de este paquete es fijo y corresponde a $L = 31$ [Bytes]. El único elemento que contiene este paquete es *NAME* y corresponde al nombre ingresado por el paciente.

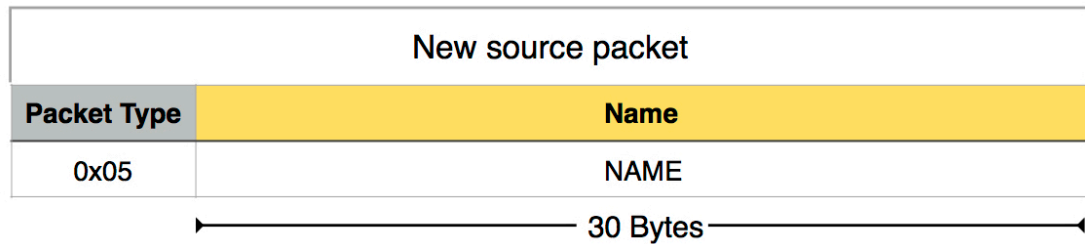


Figura 4-12: Estructura del paquete de notificación de nueva fuente.

f) Paquete de fuentes disponibles:

Este tipo de paquete es utilizado por el servidor y le permite enviar a la aplicación EEGViewer la lista de fuentes de transmisión disponibles. En la Figura 4-13 se observa la estructura de este tipo de paquete. El largo total de este paquete corresponde a $L = 2 + 33 \times m$ [Bytes], donde m es el número de fuentes incluidas en el paquete. Este paquete incluye m fuentes, donde cada una se compone de tres elementos: *SRC_ID*, que corresponde al identificador único de la fuente; *TYPE*, que indica el tipo de fuente (si corresponde a una transmisión de tiempo real o a una transmisión diferida) y *NAME*, que contiene el nombre de la fuente.

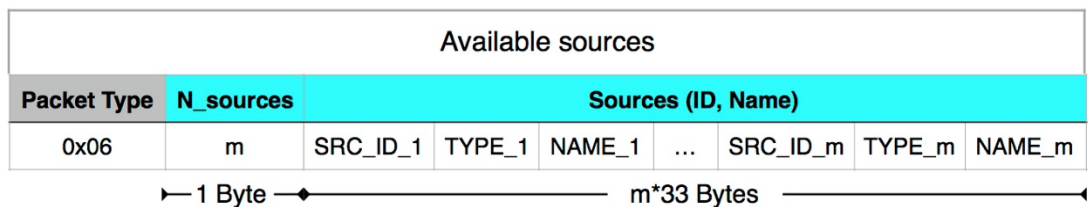


Figura 4-13: Estructura del paquete de fuentes disponibles.

A continuación se detalla la secuencia de transmisiones realizada por el sistema EEGRemote ante dos escenarios: cuando se realiza una transmisión de tiempo real, y cuando se realiza una transmisión diferida.

En la Figura 4-14 se observan las transmisiones realizadas cuando se presenta un escenario donde primero se conecta un paciente mediante la aplicación EEGClient, se presenta con un identificador (nombre) y luego transmite la información referente a la identificación de cada canal capturado por el sensor EEG. Luego, se conecta un médico a través de la aplicación EEGViewer, especificando la fuente a la cual se desea conectar, después el servidor EEGServer le transmite a la aplicación EEGViewer la información de los canales capturados en la fuente, para luego comenzar a transmitir la información correspondiente al EEG. A continuación el paciente finaliza la transmisión y se desconecta del sistema; el servidor le notifica este evento a la aplicación EEGViewer para que finalmente el médico se desconecte del servidor.

listadas (para este escenario una fuente correspondiente a una transmisión diferida) y comunica la fuente seleccionada al servidor. Luego el servidor transmite la información de identificación de los canales capturados en la transmisión seleccionada y espera los comandos del cliente. Para que el cliente comience a recibir la transmisión diferida debe enviar los comandos al servidor (iniciar y detener) y así tener un control sobre la transmisión. Finalmente, cuando el servidor alcanza el final de la transmisión guardada lo comunica al cliente y termina la transmisión.

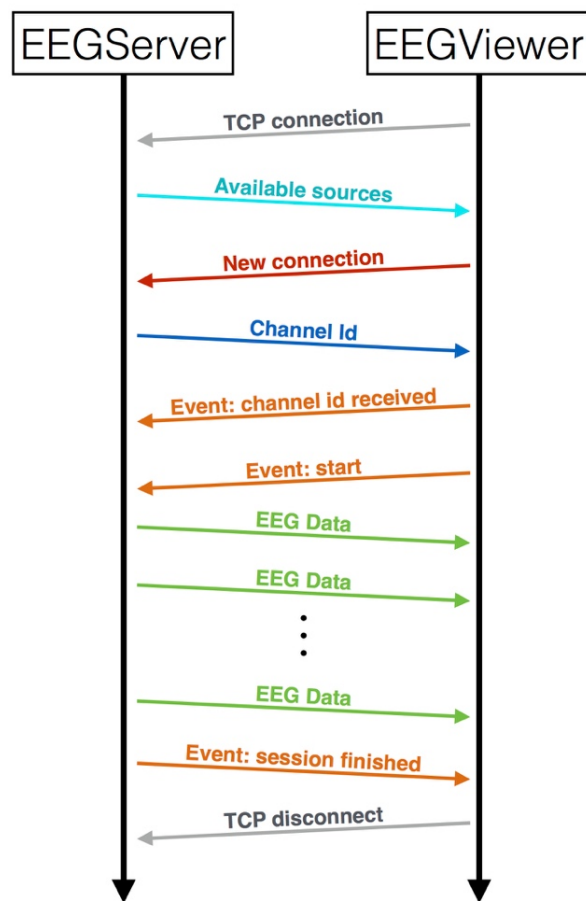


Figura 4-15: Secuencia seguida en la transmisión diferida.

Capítulo 5: Implementación del sistema EEGRemote

En este capítulo se mencionan todos los aspectos técnicos referentes a la implementación realizada del sistema EEGRemote diseñado.

El sistema desarrollado se implementa mediante una metodología de desarrollo iterativa e incremental haciendo uso del IDE Qt Creator 4.0.3 para macOS, como el lenguaje de programación orientado a objetos se utiliza C++. Las bibliotecas y dependencias utilizadas corresponden a las siguientes:

- Qt 5.7 ^[8].

La interfaz gráfica de las aplicaciones EEGClient y EEGViewer se desarrollan utilizando las herramientas ofrecidas por este *framework* multiplataforma gratuito. En ambas aplicaciones mencionadas este *framework* es altamente aprovechado mediante la herencia de la clase QMainWindow que provee manejo de eventos y *widgets* simples para la interacción con el usuario. Además en las tres aplicaciones del sistema se hace uso de las clases que ofrecen abstracción para la comunicación TCP como lo son las clases QTCPSocket y QTCPServer.

