

# Comunicación entre un dispositivo de adquisición de bioseñales cerebrales y un terminal móvil

*Communication between a brain biosignal acquisition device and a mobile terminal*

Marco Antonio Pinto Orellana

Departamento de Ciencias Exactas e Ingeniería, Universidad Católica Boliviana  
Av. General Galindo s/n, Cochabamba, Bolivia

[pinto.marco@live.com](mailto:pinto.marco@live.com)

**Resumen:** La presente investigación contrasta los beneficios del uso de diferentes métodos de codificación dentro un protocolo de comunicación, en términos de su confiabilidad. Para este fin, se diseñó un protocolo en la capa de enlace (BTLCP), con el que se midió el parámetro BER de distintas combinaciones de técnicas de modulación y codificación. La fuente de datos en este sistema fue un conjunto de ondas cerebrales muestreadas provenientes de estudios ajenos, y además, señales electroencefalográficas generadas por un modelo propio. Este conjunto de condiciones fue implementado y simulado dentro una aplicación específica de simulación de eventos discretos y cálculo matricial (OpenTonitruum). Los resultados de las pruebas dieron una alta variabilidad respecto a la codificación y una baja variación en relación a la modulación; demostrando que el efecto de las diferentes técnicas de modulación de fase es indiferente, siendo las técnicas de codificación las verdaderamente influyentes en el desempeño. Además, se encontró que la codificación por turbocódigos posee la tasa de errores más baja que el resto, aumentando la confiabilidad del protocolo.

**Palabras clave:** protocolo, codificación, modulación, bioseñales y turbocódigos.

**Abstract:** The present study contrasts the benefits of using different coding methods within a communication protocol, in terms of its reliability. For this purpose, a protocol was designed in the link layer (BTLCP), with which the parameter measured BER of various combinations of modulation and coding techniques. The source of data in this system was a set of brainwave studies sampled from outside, and also electroencephalographic signals

generated by a proprietary model. This set of conditions was implemented and simulated within a specific application of discrete event simulation and matrix calculus (OpenTonitruum). The test results were highly variable with respect to the coding and low variation in the modulation ratio, demonstrating that the effect of the different phase modulation techniques is indifferent, coding techniques being really influential in the performance. Furthermore, we found that possesses turbocodes coding error rate lower than the rest, increasing the reliability of the protocol.

**Keywords:** protocol, coding, modulation and turbo codes biosignals.

## 1 Introducción

El sistema nervioso es una inmensa red de neuronas y células gliales. La interacción de estas unidades biológicas se logra a través del impulso nervioso, que permite modelar al conjunto de neuronas como generadores bioeléctricos[1]. Existen diversas formas de registrar esta actividad, el método más común y accesible es el electroencefalograma (EEG), que consiste en un registro de la actividad eléctrica en la corteza cerebral, inventado por el psiquiatra alemán Hans Berger en 1924 para permitir visualizar las señales resultantes de la corriente iónica del conjunto de los potenciales de las dendritas[2].

Por simplicidad y accesibilidad, para obtener estas señales se suelen emplear electrodos superficiales o scalp situados directamente sobre el cráneo con algún compuesto coloidal (gel electrolito). Existe una norma internacional para la distribución de estos transductores, conocida como «Sistema 10-20» [3], que asigna una letra y un número único a 19 posiciones, que se encuentran con 20% de separación entre sí respecto a la distancia entre el inión (protuberancia externa del occipital) y nasión (sutura internasal)[2][4].

Debe resaltarse, que este tipo de señales no solamente tienen fines exclusivamente médicos, sino también aplicaciones prácticas en la automatización de tareas mediante algoritmos interpretativos. Los sistemas de procesamiento e interpretación de ondas cerebrales no son totalmente efectivos, debido a diversos factores presentes en cada individuo y en el sistema mismo. Para minimizar el primer tipo de dificultad se aplican algoritmos de aprendizaje y entrenamiento previos[5], buscando asegurar la precisión de los datos. Pero, en el segundo caso, los mecanismos de corrección se limitan a los dispositivos que captan la señal, minimizando el ruido presente en éstos[1], sin analizar el tipo de canal, codificación, e incluso, modulación óptima para enviar la información inherente a las bioseñales. Los errores introducidos sobre la señal original - en particular, el retardo - pueden generar resultados inapropiados una vez que se realice el procesamiento de la información con los

medios matemáticos temporo-espectrales (Wavelet o Fourier)[4]. Debe considerarse además, que cada algoritmo de codificación puede incluir valores particulares o singulares en los que no se garantice la detección o corrección de errores - cálculo de polinomios o vectores síndrome en el caso de códigos lineales o cíclicos -, por esta razón, al definirse un conjunto concreto de señales, o un cúmulo de señales de cierta naturaleza, se consigue analizar la validez del algoritmo en un contexto particular con mayor certidumbre. Este fue el enfoque que se otorgó al análisis de las muestras de bioseñales.

Adicionalmente, estos sistemas tienen la particularidad de emplear mecanismos independientes de control y transmisión de bioseñales. Este efecto genera que diferentes diseños de interfaces cerebro-ordenador [1][6] no puedan usarse en aplicaciones exclusivamente encargadas de las funciones de discriminación, filtrado y procesamiento[2] debido a la incompatibilidad de los sistemas y procesos de comunicación entre el dispositivo captador de señales y el destinatario, comúnmente, ordenador.

## 2 Metodología

La ejecución del estudio se realizó a través de tres actividades primordiales:

1. **Modelamiento de las bioseñales cerebrales** para generar una cantidad, teóricamente, ilimitada de señales electroencefalográficas aleatorias.
2. **Diseño y especificación del protocolo de comunicación** en la capa de enlace, denominado BTLCP, permitiendo las funcionalidades mínimas de control de canal.
3. **Modelamiento de los métodos de codificación, modulación y canal**, dentro un entorno de simulación denominado Open Tonitruum, implementado específicamente para esta investigación.

Estas tres actividades se complementan mutuamente en una estructura tal como se muestra en la Fig. 1. En el sistema simulado, existe un receptor y un transmisor, ambos equipados con un módem (modulador-demodulador) y un códec (codificador-decodificador). La salida del módem se transmite a través de un canal genérico, modelado como un canal de ruido gaussiano blanco aditivo (AWGN). La forma en la que los dos dispositivos se comunican entre sí responde a los procedimientos definidos por el protocolo BTLCP. Finalmente, las señales que ingresan al sistema provienen de dos fuentes diferentes: ondas almacenadas en una base de datos, o señales generadas por el modelo EEG.

