

Interfaz Humano-Máquina web amigable para dispositivo IoT.

Rocío Madou¹, Federico N. Guerrero¹, Enrique M. Spinelli¹

¹ Grupo de Instrumentación Biomédica Industrial y Científica (GIBIC),
LEICI (UNLP-CONICET), Facultad de Ingeniería UNLP,
116 y 48 S/N, La Plata, Argentina
{rocio.madou, federico.guerrero, spinelli}@ing.unlp.edu.ar

Resumen. Este trabajo presenta una solución para vincularse y configurar un sistema embebido que tiene conectividad WIFI pero que no dispone de ningún dispositivo de entrada ni visualización local. El equipo, diseñado bajo el paradigma IoT y denominado WIMUMO, permite adquirir señales biomédicas y transmitir las utilizando el protocolo OSC a través de una red inalámbrica. Por razones de seguridad eléctrica, se alimenta mediante baterías y está dispuesto en un gabinete plástico cerrado, sin ningún tipo de conector accesible. Sin embargo, esta característica dificulta interactuar con el equipo hasta que logra realizarse su configuración de red. Para facilitararlo, al encenderse WIMUMO genera una red propia y cuenta con un servidor web. Esto permite, mediante un browser de propósitos generales, configurar el equipo para que se conecte a una red existente, definir las direcciones a las cuales debe enviar las señales, entre otros parámetros. Toda la configuración se realiza mediante una interfaz web amigable que implementa criterios de usabilidad establecidos para el buen diseño de interfaces, pudiendo ser realizada sin problemas por usuarios sin más conocimiento que conectar su celular a una red WIFI. La solución propuesta se basa en un sistema multicapas. La capa más baja es capaz de interactuar, utilizando el lenguaje Bash, con los servicios a nivel sistema operativo, modificando los archivos de configuración siguiendo la lógica de la máquina de estados de conexión desarrollada. En las capas de alto nivel se encuentra el servidor desarrollado bajo NodeJS cuyo diseño visual está dado por la librería Bootstrap, encargado de la interacción, recolección y comunicación de datos del usuario, a través de la herramienta Socket.io.

Palabras clave: Internet de las cosas (IoT), interfaces humano-máquina (HMI), usabilidad.

1 Introducción

Un dispositivo dentro del paradigma de la internet de las cosas (IoT, en inglés) tiene la capacidad de transferir datos a través de una red sin requerir de interacciones humano a humano o humano a computadora. Esta tecnología, que comenzó en el ámbito académico-científico para un reducido conjunto de elementos, y llevada masivamente a los objetos del hogar u oficina por las empresas, se está estableciendo

en todos los ámbitos de la vida cotidiana, desde sistemas de monitoreo de la salud [1][2], hasta performances artísticas [3].

WIMUMO es un dispositivo de medida no invasivo, para procesamiento y visualización de señales electrofisiológicas o biopotenciales. Dentro de su amplio espectro de aplicaciones, puede utilizarse para performances artísticas. En este ámbito, permite tomar medidas de señales de electromiograma (EMG) generadas por un performer, que son captadas mediante electrodos y transmitidas dentro de una misma red LAN, utilizando el protocolo OSC, a un dispositivo generador de multimedia [4]. Se trata de un equipo diseñado con tecnología disponible comercialmente, el cual no posee una interfaz para la visualización o el ingreso de datos (como por ejemplo una pantalla), pero cuenta con conexión inalámbrica WiFi. En una primera versión, la instalación y configuración del equipo se realizaba a través de una conexión por consola remota y segura (SSH) por una persona con alto conocimiento técnico. Esta solución no es apropiada para este equipo, que será usado por personas sin conocimiento de la tecnología. Esto exige que su puesta en funcionamiento sea simple, amigable al usuario y cumpla con la heurística que define los criterios de usabilidad del dispositivo [5].

Teniendo en cuenta el ámbito al que está orientado WIMUMO y que debe operar en distintas locaciones, en algunos casos sin conexión a internet, debe ser utilizado y calibrado sin requerir una tecnología puntual por parte del usuario, i.e. requiere un sistema multiplataforma que cuente con un modo de configuración mínimo.

Además, debe explotar el conocimiento informal adquirido del usuario. Es decir, aprovechar el aprendizaje previo sobre la manipulación de un elemento que posea características similares, como lo puede ser la conexión de un celular a una red inalámbrica.