- QCustomPlot ^[5].

Corresponde a una biblioteca *open source* desarrollada por Emanuel Eichhammer que ofrece una abstracción sencilla de utilizar basada en OpenGL. En el desarrollo del sistema esta biblioteca es utilizada en las tareas de visualización de los datos.

- Emokit ^[7]:

Es una biblioteca *open source* disponible en GitHub que permite capturar los datos sin procesar registrados por el electroencefalógrafo Emotiv EPOC.

La implementación de un proyecto de *software* generalmente requiere que en cada iteración del desarrollo se realicen pruebas sencillas para ir comprobando el funcionamiento a medida que se desarrolla, y dado que el sistema correspondiente al presente proyecto requiere transmisión de datos, comprobar su funcionamiento conlleva la necesidad de generar datos para luego ser transmitidos. Por esta razón se decide añadir un módulo adicional: un emulador que funcione como generador de señales neuronales aleatorias emulando así el flujo de datos producido por un electroencefalógrafo real. La principal conveniencia de desarrollar este módulo adicional es facilitar y agilizar el proceso iterativo al hacer más rápida la etapa de pruebas.

El emulador desarrollado se denomina EEGDevEmulator y se incorpora en la aplicación EEGClient. Este módulo se implementa de tal forma que solo se puede utilizar definiendo variables en la etapa de compilación, previniendo así los errores indeseados que conllevarían la posibilidad de que un usuario active el emulador.

En cuanto al *hardware* utilizado en el primer prototipo, se utiliza el electroencefalógrafo Emotiv EPOC. Este instrumento es capaz de registrar información de la actividad neuronal proveniente de 14 electrodos ubicados en la cabeza, con una resolución de 14 *bits* por muestra. Es importante mencionar que el uso de este instrumento se debe primordialmente a la disponibilidad que se tiene dada la adquisición por parte del Departamento de Electrónica de la Universidad Técnica Federico Santa María. Sin embargo, si se desea utilizar como electroencefalógrafo en el sistema desarrollado, es necesario validar la utilidad en el diagnóstico de los trastornos neurológicos mencionados anteriormente en la introducción.

En primer lugar se analiza la calidad de la información registrada por el instrumento utilizado. Un estudio realizado por Duvinage *et al.*^[9] concluye que el Emotiv EPOC no posee un óptimo desempeño en comparación a los electroencefalógrafos utilizados actualmente en la medicina, por lo que es necesario hallar una alternativa que permita registrar las señales de manera confiable. Advanced Brain Monitoring ofrece una alternativa orientada a la medicina: el electroencefalógrafo inalámbrico Stat X24 que posee 20 canales con una resolución de 16 [bits].

Por otro lado, se contempla la distribución de los electrodos del instrumento sobre la cabeza que define el origen de los datos registrados por el electroencefalógrafo. En la Figura 5-1 se puede observar la distribución de los electrodos para ambas alternativas.

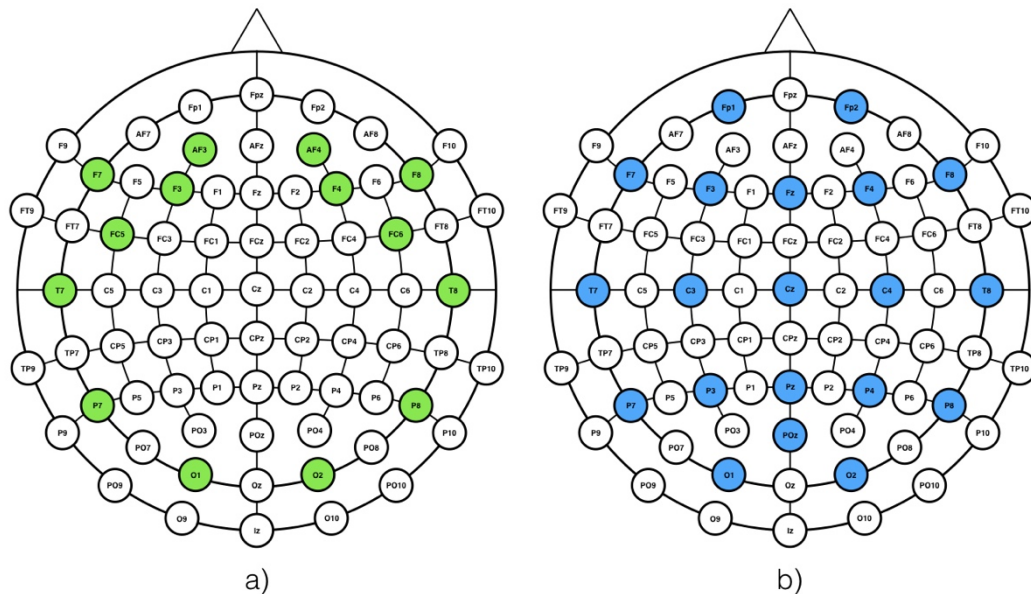


Figura 5-1: Distribución de electrodos en a) Emotiv EPOC y b) Stat X24.

En el caso del sensor Emotiv EPOC se presenta una distribución no estandar que carece de electrodos centrales, lo que limita el estudio neurológico que se puede realizar a los datos. Por otra parte, el Stat X24 presenta una distribución similar al estandar 10-20 que incorpora electrodos centrales.

Capítulo 6: Evaluación del sistema EEGRemote

En este capítulo se presentan y detallan varios escenarios de prueba de las distintas características y requisitos del sistema EEGRemote, comprobando así el correcto funcionamiento del sistema desarrollado.

6.1 Despliegue local con y sin emulador

Como prueba inicial se comprueba el funcionamiento del sistema considerando solo una conexión establecida entre EEGClient y EEGServer, es decir, sin un visualizador EEGViewer.

La secuencia de pasos a seguir en estas pruebas corresponden a los mencionados en la Tabla 6-1. La que también menciona las respuestas esperadas que debiese tener el sistema para cada una de las acciones del usuario. El funcionamiento correcto se verificó observando y comparando las señales transmitidas y visualizadas, así como haciendo uso de las distintas herramientas del sistema.

Tabla 6-1: Secuencia de pasos para las pruebas de despliegue local .

Número	Acción realizada por el usuario	Reacción del sistema esperada
1	El usuario inicia la aplicación EEGClient.	La aplicación debe desplegar la pantalla principal.
2	El usuario selecciona la opción de “Conectar” al servidor.	La aplicación le debe solicitar al usuario que ingrese un nombre identificador.
3	El usuario ingresa un nombre identificador y presiona el botón “Conectar”.	La aplicación debe intentar conectarse y notificar el resultado al usuario.
4	El usuario selecciona la opción de “Iniciar” la captura de datos.	La aplicación debe iniciar la captura y desplegar los datos recibidos en la interfaz gráfica.