Existe una motivación particular para estudiar las ondas EEG y crear un modelo de señales aleatorias de esta naturaleza: mayor fuente de datos. La base de datos contiene un poco menos de 150 ondas diferentes, cantidad insuficiente

para probar adecuadamente el sistema. Sin embargo, el uso del modelo, permitió generar una onda para cada repetición del experimento; aumentando, la variabilidad de los datos.

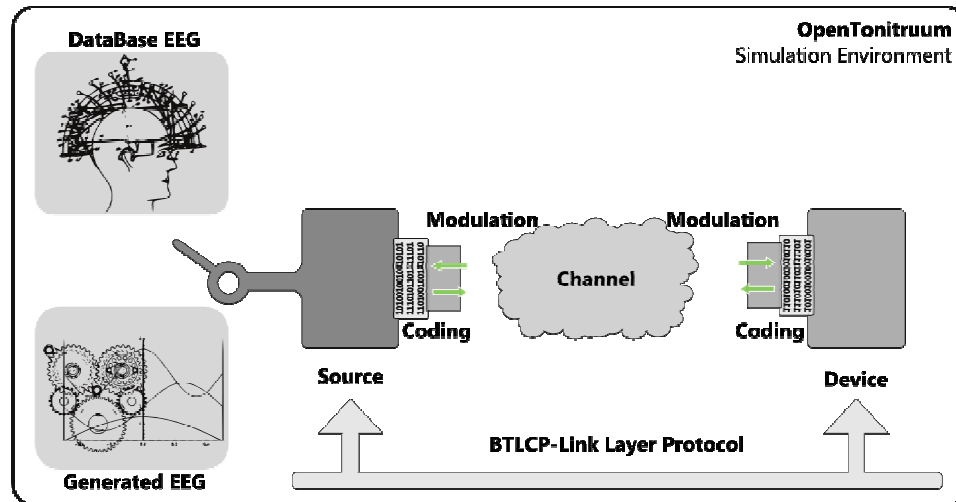


Figura 1: Entorno simulado de la investigación

## 2.1 Modelamiento de bioseñales cerebrales

Las señales electroencefalográficas aparentan un comportamiento aleatorio o errático explicado levemente mediante ritmos o señales base[7]. Para fines de la investigación, se hizo necesario el uso de la mayor variedad posible de estas ondas durante la simulación, para comprobar adecuadamente las técnicas de modulación y codificación. En correspondencia a este propósito se analizó la naturaleza de las ondas EEG en busca de una relación estocástica que permita representarlas.

Se empleó como base de datos de bioseñales, el conjunto de datos (dataset) de la «BCI Competition II» (2003) del instituto «Berlin Brain-Computer Interface» que recogió muestras de un experimento de análisis visual de contenido léxico (P300 spellerparadigm). Este conjunto de datos proveyó 1451519 muestras distribuidos en 147 segmentos de señales.

Después de una inspección y análisis preliminar a las diferentes señales en el dominio del tiempo y frecuencia[7], se observó que en ambas condiciones, las ondas no parecían asemejarse a una combinación de funciones lineales, logarítmicas, exponenciales o sinusoides, sino aparentaban responder a un comportamiento totalmente aleatorio. No obstante, se halló una peculiaridad: el espectro en frecuencia en todas las situaciones demostraba una envolvente aparentemente común.

El espectro de frecuencia de las ondas EEG tiene dos características esenciales:

- La forma de la envolvente sigue un patrón cercano a una función exponencial con un superíndice decreciente.
- Existe un impulso de tamaño variable mayor que la tendencia exponencial situado en una banda de frecuencia menor a 100Hz.

Para encontrar la envolvente del espectro se empleó un filtro pasabajo de 10Hz sobre la señal equivalente a la distribución de frecuencias basándose en el principio del detector de envolvente de AM.

Dado el aspecto incierto de las señales electroencefalográficas, desde un punto de vista determinístico, es difícil hallar una ecuación que establezca la forma de estas señales rigurosamente. Pero, es posible producir un espectro que se asemeje a estas señales en el dominio de las frecuencias. El método para ello consistiría en generar valores aleatorios entre el eje de las abscisas y la envolvente para todas las frecuencias. De esta forma el modelo de las señales EEG se traduciría en encontrar una ecuación paramétrica de la envolvente.

La envolvente, si bien tiene un carácter exponencial, no puede ser adecuadamente modelada por una función exponencial común, por su bajo índice de correlación con las señales originales; y aunque este mismo índice se mejora con el uso de una función gaussiana, no supera el 70%[7]. Después de algunos análisis preliminares se encontró que la envolvente puede ser satisfactoriamente modelada con una combinación de las expresiones anteriores:

$$E(f) = c_1 \cdot e^{\frac{-f-c_2}{2c_3}} + c_4 \cdot e^{\frac{-|f|-c_2}{c_5}} \quad (\text{ec. 1})$$

Debe notarse que los coeficientes  $c_i$  de la ecuación son valores variables.

## 2.2 Protocolo de comunicaciones BTLCF

La transmisión de bioseñales en medios inalámbricos requiere que el protocolo permita tramas de tamaño variable, al mismo tiempo que debe facilitar mecanismos para la corrección de errores en las mismas (FEC, Forward Error Correction), pero permitiendo su retransmisión en caso que las correcciones fuesen insuficientes. En el caso particular de aplicaciones en tiempo real, como la interpretación semi-instantánea de bioseñales, el protocolo debe minimizar el retraso en la recepción de los datos en el destinatario.

La alternativa más común para este tipo de situaciones suele ser el protocolo HDLC (High Data Link Control)[8] o una versión derivada enfocada en alguna condición en particular. Las ventajas de este protocolo involucran: tamaño de trama variable, uso de diferentes modos de conexión, control de flujo y bidireccionalidad. Sin embargo, el principal obstáculo que impide el uso de este protocolo es el deficiente soporte para comunicaciones inalámbricas en aspectos como la seguridad y sincronización.

En cualquier caso, el protocolo debería usar una técnica de control de flujo óptima. Este requisito se basa en la premisa de maximizar el número de tramas de información recibidas en el destinatario en el menor tiempo posible. Las opciones más comunes incluyen: «Stop and Wait ARQ», «Go Back N» o «SelectiveRepeat». Siendo la última alternativa la que más se acerca a la condición deseada.