En la literatura se encuentran estrategias para dispositivos similares, que permiten la configuración usando una conexión USB [6] u otros medios que no son los tradicionales, como la configuración a través del sonido [7], que involucran cierta dependencia con la conectividad web, aplicaciones propietarias [8] o hardware específico. El desarrollo de aplicaciones locales multiplataforma puede ser muy costoso, ya que los elementos de software utilizados varían según el dispositivo en donde se va a trabajar. En cambio, las aplicaciones web corren sobre los navegadores ya incorporados por los fabricantes del equipo, las dependencias se limitan sólo a las características del navegador, sin requerir la instalación de herramientas adicionales obteniendo estabilidad y soporte independientemente de la plataforma.

Se propone como solución un sistema de capas que involucra la interacción del algoritmo como parte del sistema operativo, con un desarrollo de alto nivel encargado del manejo de la interfaz con el usuario a través de un servidor web, capaz de cumplir con los criterios de usabilidad considerados necesarios.

2 Materiales y métodos

En las siguientes subsecciones se describe la implementación de un sistema de configuración de redes a través de un servidor web, comenzando por la especificación de las herramientas necesarias para su desarrollo, y la descripción de una situación de uso promedio de un usuario, seguido de una descripción detallada de los pasos

necesarios para llevar a cabo dicha configuración. Luego, desde un punto de vista técnico, en primer lugar, se explica la lógica de la estructura de funcionamiento y, posteriormente, la implementación del mecanismo. Finalmente, se identifican los criterios a cumplir por la heurística de usabilidad.

2.1 Plataforma

WIMUMO se desarrolló a partir de una minicomputadora Raspberry Pi Zero W, controlada por el sistema operativo Raspbian Buster lite, versión septiembre 2019, siendo su comunicación con el usuario posible sólo a través de una interfaz de comandos. Raspbian cuenta con un conjunto de procesos, librerías y herramientas llamado systemd, encargado de la administración del sistema y los servicios. A éste, se le integró el servicio de avahi-daemon, para el reconocimiento del dispositivo a través de su nombre de host.

La Raspberry Pi Zero W posee conectividad inalámbrica WiFi, pero su hardware se limita a la comunicación con un único perfil de red, y no permite su configuración como access point (AP) y cliente al mismo tiempo.

Para implementar un servidor descripto más adelante, se utilizó Node.js v.12.16.2, el gestor de paquetes de node, npm v.6.14.4, y paquetes correspondientes al módulo de comunicación paralela entre servidor y cliente socket.io. Éste fue diseñado para coexistir y trabajar en conjunto con la muestra y el envío de datos desarrollados previamente.

2.2 Descripción del funcionamiento

WIMUMO es un equipo portátil que puede ser llevado a un laboratorio, consultorio, sala de ensayo, o cualquier locación. Quien lo utiliza podría ser cualquier persona interesada en el uso de las señales de biopotencial, hasta el momento han sido usuarios ligados a la performance artística. Al llegar a la locación, el usuario debe poder conectar WIMUMO a la infraestructura WiFi existente con relativa facilidad, o como mínimo poder acceder a la interfaz del equipo a través de una red provista por el equipo mismo.

Al presentarse por primera vez al usuario, WIMUMO se encuentra en modo access point. Por este motivo, para comenzar su configuración, el usuario se conecta a la red perteneciente al WIMUMO desde un dispositivo con conexión WiFi, ingresando las credenciales correspondientes. Una vez dentro de la misma, abre una ventana del navegador disponible e ingresa la dirección IP '192.168.99.1', elegida como la receptora de pedidos del servidor. En los dispositivos que cuenten con el servicio de red Bonjour, es posible utilizar el nombre de host wimumo00x.local.

Una vez dentro del servidor, el usuario se dirige a la sección "Redes" la cual cuenta con una interfaz similar al modo de conexión conocido y utilizado por los usuarios en gran parte de los dispositivos comerciales, en donde se solicita los datos de conexión de la red a la que se desea conectar. Una vez completados los campos, hace click al botón "conectar" que genera un mensaje indicador de que la configuración de red se está realizando. Si la configuración ha sido correcta, el usuario puede ver un mensaje

en fondo verde con instrucciones para comenzar su uso. Caso contrario, un indicador rojo le solicita que vuelva a la pantalla principal y reinicie la configuración.