5	El usuario finaliza la captura seleccionando la opción de “Desconectar”	La aplicación debe finalizar la captura de datos.
---	---	---

Se definen los siguientes dos escenarios:

- **Primer escenario – con electroencefalógrafo emulado:**

Los supuestos de este escenario son:

- ✓ La aplicación EEGServer ya está iniciada en un computador disponible en Internet y posee conectividad con otro computador en donde se ejecuta la aplicación EEGClient.
- ✓ La aplicación EEGClient está configurada para trabajar con el emulador incorporado.

A continuación, en las Figuras Figura 6-1, Figura 6-2, Figura 6-3, Figura 6-4 y Figura 6-5 se muestran los resultados entregados por el sistema ante cada una de las acciones descritas en la Tabla 6-1.

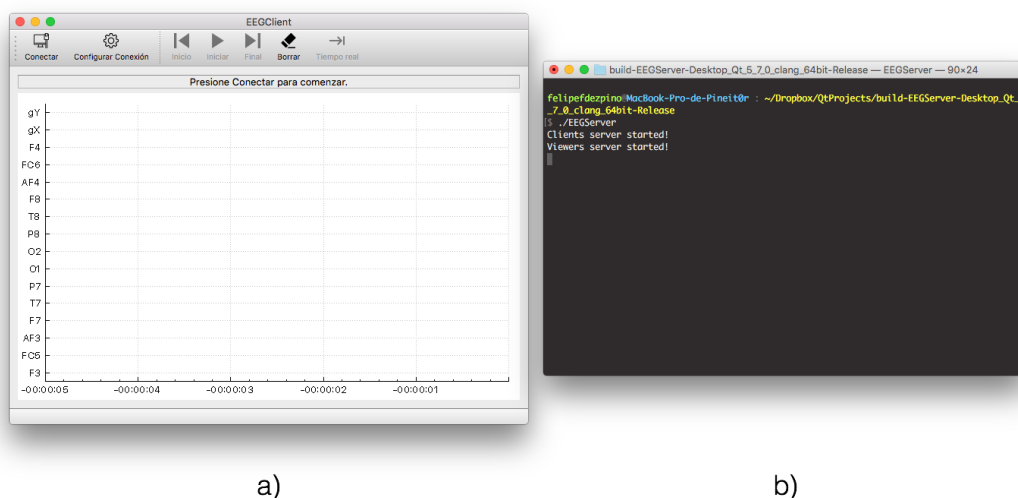


Figura 6-1: Despliegue local con electroencefalógrafo emulado, paso 1:
a) Aplicación EEGClient; b) Aplicación EEGServer.

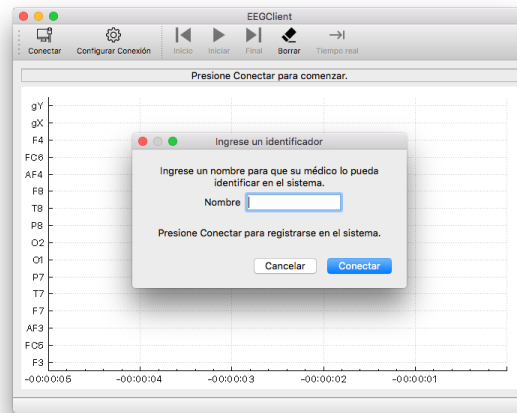


Figura 6-2: Despliegue local con electroencefalógrafo emulado, paso 2: Aplicación EEGClient.

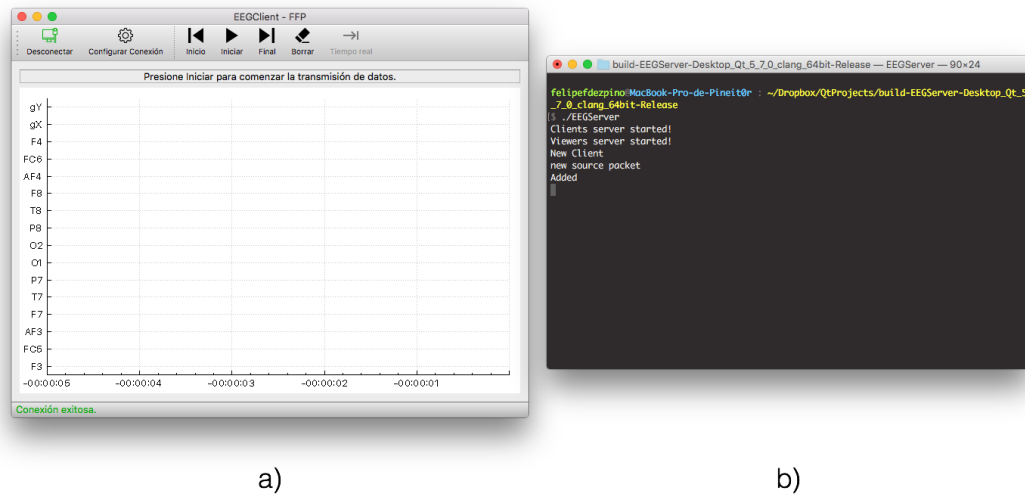


Figura 6-3: Despliegue local con electroencefalógrafo emulado, paso 3:
a) Aplicación EEGClient; b) Aplicación EEGServer.

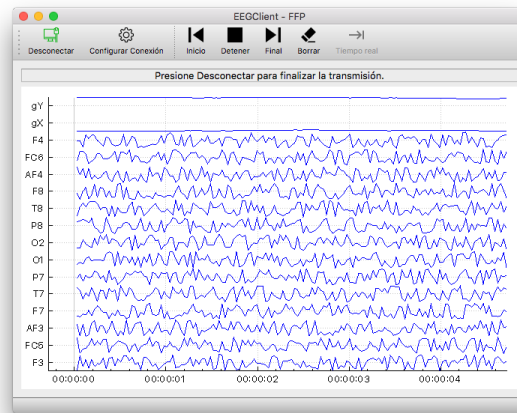


Figura 6-4: Despliegue local con electroencefalógrafo emulado, paso 4: Aplicación EEGClient.

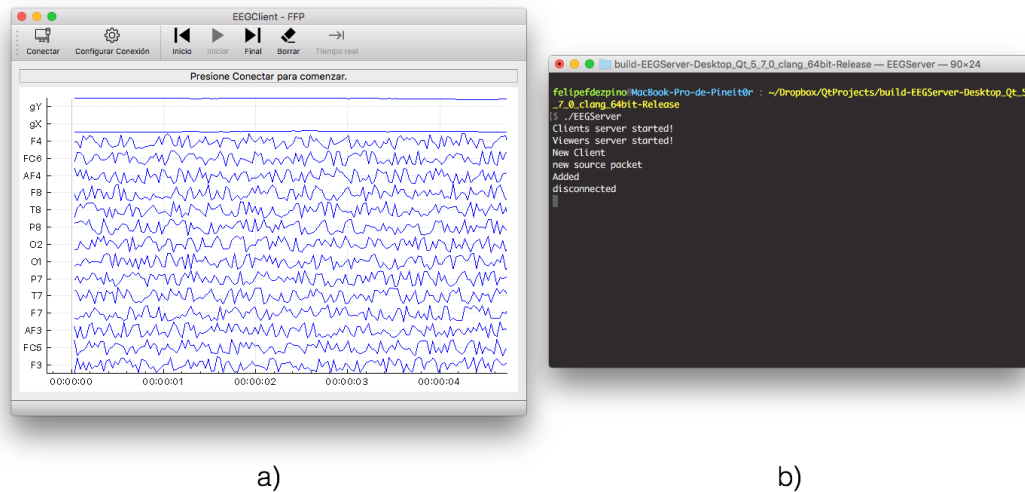


Figura 6-5: Despliegue local con electroencefalógrafo emulado, paso 5:
a) Aplicación EEGClient; b) Aplicación EEGServer.