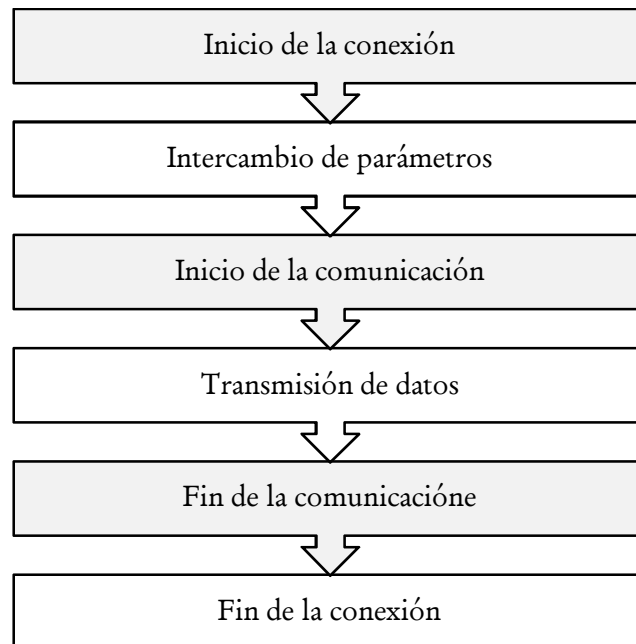
En este contexto, se encontró un protocolo que se acercó bastante a las especificaciones indicadas: AIRMAIL[9]. Éste definía el uso de control y corrección de errores a nivel de bits y tramas, esta última opción sustentada en la extrapolación de la codificación Hamming a nivel de tramas (o paquetes) en lugar de bits. Asimismo AIRMAIL dispone de una técnica diferente de control de flujo, cercana a «Selective Repeat», consistiendo en la adición de un bit especial que señala el fin de un bloque (permitiendo así grupos de trama de tamaño menor al establecido por la ventana de transmisión), con una sola trama de confirmación de recepción, o un grupo de tramas señalando los paquetes a retransmitir, por cada bloque. A pesar de estas características, este protocolo no las especificaba completamente, ni establecía un formato de trama, haciéndolo poco factible de implementarse en la presente investigación.

En el ambiente descrito, se diseñó un protocolo experimental punto-a-punto en la capa de enlace del modelo abierto OSI que pudiera satisfacer las necesidades de las transmisiones inalámbricas de bioseñales: BTLCP (Biosignals Transmission Link Control Procedure). Este protocolo posee las siguientes características:

- Se aprovechan algunas propiedades de AIRMAIL: el uso de un bit especial (bit «e») para señalar el fin de bloque y el envío de una sola trama de confirmación por cada uno de ellos.
- El control de flujo se realiza mediante «SelectiveRepeat», pero usando un formato especial para indicar las tramas a retransmitir, de manera que se envíen el menor número de tramas con el objetivo de informar.
- Se emplean dos tipos de tramas: de información y configuración. El formato de ambas se basó en HDLC descartando los campos innecesarios y añadiendo algunos más.

- Los equipos pueden intercambiar valores de configuración al inicio de la conexión, con el fin de informarse sobre algunos aspectos relacionados con la comunicación. En el caso exclusivo de bioseñales EEG, se podría emplear esta propiedad para indicar el número de canales, el tamaño de cada muestra (que define también el tamaño de la trama), entre otros factores. Este proceso, también es útil para intercambiar claves públicas e implementar técnicas de encriptación. Sin embargo, todas estas posibilidades se encuentran fuera del alcance del presente estudio.

El funcionamiento básico de BTLCP se puede ver en la Fig. 2. Aunque para describir formalmente el protocolo se utiliza una máquina de estados finitos extendidos para comunicaciones (CEFSM, Communicating Extended Finite State Machine)[8]. El diagrama del transmisor se muestra en la Fig. 3 y la notación correspondiente se detalla en la Tabla 1.