A partir de este momento se encuentra una situación crítica para usuarios poco experimentados ya que enfrentan la desafiante tarea de averiguar la dirección IP que el equipo tomó en la red. Siguiendo las indicaciones presentadas en el “aviso verde de configuración exitosa”, el usuario debe lograr conectarse a la red que indicó previamente y luego de esperar a que WIMUMO esté también conectado, selecciona una de las tres opciones para encontrar la IP recibida por el router, en caso de que no se haya indicado una dirección estática:

- Usar un servicio disponible en la página de WIMUMO en internet.
- Escanear la red con una aplicación disponible en el *store* del sistema operativo (en el propio cartel se recomienda la utilización de *Bonjour Browser* por su facilidad de uso).
- Acceder utilizando el nombre de host *wimumo00x.local* desde un dispositivo que cuente con los servicios de red Bonjour.

A partir de este momento, el usuario tiene acceso a todas las prestaciones que el WIMUMO tiene para ofrecer desde su servidor.

2.3 Descripción del algoritmo

Para lograr el modo de funcionamiento descrito previamente, se implementó una máquina de estados que se integró al inicio del sistema operativo de la Raspberry.

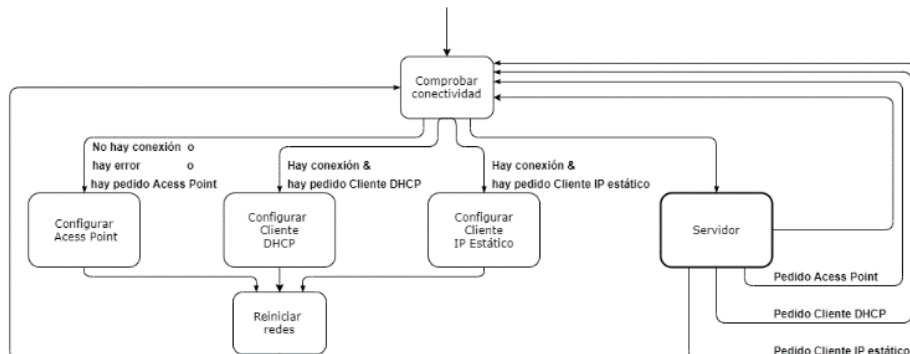


Fig. 1 Máquina de estados que representa el funcionamiento del algoritmo de configuración de modo de conexión del WIMUMO.

Como se puede ver en la Figura 1, la configuración comienza con la comprobación de conectividad del dispositivo. La primera vez que se inicia la Raspberry, no cuenta con ninguna configuración preexistente, por lo que no puede poseer un modo de conexión. Al detectar este estado, se llama a la solicitud de configuración como access point. Una vez finalizado, se reinicia el sistema de red y se vuelve a comprobar la existencia de una conexión. Esta devuelve una confirmación, lo que permite la puesta en funcionamiento del servidor.

El modo descrito se definió como *modo default*, ya que en caso de no posibilidad de conexión o falla en los archivos involucrados, siempre vuelve al modo AP.

Con el servidor en funcionamiento el usuario puede solicitar el modo de cliente de red, utilizando los protocolos DHCP (IP asignada de forma dinámica por el router) o IP estática (asignada por el usuario). Ingresados los datos correspondientes a cada protocolo, se inicia la configuración y, una vez finalizado, el reinicio de red.

Por última vez, se vuelve a comprobar de forma exitosa la conexión a la red y se reinicia el servidor disponible para comenzar con la adquisición.

2.4 Descripción de la implementación

El desarrollo de la solución de la problemática se dividió en dos capas. La primera, de bajo nivel, en donde la combinación de algoritmos se transforma en servicios del sistema operativo para una mejor sincronización y estabilidad ante la posibilidad de errores.

Simulando el funcionamiento de comandos desde una consola, Bash es el lenguaje elegido para los algoritmos correspondientes a la máquina de estados, programa principal y funciones independientes que contienen el accionar de cada estado, y la administración servidor. Se utilizaron *flags* o banderas de estado, a través de la presencia o ausencia de archivos vacíos sin extensión. Los scripts principales de la máquina y el servidor tienen asociado un servicio particular, a través del cual se desarrolla su puesta en funcionamiento o detención. Por ello, la máquina de estados puede llamar a un script de configuración, o iniciar o detener cualquier servicio del sistema operativo, según corresponda, al igual que lo hacen el resto de los bloques administrados por Systemd.