A partir de los resultados obtenidos se puede concluir que el sistema desarrollado se comporta como se espera de manera local y con el emulador, por lo que se valida la prueba como aprobada.

- **Segundo escenario – usando electroencefalógrafo EPOC:**

Los supuestos de este escenario son:

- ✓ La aplicación EEGServer ya está iniciada en un computador disponible en Internet y posee conectividad con otro computador en donde se ejecuta la aplicación EEGClient.
- ✓ La aplicación EEGClient esta configurada para trabajar con el electroencefalógrafo Emotiv EPOC.

A continuación, en las Figuras Figura 6-6, Figura 6-7, Figura 6-8, Figura 6-9 y Figura 6-10 se muestran los resultados entregados por el sistema ante cada una de las acciones descritas en la Tabla 6-1.

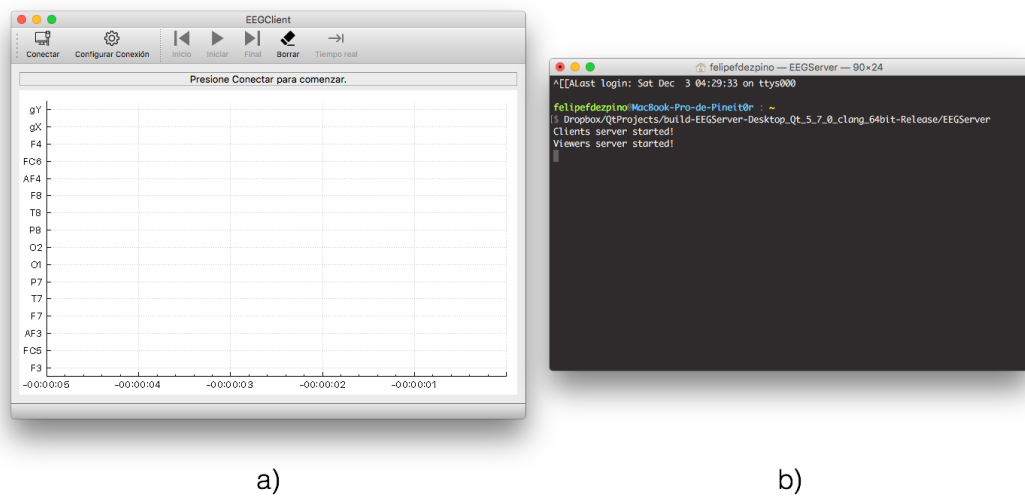


Figura 6-6: Despliegue local usando electroencefalógrafo EPOC, paso 1:
a) Aplicación EEGClient; b) Aplicación EEGServer.

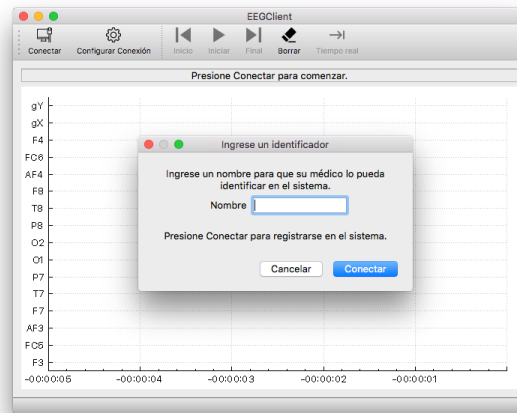


Figura 6-7: Despliegue local usando electroencefalógrafo EPOC, paso 2: Aplicación EEGClient.

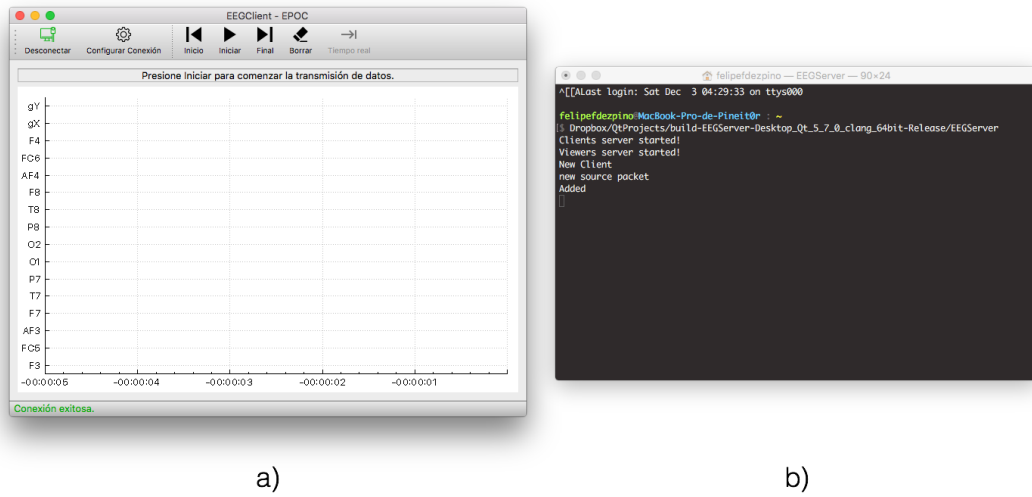


Figura 6-8: Despliegue local usando electroencefalógrafo EPOC, paso 3:
a) Aplicación EEGClient; b) Aplicación EEGServer.

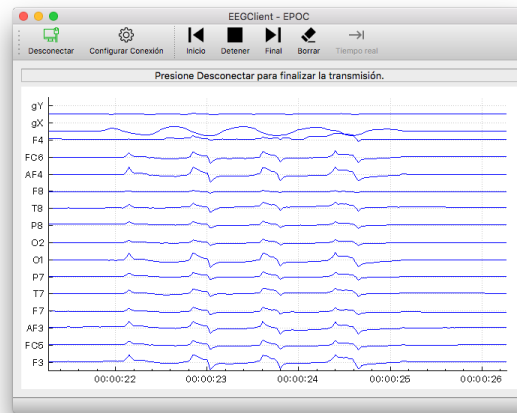


Figura 6-9: Despliegue local usando electroencefalógrafo EPOC, paso 4: Aplicación EEGClient.