**Figura 2:** Etapas en el protocolo BTLCP.

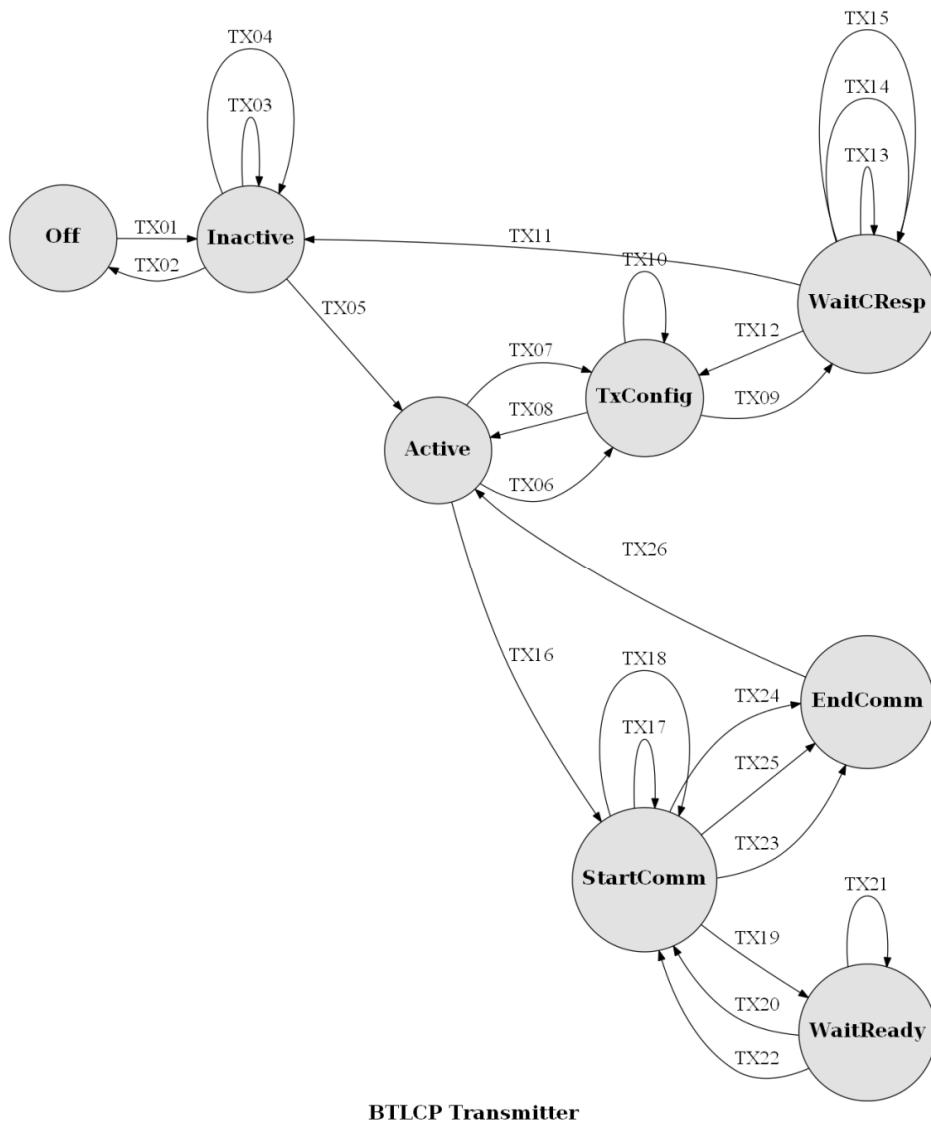
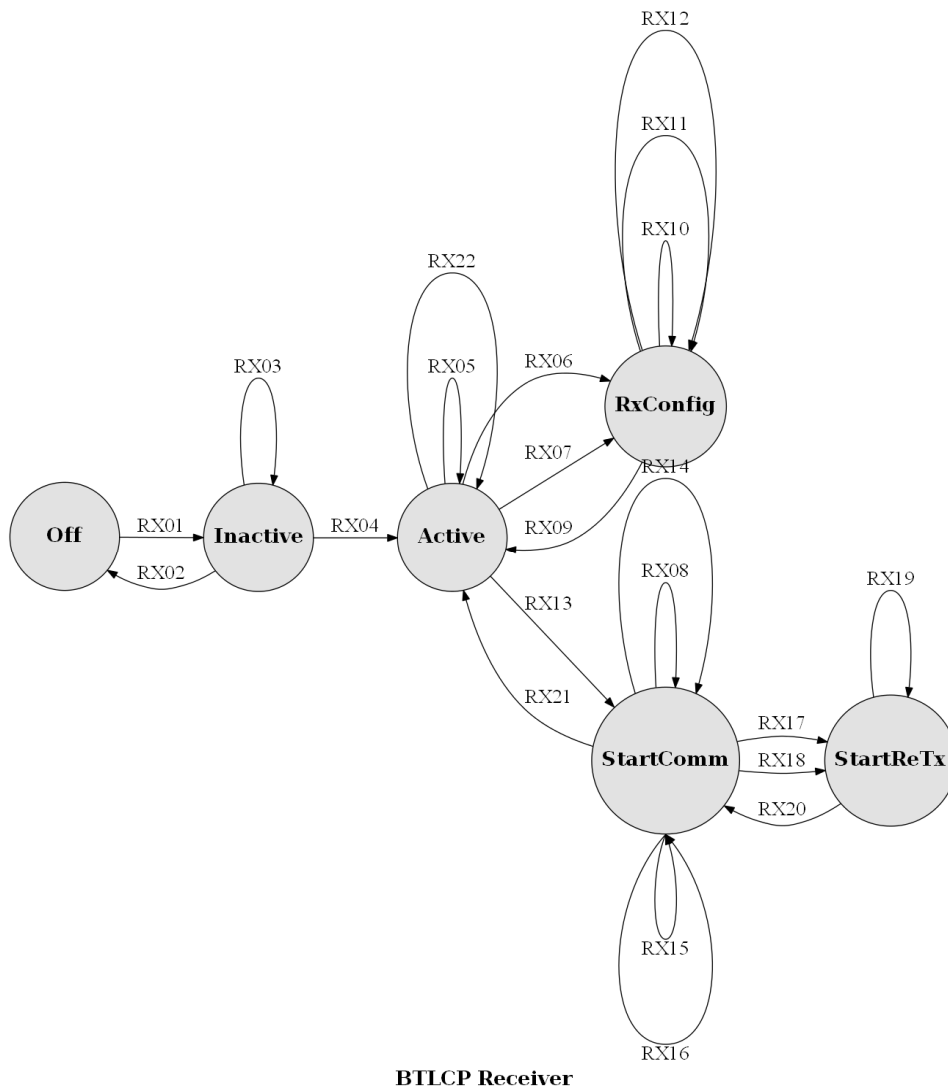


Figura 3: Diagrama CEFSM del transmisor BTLCP.





**Figura 4:** Diagrama CEFSM del receptor BTLCP.

**Tabla 1.** Notación del diagrama CEFSM del transmisor BTLCP

Notación	Descripción
TX01	app! SignalStart [rx? start connection]
TX02	TimeoutInactive [default transition]
TX03	rx! ReceivedBadFrame [rx? start connection]
TX04	rx! ReceivedRepeatFrame [rx? start connection]
TX05	rx! ReceivedStartConnection [rx? start connection; verify existence(); verify configparams(); verify initials()]

---

TX06	SignalReconfigEachTime [verify configparams(); initial configparams()]
TX07	SignalFirstConnection [verify configparams(); initial configparams()]
TX08	SignalEmptyconfigList [default transition]
TX09	SignalNotEmptyconfigList [rx? configframe]
TX10	rx! ReceivedStartConnection [rx? start connection(); verify existence()]
TX11	rx! ReceivedconfigRejected [default transition]
TX12	rx! ReceivedconfigAccepted [verify configparams(); increment config index list()]
TX13	rx! ReceivedBadFrame [rx? config frame]
TX14	rx! ReceivedRepeatFrame [rx? config frame]
TX15	rx! TimeoutWaitCResp [rx? send config frame]
TX16	SignalStartComm [initializetransmission()]
TX17	SignalBufferEmpty [initializetransmission()]
TX18	SignalEndBufferNotReached [rx? current buffer frame]
TX19	SignalEndBufferReached [del buffer indexes()]
TX20	rx! ReceivedReadyBlock [initializetransmission()]
TX21	rx! ReceivedNoFinalRepeatBlock [add indexes()]
TX22	rx! ReceivedFinalRepeatBlock [add indexes()]
TX23	SignalEndComm [rx? end comm frame]
TX24	rx! ReceivedBadFrame [rx? end comm frame]
TX25	rx! TimeoutStartComm [rx? end comm frame]
TX26	rx! ReceivedEndCommFrame [default transition]

---

El receptor, por su parte, se puede representar con el diagrama de la Fig. 3, con la notación descrita en la Tabla 2.