En una capa superior, donde se encuentra la aplicación web, los elementos de comunicación cliente-servidor y sus aspectos funcionales y estéticos, los datos de conexión son obtenidos, procesados y almacenados mediante una interfaz que combina la utilización de formularios, listas y botones, junto con avisos de estado e instructivos.

En el momento en que el usuario ingresa la IP asignada al WIMUMO en la barra de búsqueda del navegador, se inicia la ejecución de una serie de etapas de interacción cliente-servidor [2]. El servidor recibe un pedido de conexión, emitiendo una respuesta que incluye la renderización de una página HTML, cuyo funcionamiento está determinado por código JavaScript embebido en la misma, y su diseño por los elementos de la librería Bootstrap personalizados en archivos CSS. Además, se abre un canal de comunicación asincrónica paralela a través del uso de socket.io.

Al cargarse la pestaña de “configuración de redes”, comienza el llamado sincrónico de funciones que derivan en la solicitud al sistema operativo, transferencia por sockets y muestra de una lista de redes visibles por la antena de la placa, como primer elemento del formulario que compone la interfaz. Una vez seleccionada la SSID, o nombre de red, por el usuario, es necesario que se complete el campo de texto oculto correspondiente a la contraseña de la misma. Bajo el mismo, se muestra un checkbox, que habilita la muestra de los campos y el ingreso de los datos necesarios para la configuración del protocolo IP estático (dirección IP deseada, máscara de red, dirección del router y DNS). También es posible activar la solicitud de modo access

point, activando el checkbox correspondiente, el cual oculta todos los campos de ingreso de datos y selecciones, dejando sólo visible el botón de confirmación del formulario.

Al hacer clic a este último, se recolectan los datos correspondientes, dependiendo de la configuración deseada por el usuario, y se envían a través del websocket a la primera función de configuración de parámetros dentro de servidor, mostrándose una nueva pestaña, la cual contiene una barra de carga y un mensaje que indica que el proceso está en curso. A partir de este momento, se comienza a trabajar con un módulo que permite la ejecución de comandos en una consola, y la modificación de archivos de cualquier directorio del sistema operativo. Dichos datos son copiados en los archivos de redes correspondientes, los flags con creados o eliminados, y el servicio de la máquina de estados es iniciado.

Si la secuencia de elementos fue realizada con éxito, el usuario visualiza en pantalla un instructivo de acciones para poder volver a conectarse al servidor y hacer uso del mismo. En caso contrario, se le indica el fallo de la operación, solicitando que vuelva a iniciarla.

2.5 Descripción de la métrica para determinación de usabilidad

El estándar ISO 9241-11 [9] define el término usabilidad como “el alcance en el que un producto puede ser utilizado por usuarios específicos para alcanzar metas determinadas con efectividad, eficiencia y satisfacción en un contexto particular de uso”, por lo que los criterios de calidad en la experiencia de usuario son variables dependiendo la aplicación.

En el caso de los dispositivos electrónicos o aplicaciones móviles, existen una serie de factores que pueden afectar la usabilidad del instrumento, desde la interfaz gráfica, o fisiología del usuario, hasta los aspectos sociales [10]. Es por ello que se estableció cierta heurística o principios del “buen diseño de interfaces” [5].

De acuerdo a la implementación que se buscaba para la tecnología, según la heurística establecida en el artículo citado previamente, las características que se consideró necesario su cumplimiento fueron: Diseño visual intuitivo, Visibilidad del estado del sistema, Correspondencia entre el sistema y el mundo real, Flexibilidad y eficiencia de uso, Prevención de errores, Ayuda y documentación.

3 Resultados

La estrategia adoptada permitió obtener un dispositivo capaz de modificar sus configuraciones a través de un sistema multicapa con comunicación dentro y entre las mismas. En donde, la capa de más alto nivel, encargada de la interpretación directa del usuario, obtiene los datos del nuevo modo de operación, y la de más bajo nivel, integrada dentro del sistema operativo, administra los archivos y servicios logrando la nueva configuración.