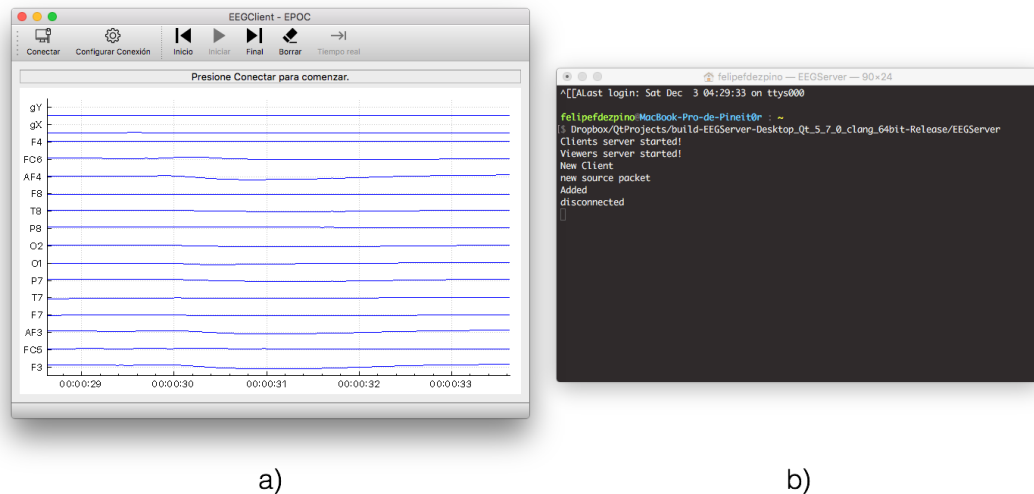


Figura 6-10: Despliegue local usando electroencefalógrafo EPOC, paso 5:
a) Aplicación EEGClient; b) Aplicación EEGServer.

A partir de los resultados obtenidos se puede concluir que el sistema desarrollado se comporta como se espera de manera local y con el uso del instrumento Emotiv EPOC, por lo que se valida la prueba como aprobada.

6.2 Despliegue remoto con y sin emulador

Luego de haber comprobado el funcionamiento del sistema únicamente como despliegue local, se procede a abordar las pruebas que consideran la transmisión remota de los datos en tiempo real. En estas pruebas participan todos los módulos del sistema: EEGClient, EEGServer y EEGViewer.

La secuencia de pasos a seguir en estas pruebas corresponden a los mencionados en la Tabla 6-2. En la cual también se mencionan las respuestas esperadas que debiese tener el sistema para cada una de las acciones del usuario. En esta tabla, se diferencian dos tipos de usuarios: el usuario “paciente” que corresponde a aquel que utiliza la aplicación EEGClient, y el usuario “médico” que es aquel que utiliza la aplicación EEGViewer

Tabla 6-2: Secuencia de pasos para las pruebas de despliegue remoto .

Número	Acción realizada por el usuario	Reacción del sistema esperada
1	El paciente inicia la aplicación EEGClient, se conecta ingresando un nombre identificador e inicia la captura de datos.	La aplicación EEGClient debe desplegar la pantalla principal, realizar la conexión e iniciar la captura.
2	El médico inicia la aplicación EEGViewer	La aplicación EEGViewer debe desplegar la pantalla principal.
3	El médico selecciona la opción “Conectar”.	La aplicación EEGViewer debe intentar conectarse y solicitar al médico que seleccione una de las fuentes de información disponibles.
4	El médico selecciona la fuente correspondiente al paciente y presiona el botón “Conectar”.	La aplicación EEGViewer debe iniciar la visualización de los datos en tiempo real.
5	El paciente finaliza la captura seleccionando la opción de “Desconectar”	La aplicación EEGClient debe finalizar la captura de datos, mientras que la aplicación EEGViewer debe notificar al

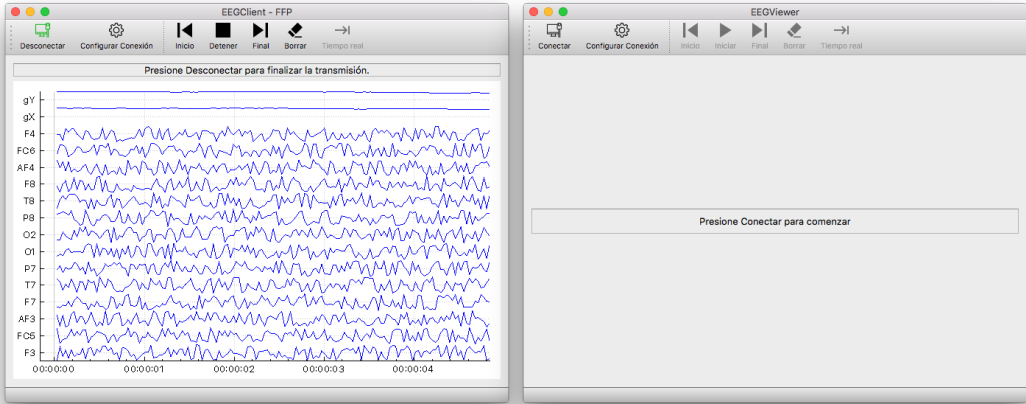
		médico sobre la desconexión del paciente.
--	--	---

- **Primer escenario – con electroencefalógrafo emulado:**

Los supuestos de este escenario son:

- ✓ La aplicación EEGServer ya está iniciada en un computador disponible en Internet y posee conectividad con los computadores en donde se ejecutan las aplicaciones EEGClient y EEGServer.
- ✓ La aplicación EEGClient está configurada para trabajar con el emulador incorporado.

A continuación, en las Figuras Figura 6-11, Figura 6-12, Figura 6-13 y Figura 6-14 se muestran los resultados entregados por el sistema ante cada una de las acciones descritas en la Tabla 6-2.



a)

b)

*Figura 6-11: Despliegue remoto con electroencefalógrafo emulado, paso 2:
a) Aplicación EEGClient; b) Aplicación EEGViewer.*

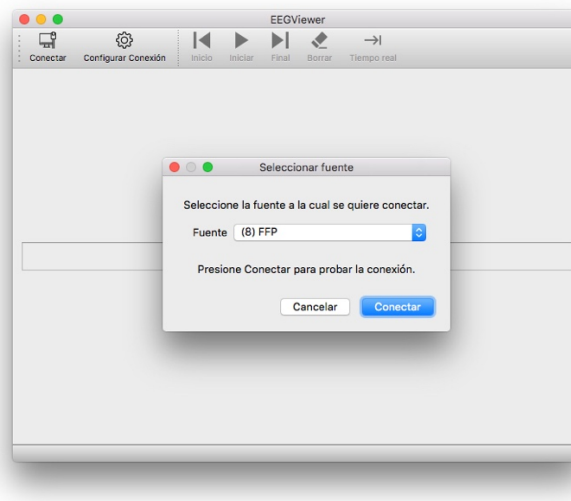


Figura 6-12: Despliegue remoto con electroencefalógrafo emulado, paso 3: Aplicación EEGViewer.