**Tabla 2.** Notación del diagrama CEFSM del receptor BTLCP

Notación	Descripción
RX01	app! SignalStart [tx? start connection]
RX02	TimeoutInactive [default transition]
RX03	rx! ReceivedBadFrame [tx? start connection]
RX04	rx! ReceivedStartConnection [rx? start connection; verify configparams()]
RX05	rx! ReceivedStartConnection [tx? start connection]
RX06	SignalReconfigEachTime [verify configparams(); initial configparams()]
RX07	SignalFirstConnection [verify configparams(); initial configparams()]
RX08	rx! ReceivedconfigFrame [analyseconfig()]
RX09	SignalEmptyconfigList [default transition]

Notación	Descripción
RX10	rx! ReceivedStartConnection [tx? start connection]
RX11	rx! ReceivedBadFrame [tx? repeat frame]
RX12	rx! ReceivedconfigFrame [verify received config(); analyse config()]
RX13	SignalStartComm [initializereception()]
RX14	rx! ReceivedBadFrame [default transition]
RX15	rx! ReceivedNoFinalInfoFrame [analysis info frame()]
RX16	rx! ReceivedFinalInfoFrameAndNoErrors [tx? info ready(); initialize reception()]
RX17	rx! ReceivedFinalInfoFrameAndErrors [prepare retx()]
RX18	TimeoutStartComm [add absent indexes(); self. prepare retx()]
RX19	SignalEndRetxNotReached [tx? retxframe]
RX20	SignalEndRetxReached [tx? retxframe]
RX21	rx! ReceivedEndCommFrame [tx? end comm frame]
RX22	rx! ReceivedEndCommFrame [tx? end comm frame]

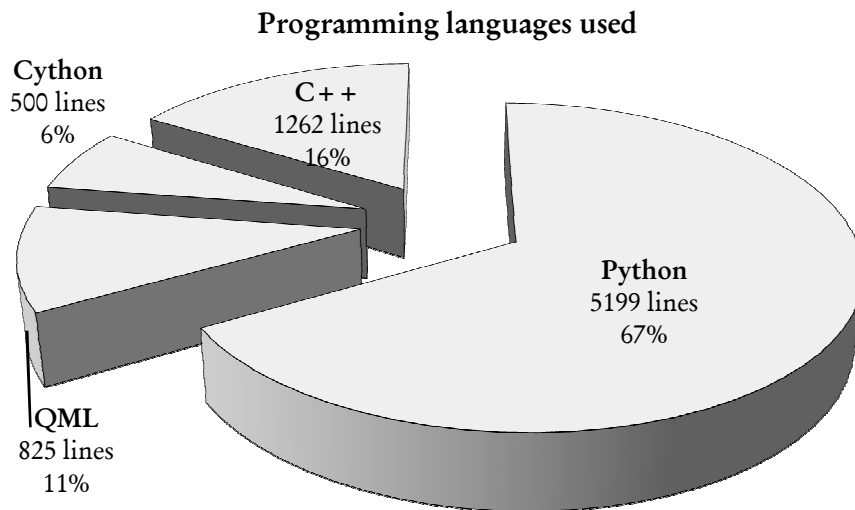
### 2.3 Herramienta de simulación Open Tonitruum

Para obtener los resultados, fue necesario realizar simulaciones sobre un protocolo determinado, empleando diversas técnicas de modulación y codificación. Esto con el objetivo de contrastar el desempeño de los turbocódigos con el algoritmo BJCR. No existe una aplicación que reúna estas características, las más próximas fueron Matlab (u Octave, SciLab como alternativas de código abierto), GNS-3 y OMNet++. Pero, ninguna de ellas permite realizar todas las actividades de forma simultánea.

- Usar diferentes combinaciones de técnicas de modulación y codificación.
- Permitir cambiar la configuración de la simulación en tiempo de ejecución.
- Incluir mecanismos para seleccionar el tipo de fuente de señales EEG a usar (generadas o de la base de datos).
- Mostrar gráficamente los avances de la señal por el sistema de transmisión.

Para fines prácticos, la aplicación se desarrolló bajo el nombre de «Open Tonitruum». Con un código escrito en 4 lenguajes de programación: Python, C++, Cython y QML (Figura 3) basándose en las siguientes librerías:

- **IT++:** funciones matemáticas escritas en C++, se empleó para algunos algoritmos de codificación y modulación.
- **SimPy:** simulación de eventos discretos diseñado en Python. Es el fundamento de la simulación del sistema. En base a esta herramienta se implementó el sistema de comunicación virtual simulado.
- **NumPy/SciPy:** cálculo matricial para Python, añade un conjunto de funciones matemáticas de índole científica; su principal uso fue destinado al análisis estadístico, método de mínimos cuadrados, filtros, entre otros.
- **Matplotlib/GuiQwt:** graficación de funciones. Matplotlib se requirió para las gráficas realizadas durante las pruebas, mientras que guiqwt se usó como base en la aplicación.
- **PyQt:** biblioteca multipropósito para Python portado de Qt (C++), facilitando las herramientas necesarias para diseñar la interfaz gráfica multiplataforma.



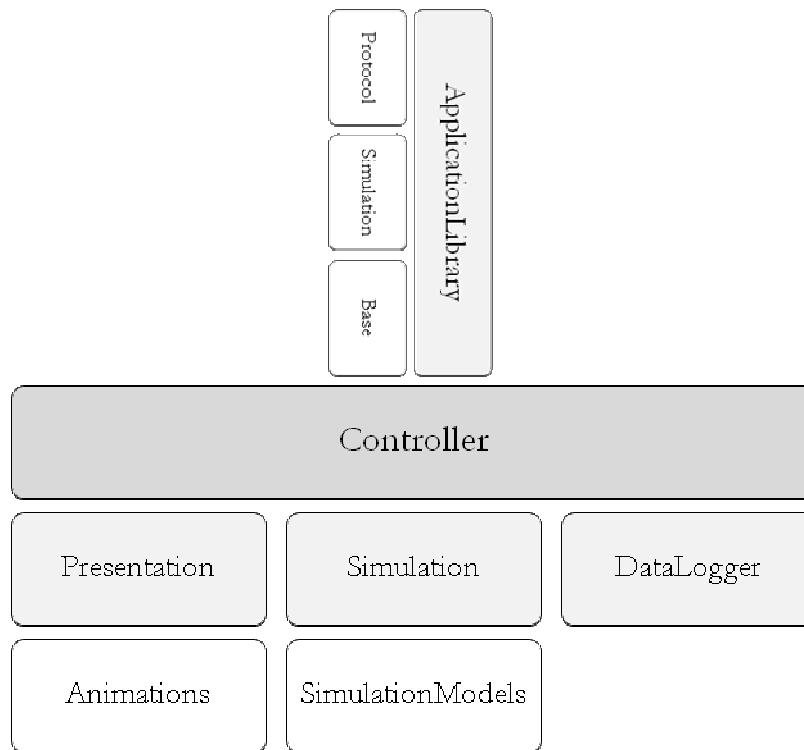
**Figura 5:** Lenguajes de programación empleados en el entorno de simulación

La aplicación está desarrollada en torno al paradigma de la programación orientada a objetos, y funciona en base a un modelo de capas, tal como se muestra en la Fig. 5:

- **Application Library Layer.** Toda la aplicación se sustenta en un conjunto especial de clases agrupadas dentro de la capa «ApplicationLibrary», que equivale al paquete «OpenTonitruum.Common» del código fuente. Este nivel de abstracción ofrece utilidades como enumeraciones y eventos.
- **Simulation Layer.** El conjunto de mecanismos de simulación, como tal, se agrupan en la capa «Simulation» que realiza las funciones de control y mecanismos de funcionamiento en tiempo de ejecución. En base a estos procedimientos, se desarrolla una serie de modelos de simulación que se cargan durante la ejecución del programa en forma externa a la aplicación central (subcapa «Simulation Models»). Estos modelos definen la forma en la que se realiza la simulación, por tanto, se dividen en modelos de canal, modulación, codificación y protocolo. Cada uno de ellos, indica dentro de su contenido, los algoritmos o cálculos necesarios a desarrollar durante la simulación.
- **Data Logger Layer.** La capa de registro de datos «DataLogger» se encarga de almacenar los resultados de la simulación en una base de datos diseñada con este fin.
- **Presentation Layer.** La capa «Presentation» y su subcapa «Animation» tienen el objetivo único de establecer la interfaz de usuario para la manipulación de los detalles de la simulación.
- **Controller Layer.** La capa «Controller» tiene las funciones de permitir la interacción entre la capa de presentación y simulación.