Previamente, si el equipo contaba con una versión de configuración previa, era imposible la comunicación sin la intervención de periféricos. Al momento de escribir este artículo, se dispone de un equipo que, ante la imposibilidad de conectarse, auto-

configura su modo de funcionamiento en access point, logrando una comunicación remota no sólo a través de una consola de comandos, sino con un servidor que contiene una interfaz web amigable al usuario, e intuitiva.

Buscando cumplir con la heurística de usabilidad de interfaces de software, se obtuvo un conjunto de pantallas aptas para la explotación del conocimiento previamente adquirido por el usuario en sistemas con similares características de conexión, las cuales, con un diseño minimalista evitando la sobrecarga de información, se pueden definir como auto-explicativas y a partir de su interacción, orientan al usuario retornando un feedback dinámico sobre el estado del equipo en cada momento, Figura 2. El soporte back-end del servidor, con una comunicación asincrónica a través de sockets, asegura que la necesidad de participación del usuario se limite al ingreso de credenciales y a la espera correspondiente a la demora de la configuración.

Al desarrollo de la tecnología se le agregó la redacción de un instructivo mínimo en formato papel, indicando los pasos a seguir por el usuario, junto con los tiempos de retardo que corresponden a cada etapa, evitando de esta forma que la impaciencia del mismo pueda llevar a un apagado abrupto causado por la persona en medio de una configuración.

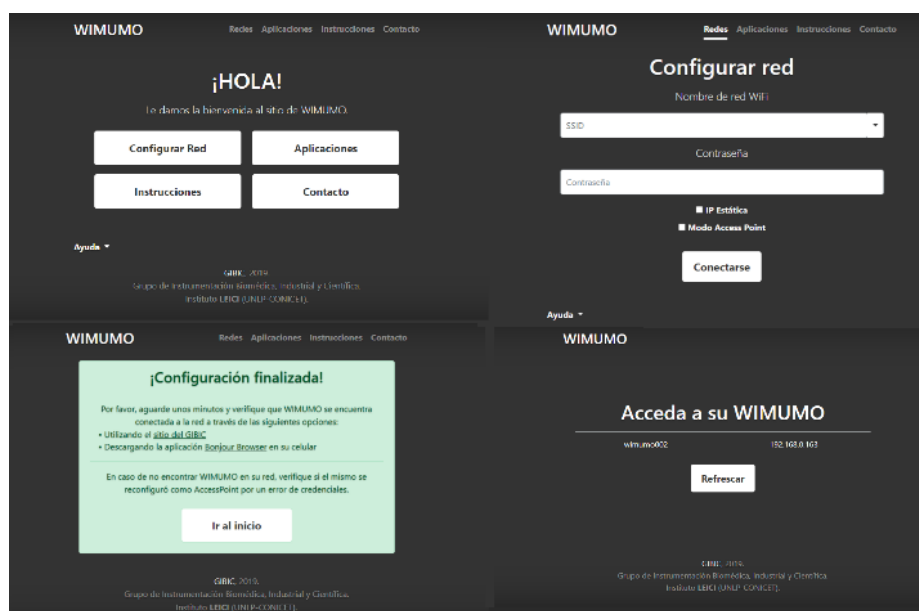


Fig. 2: Diseño de las pantallas que componen el camino de configuración exitoso del WIMUMO.

Gracias a estas características, personas con distinto nivel de conocimiento técnico pudieron cambiar la red fácilmente. Se puso a prueba el rendimiento del equipo y la comunicación con otros dispositivos siendo cliente de una red o administrador de la propia. Además, se comprobó la robustez del sistema ante el ingreso de credenciales incorrectas o la mala señal de red por distanciamiento entre dispositivos.

Además, fue posible la identificación aproximada de los tiempos de funcionamiento del equipo. Al encenderse, si la conexión es exitosa, el servidor tarda 90 segundos en estar disponible y accesible. El peor caso, donde no se puede obtener una IP ya que la red no está disponible y se realiza un cambio de configuración a AP, se aproxima a los 3 minutos. La demora de la configuración de una nueva red a través del servidor toma 2 minutos en promedio.