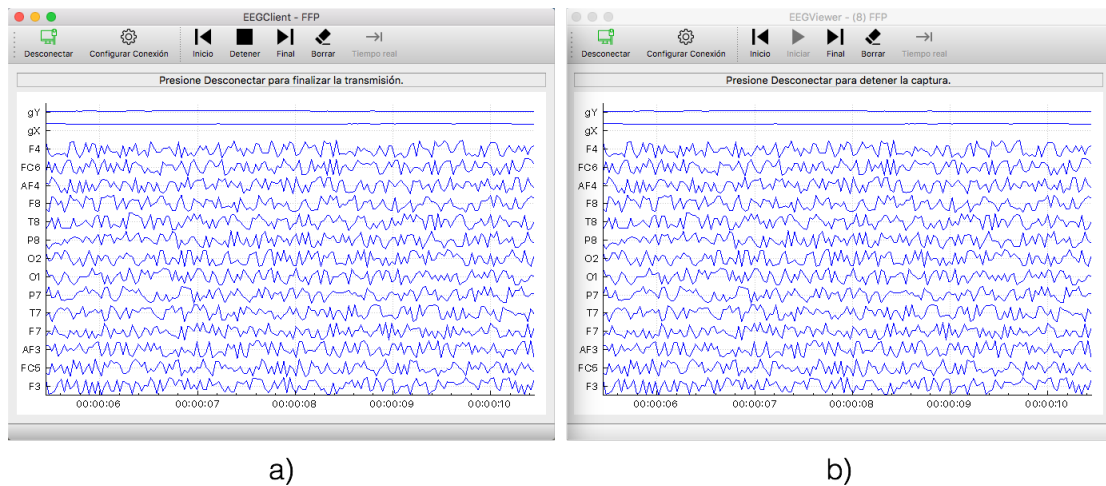


Figura 6-13: Despliegue remoto con electroencefalógrafo emulado, paso 4:
a) Aplicación EEGClient; b) Aplicación EEGViewer.

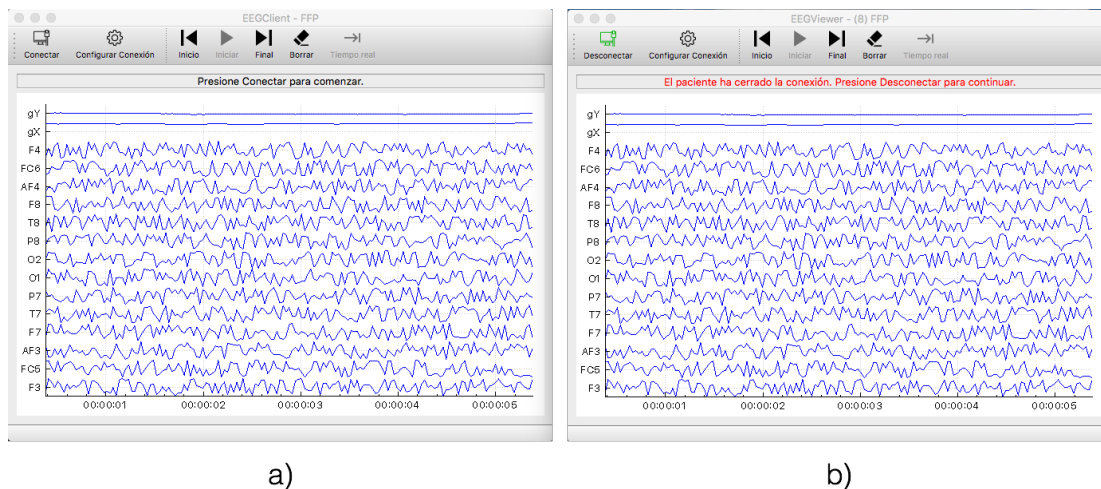


Figura 6-14: Despliegue remoto con electroencefalógrafo emulado, paso 5:
a) Aplicación EEGClient; b) Aplicación EEGViewer.

A partir de los resultados obtenidos se puede concluir que el sistema desarrollado se comporta como se espera de manera remota y con el emulador, por lo que se valida la prueba como aprobada.

- **Segundo escenario – usando electroencefalógrafo EPOC:**

Los supuestos de este escenario son:

- ✓ La aplicación EEGServer ya está iniciada en un computador disponible en Internet y posee conectividad con los computadores en donde se ejecutan las aplicaciones EEGClient y EEGServer.
- ✓ La aplicación EEGClient está configurada para trabajar con el electroencefalógrafo Emotiv EPOC.

A continuación, en las Figuras Figura 6-15, Figura 6-16, Figura 6-17 y

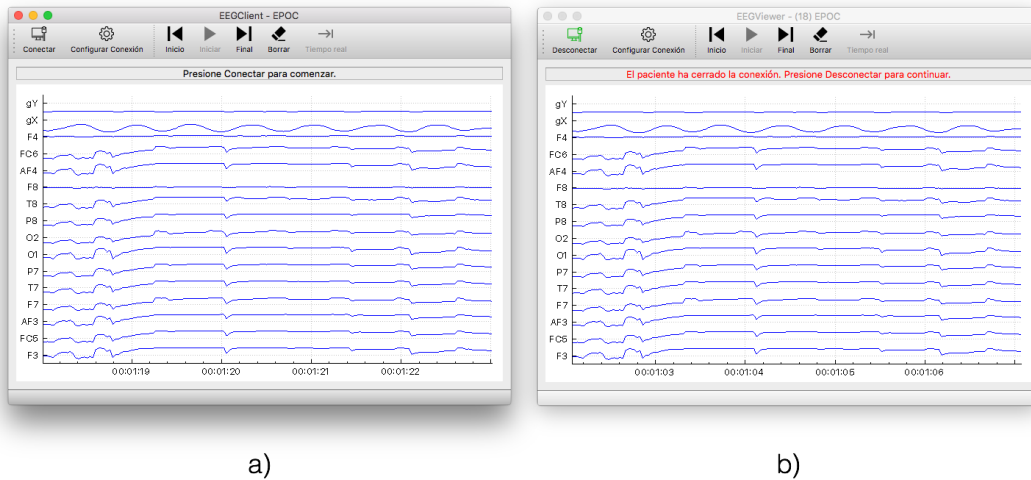


Figura 6-18 se muestran los resultados entregados por el sistema ante cada una de las acciones descritas en la Tabla 6-2.