La aplicación se basa en una biblioteca de simulación de eventos discretos (SimPy), mediante la que se realiza la interacción de componentes en una línea del tiempo independiente al tiempo del ordenador. En base a este principio, se creó dos tipos de simulaciones dentro del programa:

- Los **escenarios**, que consisten en simulaciones en las que el tiempo de simulación puede ser calibrado de forma que se ejecute proporcionalmente al tiempo real. Asimismo, en esta categoría, las simulaciones incluyen animaciones que muestran el desarrollo de la simulación desde diversas perspectivas, según los elementos del sistema de transmisión (fuente, codificador, modulador, canal, demodulador y decodificador) y el diagrama de estados del protocolo, entre otros.
- Los **entornos**, son un medio alternativo de simulación carente de animaciones y optimizado en velocidad, con la diferencia que puede simular más de un tipo de configuración en forma consecutiva.



**Figura 6:** Diagrama de capas de OpenTonitruum

En ambos casos, cuando la aplicación se inicia, se cargan los modelos de simulación de un directorio especificado y los parámetros que éstos requieren para funcionar correctamente. De esta forma, al empezar la simulación, se crea en un ambiente virtual con dos «objetos de simulación» que representan los equipos donde cada uno de ellos posee un transceptor - conformado por un componente virtual receptor y transmisor - mediante el cual es capaz de comunicarse a través del canal simulado. El canal, modulador, demodulador, codificador y decodificador se establecen mediante una configuración previamente especificada, usando los modelos existentes para representar el comportamiento de los diversos tipos de componentes. Asimismo, el protocolo de capa de enlace también se carga y configura mediante este método.

#### 2.4 Detalles de la simulación

Empleando el programa descrito anteriormente, se procedió a definir las condiciones de la simulación. Los métodos de modulación seleccionados fueron: 4-PSK, QAM, 16-QPSK y 4-PAM; mientras que las técnicas de codificación elegidas, BCH, Hamming, Reed Salomon y Turbocódigos; en un canal ideal y un canal gaussiano (AWG). Se empleó un elevado número de

repeticiones (200) de cada simulación para conseguir que el conjunto de resultados tenga una distribución normal, según el teorema del límite central, de forma que los análisis estadísticos comunes puedan ser aplicados[10][11].

Con las características establecidas, y considerando que el protocolo de comunicación se sustenta en BTLCP, se preciso varios escenarios con una combinación diferente de canal, modulación, codificación y fuente, asignándose un código identificador a cada uno de ellos. Después de aproximadamente 50 horas de simulación sobre un computador Dell Vostro 1000 (TM) con un procesador AMD Dual-Core Athlon(TM) 64X2 con 1GB de memoria RAM bajo el sistema operativo Windows Seven(TM), se produjeron los resultados de la tasa de error de bits BER (bit error rate) sobre tramas enviadas con el protocolo BTLCP presentados en la Tabla 4. Debe notarse, que el número total de bits varía de acuerdo al efecto que tuvo los errores en el protocolo. Cuando la cantidad de errores fue demasiado alta, los temporizadores dentro de cada equipo se ejecutaron, generando que se ingrese en un estado de error y/o los equipos finalizaran la comunicación.

**Tabla 3.** Especificación de modelos de codificación, modulación, canal y protocolo

Modelo	Parámetro	Valor
BTLCP	Tiempo límite de simulación	100001
	Identificador del transmisor	1
	Identificador del receptor	2
	Relación de la velocidad de simulación respecto al tiempo real	2
	Simulación en tiempo real	Verdadero
	Matriz generadora	$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}$
Turbocódigos	Matriz generadora 2	$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}$
	Algoritmo de decodificación	BCJR
	Factor de escala LOGMAX	0.7
	Longitud de restricción	3
	Número de iteraciones	8
Hamming	Secuencia de entrelazado	Entrelazado WCDMA de 40 bits de longitud
	m	3
BCH	Código sistemático	Falso
	m	4
	k	7
	t	2

Modelo	Parámetro	Valor
Reed Salomon	Código sistemático	Falso
	m	3
	t	1
PAM – PSK – PAM	Símbolos por modulación (m)	4
	Frecuencia de portadora	50KHz
QAM	Símbolos por modulación (m)	4
	Frecuencia de portadora	50KHz
	Número máximo de muestras	10000
Fuente de señal basada en el modelo	Longitud de frecuencias analizadas en el modelo	150
	Número de bits en el formato IEEE-754	16
	Frecuencia de muestreo	250Hz
	Número máximo de muestras	10000
Fuente de señal de la base de datos	Número de bits en el formato IEEE-754	16
	Frecuencia de muestreo	250Hz
	Escala de muestreo	1e-6V

Tabla 4. Resultados de la simulación

Canal	Modul.	Codifi.	Fuente	Bits enviados	BER promedio	Desviación típica BER
AWG	PAM	BCH	Modelo	500	37,016	7,988
AWG	PAM	Hamming	Modelo	500	19,168	8,636
AWG	PAM	Reed-Salomon	Modelo	2000	32,078	7,874
AWG	PAM	Turbocódigos	Modelo	1615	10,944	14,807
AWG	QPSK	BCH	Base de datos	500	26,732	9,920
AWG	QPSK	Hamming	Base de datos	500	14,296	9,090
AWG	QPSK	Reed-Salomon	Base de datos	1968	23,262	9,534
AWG	QPSK	Turbocódigos	Base de datos	3612	1,515	5,168
AWG	PSK	BCH	Base de datos	500	25,816	10,076
AWG	PSK	Hamming	Base de datos	500	14,172	8,919
AWG	PSK	Reed-Salomon	Base de datos	1950	23,015	9,660
AWG	PSK	Turbocódigos	Base de datos	3630	1,519	5,166