3.1 Implementación de usabilidad

Los criterios de usabilidad fueron satisfechos con éxito a partir de las siguientes implementaciones:

Un *diseño visual intuitivo* se logró a partir de la utilización de elementos simples, con nombres auto-explicativos y textos de ayuda cortos. Los mensajes fueron ideados para ser explicativos y junto con los colores de las alertas y la barra dinámica de carga, permiten *visibilizar el estado del sistema*. Cada pieza dispuesta en un diseño de interfaz conocido previamente por el usuario lleva a que *el sistema pueda corresponderse con el mundo real*. La utilización de las herramientas dinámicas de Bootstrap permitieron la *flexibilidad* necesaria para cada pantalla y *eficiencia de uso* en las características de los distintos dispositivos.

En cuanto a la *prevención de errores*, se implementó con la verificación de que todos los datos necesarios sean ingresados en los campos de texto y la adopción del modo AP ante el ingreso de datos de conexión erróneos por parte del usuario.

Finalmente, se incorporaron dos instructivos al equipo como *ayuda y documentación*. El primero virtual disponible como una pestaña del servidor ante la consulta necesaria al momento de ingreso de las credenciales nuevas. El segundo, anexo al equipo en formato papel, indica los pasos necesarios para lograr una conexión exitosa.

4 Discusión

A partir de las pruebas realizadas fue posible distinguir una clara relación entre el conocimiento técnico de una persona y la necesidad de documentación para la configuración del equipo. Para los usuarios con mayor grado de tecnicidad, la documentación sólo fue necesaria al comenzar con la ejecución. En el caso contrario, el no conocimiento sobre el funcionamiento de las redes, limitó su progreso en todos los pasos que involucraban el accionar con cambio de configuración y dirección IP; momentos como la conexión inicial en modo access point, el ingreso al servidor con una dirección particular, la búsqueda de la nueva dirección IP por la solicitud de conexión como cliente con el protocolo DHCP, y el ingreso de dicha dirección para poder volver acceder al servidor. Sin una documentación clara y detallada, los usuarios con este nivel de conocimiento no podrían utilizar el dispositivo.

Existen otras plataformas, como el ESP32, que poseen doble perfil de red. De esta forma es posible la comunicación con el equipo a través de una red externa o la propia del WIMUMO simultáneamente. Esto permite que no sea necesario el cambio de configuración, por lo que un usuario con pocos conocimientos técnicos puede

interaccionar con el servidor desde su dirección de administrador de red, mientras se realiza la comunicación con otros instrumentos a través de la red externa, como cliente.

La retroalimentación del usuario, al igual que la documentación, es un factor que ganó importancia a medida que desarrollaron los componentes. Al presentarse como una caja cerrada, el equipo no cuenta con una respuesta que le permita al usuario conocer el estado del mismo. Esto generó necesidad de implementar notificaciones dentro de la interfaz, indicando, por ejemplo, si se encuentra dentro de los pasos de procesamiento de la información para la configuración de un nuevo perfil de red o, en caso de finalizado, el resultado del proceso. Sin embargo, existe una porción de tiempo en donde el servidor no se encuentra disponible, por lo que se considera importante, con posibilidad de adoptar a futuro, la incorporación de un led RGB indicador de estado.

Lo dicho anteriormente toma mayor relevancia al considerar los tiempos de funcionamiento del mismo. Actualmente, los tiempos de operación son tolerables gracias a los elementos indicadores de estado propios de la interfaz. Sin embargo, se busca reducir dicha tardanza optimizando el software buscando hacer mejor uso de la capacidad computacional de la placa, e incluso se propone pasar a una plataforma como el ESP32.

Es importante destacar que los algoritmos como servicios permitieron una buena integración al sistema operativo estableciendo explícitamente el momento en que se ejecutan los mismos dentro de la secuencia de inicio.

Finalmente, se encontró con una limitación en la flexibilidad del funcionamiento del equipo. Si la red externa que fue configurada por el usuario no se encuentra disponible al momento de encendido, este se auto-configura a su red propia olvidando la conexión anterior, teniéndose que volver a realizar el proceso de cambio de red. En el caso donde se trabaja siempre en la misma red, es esencial que se respeten los pasos de encendido de cada uno de los elementos involucrados, volviéndolo impráctico. Por lo que es necesario la creación de perfiles de red, comprobando por la última configuración como cliente al encender el equipo.