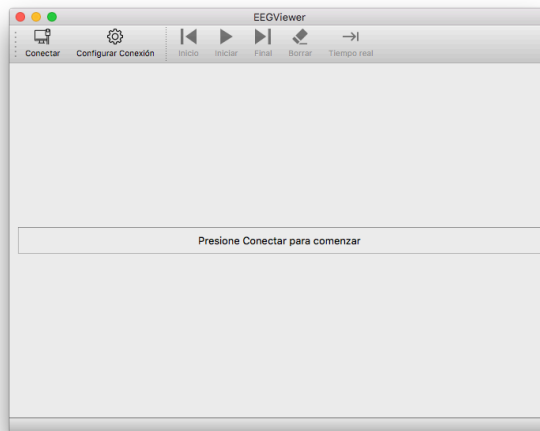


Figura 6-15: Despliegue remoto usando electroencefalógrafo EPOC, paso 2: Aplicación EEGViewer.

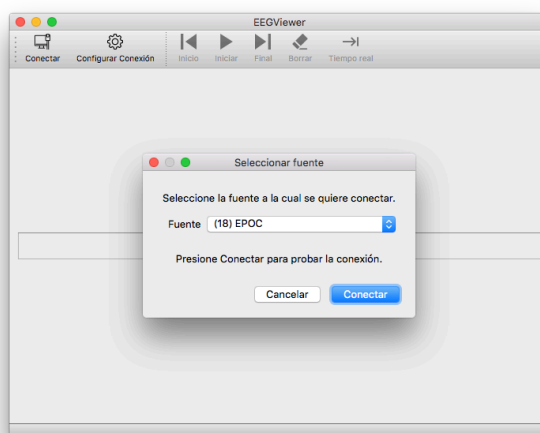
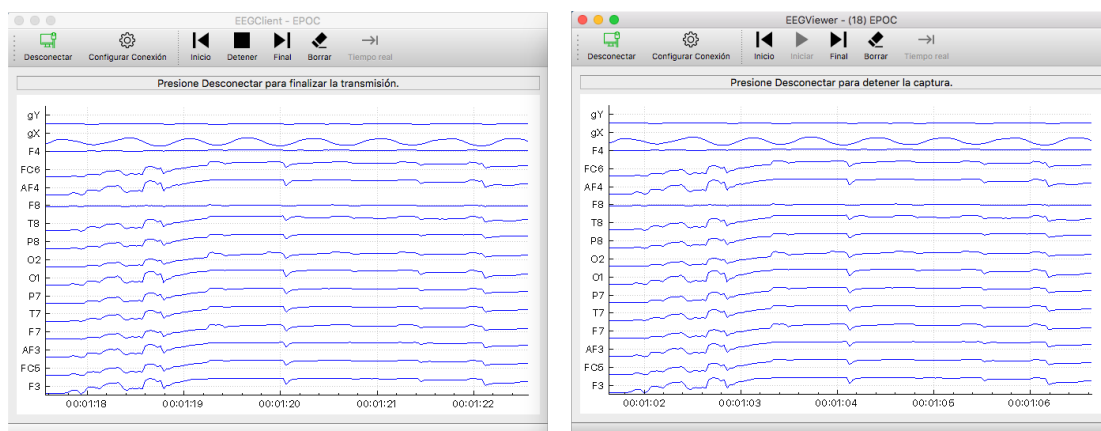


Figura 6-16: Despliegue remoto usando electroencefalógrafo EPOC, paso 3: Aplicación EEGViewer.



a)

b)

Figura 6-17: Despliegue remoto usando electroencefalógrafo EPOC, paso 4:
a) Aplicación EEGClient; b) Aplicación EEGViewer.

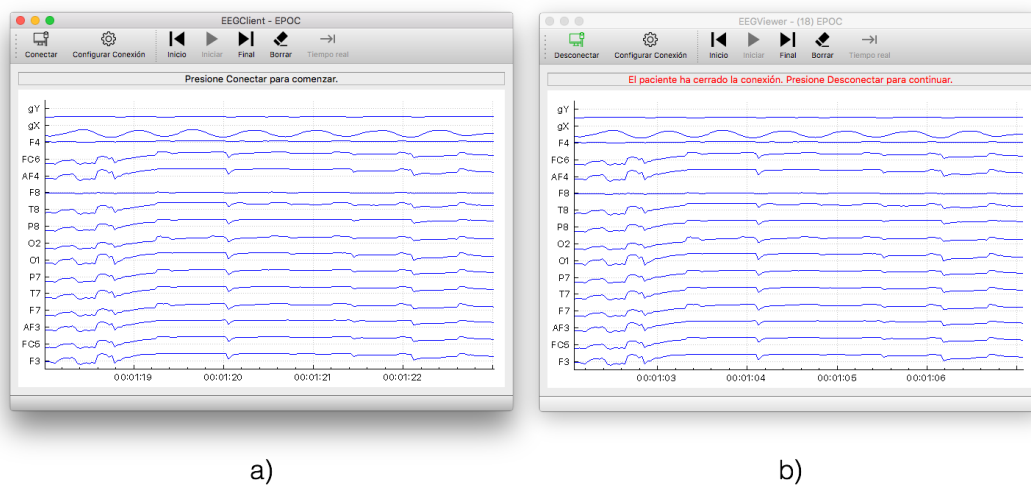


Figura 6-18: Despliegue remoto usando electroencefalógrafo EPOC, paso 5:
a) Aplicación EEGClient; b) Aplicación EEGViewer.

A partir de los resultados obtenidos se puede concluir que el sistema desarrollado se comporta como se espera de manera remota y con el uso de del instrumento Emotiv EPOC, por lo que se valida la prueba como aprobada.

Adicionalmente a las pruebas realizadas en los escenarios correspondientes al despliegue remoto del sistema, se realiza una medición al ancho de banda utilizado por la aplicación en general y el utilizado sólo por la aplicación EEGClient, y luego se contrasta este resultado con el óptimo mencionado en la sección 3.2.

Se utiliza el *software* gratuito Wireshark ^[10] que permite estudiar y analizar a fondo las transmisiones realizadas en un computador, permitiendo aplicar filtros a las distintas interfaces o a los distintos protocolos entre varios.

El escenario medido corresponde al primer escenario de la sección actual (despliegue remoto con emulador) y los pasos realizados corresponden a los mencionados en la Tabla 6-2.

Luego de realizar la captura de los paquetes detectados por la aplicación, se utiliza un filtro con el fin de seleccionar los paquetes pertenecientes a las transmisiones realizadas por las aplicaciones EEGClient (utilizada por el paciente) y EEGViewer

(utilizada por el médico). Luego, haciendo uso del análisis estadístico que proporciona Wireshark (*Statistics – Protocol Hierarchy*) se obtienen los resultados de la .

Tabla 6-3: Velocidades de subida y bajada de cada aplicación.

Aplicación	Capa	Velocidad de subida [kBits/s]	Velocidad de bajada [kBits/s]
EEGClient	Física	23	13
	Aplicación	8,9	0
EEGViewer	Física	13	23
	Aplicación	0	8,9

Ahora que se conoce el ancho de banda de cada aplicación del sistema es posible contrastarlo con el óptimo calculado en la sección 3.2. Teniendo en cuenta los parámetros implementados en el emulador desarrollado (listados en la Tabla 6-4) se procede a calcular el mínimo de ancho de banda necesario para transmitir la información capturada:

Tabla 6-4: Parámetros del emulador desarrollado .