Canal	Modul.	Codifi.	Fuente	Bits enviados	BER promedio	Desviación típica BER
AWG	QAM	BCH	Modelo	500	26,316	10,073
AWG	QAM	Hamming	Modelo	500	14,456	9,057
AWG	QAM	Reed-Salomon	Modelo	1934	22,957	9,694
AWG	QAM	Turbocódigos	Modelo	3609	1,533	5,536
Ideal	PAM	BCH	Modelo	500	15,500	9,536
Ideal	PAM	Hamming	Modelo	500	11,500	9,474
Ideal	PAM	Reed-Salomon	Modelo	1500	7,167	5,713
Ideal	PAM	Turbocódigos	Modelo	3750	1,267	4,732
Ideal	QPSK	BCH	Base de datos	500	15,500	9,536
Ideal	QPSK	Hamming	Base de datos	500	11,500	9,474
Ideal	QPSK	Reed-Salomon	Base de datos	1500	7,167	5,713
Ideal	QPSK	Turbocódigos	Base de datos	3750	1,267	4,732
Ideal	PSK	BCH	Base de datos	500	15,500	9,526
Ideal	PSK	Hamming	Base de datos	500	11,500	9,474
Ideal	PSK	Reed-Salomon	Base de datos	1500	7,167	5,713
Ideal	PSK	Turbocódigos	Base de datos	3750	1,267	4,732
Ideal	QAM	BCH	Modelo	500	15,500	9,526
Ideal	QAM	Hamming	Modelo	500	11,500	9,474
Ideal	QAM	Reed-Salomon	Modelo	1500	7,167	5,713
Ideal	QAM	Turbocódigos	Modelo	37350	1,267	4,732

## 2.5 Análisis estadístico de resultados

Para realizar una correcta evaluación de los resultados, se procedió a desarrollar un análisis de varianza, y determinar si los valores obtenidos fueron significativamente diferentes y no un producto del azar. Sin embargo, en este análisis, se excluyó los datos obtenidos mediante la técnica de modulación PAM, dado que el fin de su adición en las pruebas fue para demostrar el efecto de un deficiente método de modulación - que no se usa en la práctica - independientemente del tipo de codificación que implementado en un sistema

de comunicación. Por esta razón, puede excluirse para fines de este análisis estadístico.

Se organizó los datos obtenidos en una estructura tabular respecto a los tratamientos del experimento[2], tal como se muestra en la Tabla 5.

**Tabla 5.** Tratamientos del experimento

	QPSK	PSK	QAM	$\sum_{j=1}^b x_{ij}$	$\frac{1}{b} \sum_{j=1}^b x_{ij}$	$\left(\sum_{j=1}^b x_{ij}\right)^2$	$\sum_{j=1}^b x_{ij}^2$
BCH	26,732	25,816	26,316	78,864	26,288	6219,531	2073,598
Hamming	14,296	14,712	14,456	43,464	14,488	1889,120	629,795
Reed-Salomon	23,262	23,015	22,957	69,234	23,078	4793,347	1597,835
Turbo-códigos	1,515	1,519	1,553	4,567	1,522	20,857	6,953
$\sum_{i=1}^a x_{ij}$	65,805	65,062	65,262				
$\frac{1}{a} \sum_{i=1}^a x_{ij}$	16,451	16,266	16,316				
$\left(\sum_{i=1}^a x_{ij}\right)^2$	4330,298	4233,063	4259,129				
$\sum_{i=1}^a x_{ij}^2$	1462,3913	1414,906	1430,882				

Con esta información, se procedió a realizar algunos cálculos auxiliares (Tabla 6) para formar la Tabla 7, que corresponde al análisis de varianza, propiamente dicho.

**Tabla 6.** Cálculos auxiliares sobre los tratamientos del experimento

Expresión	Valor
$\sum_{i=1}^a \sum_{j=1}^b x_{ij}$	196,129
$\frac{1}{ab} \sum_{i=1}^a \sum_{j=1}^b x_{ij}$	16,344
$\sum_{i=1}^a \left(\sum_{j=1}^b x_{ij}\right)^2$	12922,854
$\sum_{j=1}^b \left(\sum_{i=1}^a x_{ij}\right)^2$	12822,491
$\sum_{i=1}^a \sum_{j=1}^b x_{ij}^2$	4308,179
$v_r = \sum_{j=1}^b \left(\sum_{i=1}^a x_{ij}\right)^2 - \frac{1}{ab} \sum_{i=1}^a \sum_{j=1}^b x_{ij}$	1102,069

$$\begin{aligned}
 v_c &= \sum_{i=1}^a \left( \sum_{j=1}^b x_{ij} \right)^2 - \frac{1}{ab} \sum_{i=1}^a \sum_{j=1}^b x_{ij} && 0,0739 \\
 v &= \sum_{i=1}^a \sum_{j=1}^b x_{ij}^2 - \frac{1}{ab} \sum_{i=1}^a \sum_{j=1}^b x_{ij} && 1102,5307 \\
 v_e &= v - v_r - v_c && 0,4875
 \end{aligned}$$

Tabla 7. Análisis de varianza del experimento

Expresión	Grados de libertad	Cuadrado medio	F
$v_r = 1102,06929$	$a - 1 = 3$	$\hat{S}_r^2 = \frac{v_r}{a - 1} = 367,35643$	$F_{3,6} = \frac{\hat{S}_r^2}{\hat{S}_e^2} = 367,35643$
$v_c = 0,0739$	$b - 1 = 2$	$\hat{S}_c^2 = \frac{v_c}{b - 1} = 0,03695$	$F_{2,6} = \frac{\hat{S}_c^2}{\hat{S}_e^2} = 0,45482$
$v_e = 0,4875$	$(a - 1)(b - 1) = 6$	$\hat{S}_e^2 = \frac{v_e}{(a - 1)(b - 1)} = 0,08125$	
$v = 1102,6307$	$ab - 1 = 11$		

Conociendo que el tipo de codificación condiciona el desempeño dentro del protocolo; es necesario establecer si la tasa de bits de error BER obtenida por el uso de los turbocódigos, estadísticamente es superior a las otras técnicas de codificación. Dado que del análisis precedente se conoce que el tipo de modulación no varía significativamente los resultados, se eligió los valores de simulación obtenidos con la modulación PSK (escenarios A-PS-B-F, A-PS-H-F, A-PS-R-F y A-PS-T-F).

Definimos las hipótesis de contraste genéricas:

- $H_0 = \mu_x \geq \mu_{turbo}$ . La tasa de bits de error BER generada por los métodos de codificación puede ser igual o mayor a la obtenida mediante turbocódigos.
- $H_1 = \mu_x < \mu_{turbo}$ . La técnica de codificación «x» tiene una BER media menor que la BER resultado de la aplicación de los turbocódigos.