5 Conclusiones

El trabajo desarrollado consiste en un sistema de configuración remota de un dispositivo IoT, a través de un servidor web. La utilización de dicha tecnología, permite el acceso a una interfaz amigable e intuitiva con cualquier plataforma con conexión a inalámbrica, que contenga un navegador, pudiendo explotar el conocimiento informal adquirido previamente por el usuario, en dispositivos similares.

El desarrollo de los algoritmos de modificación de configuración en lenguaje Bash manipulados como servicios, posibilita una mejor integración con el sistema operativo Raspbian, añadiendo robustez ante fallos. En cuanto a la interfaz de usuario brindada por el servidor, la utilización de la herramienta Node.js permite el desarrollo rápido y de bajo costo, utilizando de paquetes estables con buen soporte.

Además, se cumple con los principios para un correcto diseño de interfaces, destacando la importancia de la incorporación de documentación clara y ordenada para la interacción con el dispositivo.

Dada la relevancia dentro del tema, se establece como trabajo futuro la realización y documentación de pruebas de campo con usuarios reales.

6 Agradecimientos

Este trabajo fue financiado en parte mediante los subsidios I-219 de la Universidad Nacional de La Plata, PIP0558 del CONICET y PICT-2015/2257 de la Agencia Nacional de Promoción Científica y Tecnológica. El trabajo forma parte del proyecto de extensión acreditado FI UNLP “Fabricación de equipos vestibles de medición de señales bioeléctricas para uso en performances artísticas inclusivas”.

7 Referencias

1. M. Giannetto, V. Bianchi, S. Gentili, S. Fortunati, I. De Munari, and M. Careri: An integrated IoT-Wi-Fi board for remote data acquisition and sharing from innovative immunosensors. Case of study: Diagnosis of celiac disease. In: *Sensors Actuators, B Chem.*, vol. 273, no. June, pp. 1395–1403 (2018), doi: 10.1016/j.snb.2018.07.056.
2. V. I. Kubov, Y. Y. Dymyrov, and R. M. Kubova: Wireless Devices HTML-interface for Medical Applications. In: 2019 8th Mediterr. Conf. Embed. Comput. MECO 2019 - Proc., no. June, pp. 1–4 (2019) doi: 10.1109/MECO.2019.8760057.
3. A. Ceriani, R. Madou, M. Haberman, and E. Spinelli: Bioelectric signals of the body: from electronic engineering to artistic performance. In: *EAI Endorsed Trans. Creat. Technol.*, vol. 7, no. 22, p. 163481 (2020), doi: 10.4108/eai.13-7-2018.163481.
4. F. N. Guerrero et al.: WIMUMO Project: A Wearable Open Device for Physiological Signals Acquisition. In: *Revista Argentina de Bioingeniería*, vol. 24, no. 3, pp. 13–18 (2020).
5. A. Alexandru, M. Ianculescu, and D. Coardos: Improved patient engagement in self-management of health, a key to sustainable preventative healthcare systems. In: vol. 247, Springer International Publishing (2018).
6. M. O. Jewell, E. Costanza, and J. Kittley-Davies Agents: Connecting the things to the internet: An evaluation of four configuration strategies for wi-fi devices with minimal user interfaces. In: *UbiComp 2015 - Proc. 2015 ACM Int. Jt. Conf. Pervasive Ubiquitous Comput.*, pp. 767–778 (2015), doi: 10.1145/2750858.2807535.
7. L. Liu, Z. Han, L. Fang, and Z. Ma: Tell the device password: Smart device wi-fi connection based on audio waves. In: *Sensors (Switzerland)*, vol. 19, no. 3 (2019), doi: 10.3390/s19030618.
8. D. T. Nguyen and T. Kim: An SDN-Based Connectivity Control System for Wi-Fi Devices. In: *Wirel. Commun. Mob. Comput.*, vol. 2018 (2018), doi: 10.1155/2018/9359878.
9. ISO/IEC. ISO 9241-11 Ergonomic Requirements for Office Work with Visual Display Terminals (VDTs). Part 11. Guidance on Usability, ISO/IEC 9241-11:1998(E) (1998).
10. D. Svanæs, O. A. Alsos, and Y. Dahl.: Usability testing of mobile ICT for clinical settings: Methodological and practical challenges. In: *Int. J. Med. Inform.*, vol. 79, no. 4 (2010), doi: 10.1016/j.ijmedinf.2008.06.014.