Parámetro	Valor
Número de canales	16 [canales]
Frecuencia de muestreo	30 [Hz]
Resolución	14 [bits/canal]

$$bw = n \times f \times r = 16 \times 30 \times 14 \approx 6,6 \left[\frac{kBits}{segundo} \right]$$

Este ancho de banda, como se menciona anteriormente corresponde a la tasa de datos generada por un electroencefalógrafo con las características anteriores, y por lo tanto corresponde al ancho de banda mínimo que se requiere para transmitir dicha información. Sin embargo, dado el protocolo definido el ancho de banda utilizado por el sistema es mayor dado que no solo se transmite la información capturada por el electroencefalógrafo, sino que también es necesario transmitir más información como por ejemplo el instante de captura de una muestra. Conociendo la tasa de datos generada por el electroencefalógrafo, y el ancho de banda (en la capa de aplicación) utilizado por las aplicaciones del sistema (velocidad de subida en el caso de EEGClient, o velocidad de bajada en el caso de EEGViewer) se puede calcular el *overhead* promedio de los paquetes transmitidos:

$$overhead = \frac{bw_{sensor}}{bw_{sistema}} = \frac{8,9 - 6,6 [kBits/s]}{8,9 [kBits/s]} \approx 25,8 \%$$

El valor de *overhead* promedio refleja que aproximadamente un cuarto de los Bytes transmitidos corresponden a los encabezados. Este alto valor puede ser mejorado si se aplican correcciones al diseño del protocolo como por ejemplo:

1. Transmitir los bits correspondientes a las muestras de manera contigua (sin bits de alineación), Sin embargo esto implica una dependencia de la resolución de muestreo del electroencefalógrafo utilizado.
2. Acumular varias muestras (conjunto de canales muestreados en un instante) en un único paquete de datos y transmitir la frecuencia de muestreo del electroencefalógrafo.

Capítulo 7: Conclusiones y trabajo futuro

En base a los resultados obtenidos en el presente proyecto de memoria, se puede concluir en primer lugar que el desarrollo de cualquier aplicación orientada a la constante mejora debe poseer una etapa de diseño exhaustiva donde se identifiquen tempranamente aquellos aspectos dinámicos a los cuales se debe adaptar la solución buscada, ya que todo cambio de diseño durante la marcha es mucho más costoso en tiempo.

En segundo lugar, se puede concluir que el uso del *framework* Qt se ofrece como principal solución gratuita a la programación multiplataforma de aplicaciones que requieren de una interfaz gráfica con rendimiento nativo.

En tercer lugar, se destaca el buen rendimiento obtenido del prototipo desarrollado, presentando una baja utilización del ancho de banda así como una correcta transmisión y despliegue de los datos, sin presentar deformaciones o pérdidas de información.

Por último, se concluye que el prototipo obtenido en el desarrollo del presente proyecto consiste en una primera solución utilizable al problema identificado, sin embargo, como ya se mencionó en la introducción, la solución ideal al problema contempla varias características que no se abordaron (como la transmisión de los datos desde un dispositivo móvil).

Parte de los problemas presentados en el desarrollo fue la dificultad de hallar una biblioteca que permitiese obtener los datos sin procesar provenientes del electroencefalógrafo Emotiv EPOC; así como la dificultad de graficar correctamente los datos haciendo uso de la biblioteca QCustomPlot.

Como trabajo a futuro se considera importante el desarrollo de una mejora que asegure la confidencialidad de los datos dentro del sistema y que aisle y controle los permisos de acceso de los médicos. Otra mejora de gran importancia es la migración de la aplicación EEGClient a una plataforma móvil capaz de capturar datos de electroencefalógrafos inalámbricos, esto significaría un gran avance en el aspecto

portátil del proyecto, ya que parte de los objetivos del sistema buscado es permitir capturar un EEG en el momento en que se presentan los principales síntomas de las enfermedades neurológicas.

Por otro lado, también se propone como trabajo futuro una mejora en la interfaz gráfica, principalmente en el aspecto de visualización de los datos desplegados, ya que herramientas como una que permita reordenar los canales, o la posibilidad de observar un mapa de actividad cerebral no han sido incorporadas en el primer prototipo.

Finalmente, proponer el trabajo correspondiente al desarrollo de un visualizador equivalente a la aplicación EEGViewer pero implementado en un dispositivo móvil como un celular o una tablet para así otorgar mayores libertades a los médicos y eliminar la necesidad de contar con un computador para hacer uso del sistema.

Referencias

- [1] González De Guevara, Lucía y Guevara Campos, José. *Arch Venez Puer Ped* [en línea] Utilidad de la electroencefalografía en las epilepsias y síndromes epilépticos de la infancia. <http://www.scielo.org.ve/scielo.php?script=sci_arttext&pid=S0004-06492007000200005> [consulta_ diciembre 2016]
- [2] Ricardo, João A. G., França Jr., Marcondes C., Lima, Fabrício O., Yassuda, Clarissa L., & Cendes, Fernando. (2012). The impact of EEG in the diagnosis and management of patients with acute impairment of consciousness. *Arquivos de Neuro-Psiquiatria*, 70(1), 34-39. <<https://dx.doi.org/10.1590/S0004-282X2012000100008>> [consulta: diciembre 2016]
- [3] Talamillo García, Teresa. Manual básico para enfermeros en electroencefalografía. <<http://www.juntadeandalucia.es/servicioandaluzdesalud/huvvsites/default/files/revistas/ED-094-07.pdf>> [consulta: diciembre 2016]
- [4] Listado de electroencefalógrafos en Medical Expo <<http://www.medicalexpo.es/fabricante-medical/electroencefalografo-2716.html>> [consulta: diciembre 2016]
- [5] QCustomPlot, Sitio web del proyecto Open Source QCustomPlot [en línea] <<http://www.qcustomplot.com>> [consulta: diciembre 2016]
- [6] Emotiv, Sitio web de Emotiv [en línea] <<https://www.emotiv.com>> [consulta: diciembre 2016]
- [7] GitHub-Emokit, Sitio web del proyecto Open Source Emokit [en línea] <<https://github.com/openyou/emokit>> [consulta: diciembre 2016]
- [8] Qt, Página web oficial de Qt para desarrolladores [en línea] <<https://www.qt.io/developers/>> [consulta: diciembre 2016]
- [9] Duvinage *et al.*: Performance of the Emotiv Epoc headset for P300-based applications. *BioMedical Engineering OnLine* 2013

<<https://biomedical-engineering-online.biomedcentral.com/articles/10.1186/1475-925X-12-56>> [consulta: diciembre 2016]

[10] Wireshark, Página web oficial de Wireshark [en línea]
<<https://www.wireshark.org>> [consulta: diciembre 2016]