La técnica de codificación «x» puede hacer referencia a BCH, Hamming o Reed Salomon, según sea la situación. La Tabla 8 resume los procesos de cálculo de las pruebas de hipótesis para estas técnicas.

**Tabla 8.** Prueba estadística de contraste de hipótesis

Método de codif.	$n_{turbo}$	$\mu_x$	$n_x$	$\frac{\sigma_{\hat{x}_x - \hat{x}_{turbo}}^2}{\sqrt{\frac{\sigma_x^2}{n_x} + \frac{\sigma_{turbo}^2}{n_{turbo}}}}$	Z	$H_0$
BCH	500	25,816	101,534	0,459	-52,967	Refutada
Hamming	500	14,712	79,557	0,408	-13,193	Refutada
Reed-Salomon	1950	23,015	93,316	0,234	-21,496	Refutada
Turbocódigos	3630	1,519	26,693	-	-	-

## 2.6 Análisis de resultados

Los datos obtenidos por la extensa simulación del protocolo fueron sometidos a dos tipos de pruebas estadísticas: análisis de varianza y pruebas de hipótesis. La primera operación dejó dos conclusiones importantes:

- El método de modulación no influyó relevantemente en los resultados de la simulación. Las diferencias de su efecto entre las técnicas de codificación fue mínima.
- Por el contrario, cada proceso de codificación tuvo un impacto diferente en la tasa de errores, de tal forma que esta diversidad no se vio causada por un factor netamente aleatorio. Esta premisa se sustenta en la demostración de la variación de promedios BER.

Basándose en la segunda conclusión precedente, existe una tasa de error diferente en función del tipo de codificación que se aplica. Si consideramos el factor de desempeño entre cada tipo de codificación obtenemos que los turbocódigos minimizan la tasa de error hasta un 93% (en el caso de la confrontación con BCH).

Además, las pruebas de hipótesis de contraste se diseñaron para confirmar que los turbocódigos ofrecen un mejor rendimiento respecto a BCH, Hamming y Reed Salomon; y en todos los ensayos, estadísticamente, se puede sustentar - con un 0,2% de nivel de significancia - que los turbocódigos tienen la tasa de error más baja.

### 3 Conclusiones

Como resultado de la presente investigación, en base a las pruebas de simulación y comparaciones estadísticas, se concluye que la aplicación de los turbocódigos con el algoritmo iterativo BCJR como método de codificación de señales, dentro del protocolo experimental BTLCP (BiosignalsTransmission Link Control Procedure) mejora ampliamente la confiabilidad del transporte de las ondas cerebrales en un sistema digital inalámbrico, llegando a reducir la tasa de errores de bit más de 20 veces respecto a algunos tipos de codificación. Esta misma aseveración se obtuvo en diferentes condiciones de simulación de canal y modulación. De acuerdo a este argumento, se sostiene la certeza de la hipótesis planteada.

Además, de los ensayos experimentales se encontró que no existe una técnica de codificación ideal, puesto que en todos los algoritmos se hallaron valores especiales - singularidades o con alto número de errores - que los diferentes métodos fueron incapaces de detectar o corregir apropiadamente. Ante estas situaciones, la única alternativa para obtener las tramas correctas fue la retransmisión con el procedimiento descrito por el protocolo..

Al mismo tiempo, el análisis de varianza sobre los resultados de la simulación, evidenció una muy baja diferencia entre el uso de las técnicas de modulación analizadas, a pesar de las distintas probabilidades teóricas de error. Con este fundamento, se sustentó que no existe una variación significativa entre los diferentes tipos de métodos de modulación binaria y multinivel de fase en el transporte de bioseñales con el protocolo BTLCP.

También, con los avances desarrollados en la investigación se pudo demostrar que es posible generar una bioseñal cerebral aleatoria en base a un modelo matemático fundamentado en la descripción de la envoltura de su espectro en frecuencia. Las pruebas realizadas sostienen que la expresión hallada tiene un coeficiente de correlación mayor al 80% respecto a las señales incluidas en la base de datos. De este modo se asegura su validez de uso en el ambiente virtual que se creó exclusivamente para esta tesis, con el fin de realizar todas las simulaciones necesarias.

### Referencias

- [1] Enrique Mario Spinelli. *Interfaces de control cerebral*. Tesis maestra. Universidad Nacional de la Plata, 2000.
- [2] María T. Luque. *Mejora de la herramienta Mind Reader. Adquisición y análisis de señales EEG*. Tesis de fin de carrera. Universidad Carlos III de Madrid, 2010.

- 
- [3] A. B. Benevides, T. F. Bastos y M. Sarcinelli. *Selección de Parámetros para Clasificación de Tareas Mentales en Tiempo Real*. Simposio CEA de Bioingeniería, Grupo de Bioingeniería del Comité Español de Automática - Red Temática en Ingeniería Biomédica, 2010.
- [4] Ricardo Ron Angevin. *Retroalimentación en el entrenamiento de una interfaz cerebro-computadora usando técnicas basadas en realidad virtual*. Tesis doctoral. Universidad de Málaga, 2005.
- [5] I. Iturrate, L. Montesano y J. Minguez. *Detección en tiempo real de potenciales de error durante la observación de un brazo robótico*. Simposio CEA de Bioingeniería 2010. Simposio CEA de Bioingeniería, Grupo de Bioingeniería del Comité Español de Automática - Red Temática en Ingeniería Biomédica, 2010.
- [6] Lucio González Sanz. *Estudio conjunto de magnetoencefalografía y electroencefalografía en epilepsia*. Tesis doctoral. Universidad Complutense de Madrid, 2011.
- [7] Marco A. Pinto O. *Comunicación entre un dispositivo de adquisición de bioseñales cerebrales y un terminal móvil*. Tesis de grado de Ingeniería de Telecomunicaciones. Universidad Católica Boliviana. Cochabamba, Bolivia, 2012.
- [8] Andrew S. Tanenbaum. *Redes de computadoras*. Prentice Hall, 1997.
- [9] E. Ayanoglu, S. Paul, T. F. LaPorta, et. al. *A link-layer protocol for wireless networks*. *Wireless Networks*. J. C. Baltzer AG, Science Publishers, 1995.
- [10] G. Castellanos y Y. Shinakov. *Teoría de señales: Fundamentos*. Centro de publicaciones UN Manizales, 2009.
- [11] Simon Haykin. *Sistemas de comunicación*. Editorial Limusa, 2002.
- [12] Murray R. Spiegel. *Probabilidad y estadística*. Schaum's outline series, McGrawHill, 2003.